

[Skip to main content](#)

API documentation

Endpoint

All requests to the external API are made through a single endpoint:

```
http://localhost/totara/server/api/graphql.php
```

Requests are made via the HTTP POST method with a JSON-encoded body:

```
$ curl 'http://localhost/totara/server/api/graphql.php' \  
-X POST \  
-H 'Authorization: Bearer YOUR-API-BEARER-TOKEN' \  
-H 'Content-Type: application/json' \  
-H 'Accept: application/json' \  
--data-binary '{ "query": "query { totara_webapi_status { status } }"}'
```

See the [Authentication](#) section below for how to obtain a valid Bearer token.

The expected response for this request would be the following JSON object:

```
{  
  "data": {  
    "totara_webapi_status": {  
      "status": "ok"  
    }  
  }  
}
```

Authentication

The external API uses OAuth 2.0 via the client credentials grant type to authenticate requests. This involves three steps as outlined below.

1. Register a client

To register a client, log in to your Totara site and navigate to 'API Clients'.

Click the **Add client** button and provide information to describe the purpose of your client.

Make a note of the client id and secret for step 2.

2. Request a token

To programmatically request a token, call the OAuth 2.0 token endpoint as follows, passing the `client_id` and `client_secret` obtained during step 1:

```
curl -X POST 'http://localhost/totara/server/totara/oauth2/token.php' \  
-H 'Content-Type: application/x-www-form-urlencoded' \  
-d 'grant_type=client_credentials&client_id=CLIENT_ID_HERE&client_secret'
```

The response will be:

```
{  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "access_token": "YOUR-API-BEARER-TOKEN"  
}
```

3. Submit a request with a valid token

Copy the value from the `"access_token"` property in the response into the `'Authorization: Bearer'` header of your request.

Request format

A simple GraphQL request might look like this:

```
query my_query {  
  totara_webapi_status {  
    status  
  }  
}
```

This request consists of the following.

The query keyword

This indicates this is a read-only request. `'mutation'` is used for requests which write data. The keyword is optional, but when provided the specific actions within the request must match the keyword given.

Request name (`my_query`)

Optional string to allow the request author to describe the purpose of the request. It is not used by the server.

Query or mutation name (totara_webapi_status)

Within the outer curly braces you must specify at least one query or mutation to execute (typically just one but requesting multiple is allowed).

The available queries and mutations are specified in the reference documentation.

Response structure

In GraphQL it is up to the query author to specify the data they want to receive in the response.

The curly braces and field names within them specify the data structure. They must match the structure of the type that is returned by that query or mutation.

Types can be complex (contain properties that represent other types), so additional curly braces are used to obtain properties of subresources in one request. For example, to get a list of course names and the name of each course's category:

```
query get_courses {
  get_courses {
    fullname
    category {
      name
    }
  }
}
```

Arguments and variables

Some queries and mutations have optional or required arguments which require additional data to be passed with the request.

User-provided data is kept separate from the body of the request as variables. Variables are passed as a separate JSON object with the query. To make use of variables, the structure of the query changes slightly:

Query

```
query get_course($courseid: core_id!) {
  core_course(courseid: $courseid) {
    fullname
  }
}
```

Variables

```
{  
  "courseid": 2  
}
```

In this example the 'core_course' query has an argument 'courseid'. It must be of the type 'core_id' and it is required (the ! suffix is used when a field must be provided). Available arguments are listed in the reference documentation for a query or mutation. Although values can also be hardcoded in the query, it is good practice to use variables to support argument validation and query reuse.

Variables are represented by strings starting with the dollar sign (\$). Any \$variable specified within the body of the request must be defined in the round brackets after the outer query name. When defining a variable you must specify its type. Variables will be validated according to their type before query execution. Complex types (types with properties made of other types) are allowed but must be defined in the schema.

The variables object that is passed with the request must have keys that match the variable names and give values that are compatible with the specified type for that variable.

Field arguments

Like queries and mutations, fields can support arguments. For example, here the argument on the timestamp field determines how the field is formatted in the response:

```
query test {  
  totara_webapi_status {  
    status  
    timestamp(format: DATETIMELONG)  
  }  
}
```

Aliases

You can prefix a query, mutation or field with a different name and the server will return it as the key you specify. This can be used to return data to match a certain structure, or to differentiate if you are requesting the same field multiple times.

```
query test {  
  my_query_name: totara_webapi_status {  
    status  
    long_year: timestamp(format: DATETIMELONG)  
    short_year: timestamp(format: DATETIMESHORT)  
  }  
}
```

would return:

```
{
  "data": {
    "my_query_name": {
      "status": "ok",
      "long_year": "27/05/2022, 10:51",
      "short_year": "27/05/22, 10:51"
    }
  }
}
```

For more information on the GraphQL language see this [Introduction to GraphQL](#).

API schema

You can [perform introspection](#) against the external API directly.

Alternatively, you can download the current external API schema for your site here:
<http://localhost/totara/server/totara/api/documentation/schema.php>

Core

Queries

core_course_courses

Retrieve a paginated list of all available courses.

Response

Returns a [core_course_courses_result!](#)

Arguments

Name	Description
query - core_course_courses_query	The query parameter contains pagination and sorting information.

QUERY

```
query core_course_courses($query: core_course_courses_query) {
  core_course_courses(query: $query) {
    items {
      ...core_courseFragment
    }
  }
}
```

```
    }
    total
    next_cursor
  }
}
```

VARIABLES

```
{"query": core_course_courses_query}
```

RESPONSE

```
{
  "data": {
    "core_course_courses": {
      "items": [core_course],
      "total": 123,
      "next_cursor": "xyz789"
    }
  }
}
```

Queries

core_course_is_user_enrolled

Query to check whether the user is enrolled a course.

Response

Returns a [core_course_is_user_enrolled_result!](#)

Arguments

Name	Description
course - core_course_course_reference!	Target course.
user - core_user_user_reference!	Target user.

QUERY

```
query core_course_is_user_enrolled(
  $course: core_course_course_reference!,
  $user: core_user_user_reference!
) {
  core_course_is_user_enrolled(
    course: $course,
    user: $user
  )
}
```

```
) {  
  user_currently_enrolled  
}
```

VARIABLES

```
{  
  "course": core_course_course_reference,  
  "user": core_user_user_reference  
}
```

RESPONSE

```
{"data": {"core_course_is_user_enrolled": {"user_currently_enrolled": true
```

Types

Boolean

The Boolean scalar type represents true or false.

EXAMPLE

```
true
```

Types

Float

The Float scalar type represents signed double-precision fractional values as specified by [IEEE 754](#).

EXAMPLE

```
987.65
```

Types

Int

The `Int` scalar type represents non-fractional signed whole numeric values. `Int` can represent values between $-(2^{31})$ and $2^{31} - 1$.

EXAMPLE

```
123
```

Types

String

The `String` scalar type represents textual data, represented as UTF-8 character sequences. The `String` type is most often used by GraphQL to represent free-form human-readable text.

EXAMPLE

```
"xyz789"
```

Types

core_category

A single category. Categories are a hierarchical structure which contain learning items such as courses.

Field Name	Description
<code>id</code> - <code>core_id!</code>	Category ID.
<code>name</code> - <code>String!</code>	Category name.
<hr/> <i>Arguments</i> <hr/>	
<code>format</code> - <code>core_format</code>	
<code>idnumber</code> - <code>String</code>	Category idnumber
<code>description</code> - <code>String</code>	Category description
<hr/> <i>Arguments</i> <hr/>	
<code>descriptionformat</code> - <code>core_format</code>	The format of the categories description
<code>full_path</code> - <code>String</code>	The displayable path of the category. I.e. "Grandparent Cat1 / Parent Cat2 / This Cat 3"

Field Name	Description
Arguments	
format - <code>core_format</code>	
parent - <code>core_category</code>	Parent category this category is inside, or null for top level categories.
children - <code>[core_category!]!</code>	Sub-categories that are inside this category.
depth - <code>Int</code>	Depth of this category in the tree. One means top-level category, add one for each level below that.
courses - <code>[core_course!]!</code>	Courses that are inside this category.
coursecount - <code>Int!</code>	Number of courses in this category (not including courses in sub-categories).
timemodified - <code>core_date</code>	
Arguments	
format - <code>core_date_format</code>	
theme - <code>String</code>	Force theme used in category, or null if not overridden in category.

EXAMPLE

```
{
  "id": 4,
  "name": "xyz789",
  "idnumber": "abc123",
  "description": "abc123",
  "descriptionformat": "RAW",
  "full_path": "abc123",
  "parent": core_category,
  "children": [core_category],
  "depth": 123,
  "courses": [core_course],
  "coursecount": 987,
  "timemodified": "2022-04-17",
  "theme": "abc123"
}
```

Types

core_course

A single course object. Courses are containers for a collection of activities that make up a unit of learning.

Field Name	Description
<code>id</code> - <code>core_id!</code>	ID of the course
<code>idnumber</code> - <code>String</code>	Course idnumber
<code>fullname</code> - <code>String!</code>	Course fullname
Arguments	
<code>format</code> - <code>core_format</code>	
<code>shortname</code> - <code>String!</code>	Course shortname
Arguments	
<code>format</code> - <code>core_format</code>	
<code>summary</code> - <code>String</code>	Course summary
Arguments	
<code>format</code> - <code>core_format</code>	
<code>summaryformat</code> - <code>core_format</code>	The format of the courses summary
<code>timecreated</code> - <code>core_date</code>	The date/time the course was created
Arguments	
<code>format</code> - <code>core_date_format</code>	
<code>timemodified</code> - <code>core_date</code>	The date/time the course was last modified
Arguments	
<code>format</code> - <code>core_date_format</code>	
<code>category</code> - <code>core_category</code>	Category this course is in, or null for the frontpage course.
<code>categoryid</code> - <code>core_id</code>	ID of this course's category, or zero for the frontpage course.
<code>startdate</code> - <code>core_date</code>	Course start date
Arguments	

Field Name	Description
format - core_date_format	
enddate - core_date	Course end date
Arguments	
format - core_date_format	
theme - String	Force theme used in course, or null if not overridden in course.
lang - String	Force language used in course, or null if language not overridden in course.
format - String	Course format.
coursetype - Int	Course type values are integers matching TOTARA_COURSE_TYPE_* constants defined in course/lib.php.
icon - String	Name of course icon used to represent the course, or 'default' if no icon has been set.
image - String	A URL for the course image
sections - [core_course_section!]!	Course sections/topics/weeks
modcount - Int	The number of modules in the course accessible to the user.
showgrades - Boolean!	The showgrades (show gradebook to learners) flag for the course
completionenabled - Boolean!	The enablecompletion (enable completion tracking) flag for the course
completion - core_course_completion	The completion information relating to the course and current user
criteria - [core_course_criteria!]!	The completion criteria for the course, and associated completion info for the current user
criteriaaggregation - String	The aggregation method for the overall course criteria
url - String!	Course view url.
course_format - core_course_format!	The course's format metadata information.
tags - [core_tag]	The tags applied to a course.
custom_fields - [totara_customfield_field!]!	The custom fields associated with a course.

```

{
  "id": 4,
  "idnumber": "abc123",
  "fullname": "abc123",
  "shortname": "abc123",
  "summary": "abc123",
  "summaryformat": "RAW",
  "timecreated": "2022-04-17",
  "timemodified": "2022-04-17",
  "category": core_category,
  "categoryid": 4,
  "startdate": "2022-04-17",
  "enddate": "2022-04-17",
  "theme": "xyz789",
  "lang": "xyz789",
  "format": "xyz789",
  "coursetype": 987,
  "icon": "abc123",
  "image": "abc123",
  "sections": [core_course_section],
  "modcount": 123,
  "showgrades": true,
  "completionenabled": true,
  "completion": core_course_completion,
  "criteria": [core_course_criteria],
  "criteriaaggregation": "xyz789",
  "url": "xyz789",
  "course_format": core_course_format,
  "tags": [core_tag],
  "custom_fields": [totara_customfield_field]
}

```

Types

core_course_completion

A single course completion object. A completion object tracks the status of a specific user in a specific course.

Field Name	Description
<code>id</code> - Int	The id of the completion record, however this can be null if not started or not enrolled.
<code>status</code> - String	A string representing the current learner's completion state in a course
<code>statuskey</code> - String	A string used as a <code>get_string</code> key for a longer completion status description

Field Name	Description
Arguments	
format - <code>core_format</code>	
progress - <code>Float</code>	A percentage describing how far the user has progressed through this course
timecompleted - <code>core_date</code>	The time the user completed this course
Arguments	
format - <code>core_date_format</code>	
gradefinal - <code>Float</code>	The user's grade in the course as an absolute value
grademax - <code>Float</code>	The maximum achievable grade for the course
gradepercentage - <code>Float</code>	The user's grade in the course as a percentage of grademax

EXAMPLE

```
{
  "id": 123,
  "status": "abc123",
  "statuskey": "xyz789",
  "progress": 123.45,
  "timecompleted": "2022-04-17",
  "gradefinal": 123.45,
  "grademax": 987.65,
  "gradepercentage": 987.65
}
```

Types

core_course_course_reference

Input for identifying a course.

The course must be specified by providing one of the following:

- The course's internal database id
- The course's idnumber
- The course's shortname

Input Field	Description
id - <code>core_id</code>	Internal database id of the course.
shortname - <code>String</code>	Shortname of the course.
idnumber - <code>String</code>	The id number of the course.

EXAMPLE

```
{
  "id": 4,
  "shortname": "abc123",
  "idnumber": "xyz789"
}
```

Types

core_course_courses_filters

Input type used when querying for a list of courses. Specifies filters to be applied to the results.

Input Field	Description
since_timecreated - <code>core_date</code>	Unix creation timestamp of tag including an integer timestamp, numeric string or an ISO-8601 string.
since_timemodified - <code>core_date</code>	Unix timemodification timestamp of tag including an integer timestamp, numeric string or an ISO-8601 string.
tag_filter - <code>String</code>	Raw name of a tag.

EXAMPLE

```
{
  "since_timecreated": "1696837540",
  "since_timemodified": "1696837540",
  "tag_filter": "education"
}
```

Types

core_course_courses_query

Input type for retrieving the list of all available courses. The query parameter includes pagination and sorting information for organising results.

Input Field	Description
pagination - <code>core_pagination_input</code>	The pagination object contains information about the number of items to be returned, and the page offset to start from.
sort - <code>[core_sort_input!]</code>	The sort order for the query. The query is currently only able to sort by one order.
filters - <code>core_course_courses_filters</code>	The filters for querying course.

EXAMPLE

```
{
  "pagination": core_pagination_input,
  "sort": [core_sort_input],
  "filters": core_course_courses_filters
}
```

Types

core_course_courses_result

The result of the courses query.

Field Name	Description
items - <code>[core_course]!</code>	A list of vector course items.
total - <code>Int!</code>	The number of returned courses.
next_cursor - <code>String!</code>	The cursor indicates where the returned result ends, and where the next query should begin.

EXAMPLE

```
{
  "items": [core_course],
  "total": 987,
  "next_cursor": "abc123"
}
```

Types

core_course_criteria

A single course criteria object. Criteria define the rules that determine whether a user has completed a course or activity.

Field Name	Description
<code>id - core_id!</code>	The id of the course module
<code>type - String</code>	The type of the criteria
<code>typeaggregation - String</code>	The aggregation used for this criteria type grouping
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - <code>core_format</code></p> </div>	
<code>criteria - String</code>	The name or descriptor of the criteria
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - <code>core_format</code></p> </div>	
<code>requirement - String</code>	What needs to be done to complete the criteria
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - <code>core_format</code></p> </div>	
<code>status - String</code>	How far the user is through the requirement
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - <code>core_format</code></p> </div>	
<code>complete - Boolean</code>	Whether the current user has completed the criteria
<code>completiondate - core_date</code>	The date the current user completed the criteria
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - <code>core_date_format</code></p> </div>	

EXAMPLE

```
{
  "id": 4,
  "type": "abc123",
  "typeaggregation": "abc123",
  "criteria": "xyz789",
  "requirement": "xyz789",
  "status": "xyz789",
  "complete": true,
```

```
"completiondate": "2022-04-17"
}
```

Types

core_course_format

GraphQL type for the course's format.

Field Name	Description
format - String!	The type name of course format, which can be any of the course's format plugin names. For example it can be any of the following: <ul style="list-style-type: none">• topics• singleactivity• demo• none
name - String!	A human-readable label for the course format.
has_course_view_page - Boolean!	Flag to identify whether the course format supports the course view page or not.

EXAMPLE

```
{
  "format": "abc123",
  "name": "xyz789",
  "has_course_view_page": true
}
```

Types

core_course_is_user_enrolled_result

Result object for query 'core_course_is_user_enrolled_result'.

Field Name	Description
user_currently_enrolled - Boolean!	Whether the user is enrolled a course.

EXAMPLE

```
{"user_currently_enrolled": true}
```

Types

core_course_module

A single course activity module. Activities are individual pieces of learning within a course.

Field Name	Description
<code>id</code> - core_id!	The id of the course module
<code>idnumber</code> - String	The idnumber of the course module
<div>Arguments</div> <hr/> <code>format</code> - core_format	
<code>instanceid</code> - core_id!	The id in the instance, e.g. the id of the scorm in the scorm table
<code>modtype</code> - String!	The type of the module (assignment, quiz, etc)
<div>Arguments</div> <hr/> <code>format</code> - core_format	
<code>name</code> - String!	The name of the course module
<div>Arguments</div> <hr/> <code>format</code> - core_format	
<code>viewurl</code> - String	The view url of the course module
<div>Arguments</div> <hr/> <code>format</code> - core_format	
<code>completion</code> - String	The completion settings of the course module
<code>completionenabled</code> - Boolean!	Whether completion is enabled or not for the course module
<code>completionstatus</code> - String	The completion status of the current user in the course module
<code>rpl</code> - Boolean	The rpl completion status of the current user in the course module

Field Name	Description
progress - Int	The percentage completion progress of the current user in the course module
available - Boolean!	Whether the course module is available for the current user
availablereason - [String]!	The reason for the course module not being available to the current user
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> format - core_format </div>	
visible - Boolean!	Whether the module is visible to the viewing user
showdescription - Boolean	Whether the course module displays the description or not
description - String	The description of the course module
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> format - core_format </div>	
descriptionformat - core_format	The format of the description field.
gradefinal - Float	The user's grade in the course as an absolute value
grademax - Float	The maximum achievable grade for the course
gradepercentage - Float	The user's grade in the course as a percentage of grademax

EXAMPLE

```
{
  "id": 4,
  "idnumber": "abc123",
  "instanceid": 4,
  "modtype": "abc123",
  "name": "xyz789",
  "viewurl": "xyz789",
  "completion": "abc123",
  "completionenabled": true,
  "completionstatus": "abc123",
  "rpl": true,
  "progress": 987,
  "available": true,
  "availablereason": ["abc123"],
  "visible": true,
  "showdescription": true,
```

```

"description": "abc123",
"descriptionformat": "RAW",
"gradefinal": 123.45,
"grademax": 123.45,
"gradepercentage": 987.65
}

```

Types

core_course_section

A single course section. Courses contain named sections to organise a set of activities.

Field Name	Description
<code>id</code> - core_id!	Course section id
<code>title</code> - String	Title of the course section
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - core_format </div>	
<code>modules</code> - [core_course_module]	The course modules in the section
<code>available</code> - Boolean!	Whether the course section is available to the viewing user
<code>availablereason</code> - [String]	The reason why the course section is not available to the viewing user. It will be empty if the course section is available.
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - core_format </div>	
<code>visible</code> - Boolean!	Whether the course section is visible to the viewing user
<code>summary</code> - String	The summary of the course section
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - core_format </div>	
<code>summaryformat</code> - core_format	

EXAMPLE

```
{
  "id": 4,
  "title": "xyz789",
  "modules": [core_course_module],
  "available": true,
  "availablereason": ["xyz789"],
  "visible": true,
  "summary": "abc123",
  "summaryformat": "RAW"
}
```

Types

core_date

Represents standard unix timestamps stored in database.

For input, use either an integer timestamp, numeric string or an ISO-8601 string.

Output can usually be selected via format argument. See `core_date_format` enum for the list of supported formats.

EXAMPLE

```
"2022-04-17"
```

Types

core_date_format

List of supported output formats for `core_date` scalar.

The actual format used for `userdate()` is: `get_string('strftime' . strtolower($arg['format']), 'langconfig')`

Enum Value	Description
TIMESTAMP	A standard UNIX timestamp value. 0 will be replaced with null.
ISO8601	ISO-8601 time string in current user timezone, null if no date. Example output: 2022-09-01T12:06:53+1200
DAYDATETIME	Datetime string in the format: Thursday, 1 September 2022, 12:05 PM
TIME	Time string in the format: 12:08 PM
TIMESHORT	Time string in the format: 12:08

Enum Value	Description
DATE	Date string in the format: 1 September 2022
DATESHORT	Date string in the format: 1 September
DATELONG	Date string in the format: 1/09/2022
DATETIME	Datetime string in the format: 1 September 2022, 12:12 PM
DATETIMESHORT	Datetime string in the format: 1/09/22, 12:12
DATETIMELONG	Datetime string in the format: 1/09/2022, 12:10
DATETIMESECONDS	Datetime string in the format: 1 Sep 2022 at 12:12:07

EXAMPLE

"TIMESTAMP"

Types

core_format

Rich-text output format. Note that some rich-text formats may not be renderable in other formats.

Enum Value	Description
RAW	Outputs the stored value unchanged. Most commonly used when editing a value.
HTML	Outputs the stored value for display within an HTML page. Handles escaping of HTML entities.
PLAIN	Outputs the stored value in a way suitable for plain-text display.
MARKDOWN	Outputs the stored value in Markdown format.
JSON_EDITOR	Outputs the stored value in a JSON structure, suitable for use by a compatible JSON editor (e.g. Weka).
MOBILE	Outputs the stored value in a format suitable for use within the Totara mobile app.

EXAMPLE

"RAW"

Types

core_id

Represents database id fields (numeric string). 0 is converted to NULL.

EXAMPLE

```
4
```

Types

core_pageable_result

Represents a result that is broken up into 'pages' returning a subset of the data, an overall total record count plus the next cursor containing the information needed to load the next page.

Field Name	Description
<code>total</code> - Int!	Total number of records returned by the request (not the number returned in this result/page, but the overall count).
<code>next_cursor</code> - String!	Opaque string containing information to allow the system to identify the next set of results that should be returned after this one. This value can be passed into a query that uses <code>core_pagination_input</code> to fetch the next page.

core_pageable_result Types

[core_cohort_static_cohorts_result](#)

[core_course_courses_result](#)

[core_user_users_result](#)

[totara_job_job_assignments_result](#)

[hierarchy_organisation_organisations_result](#)

[hierarchy_position_positions_result](#)

EXAMPLE

```
{"total": 84, "next_cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319"}
```

Types

core_pagination_input

Contains pagination information needed to load the next page of a query. Note this input type supports both cursor-based and offset-based (page-based) pagination. The method to use will depend on the implementation by the query.

Input Field	Description
cursor - <code>String</code>	Next cursor. Opaque string obtained from the <code>next_cursor</code> property of <code>core_pageable_result</code> . Used by the query to identify which records to return.
limit - <code>param_integer</code>	Number of items to return.
page - <code>param_integer</code>	Page number (only used by offset-based cursors).

EXAMPLE

```
{"cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319", "limit": 20, "page": 1}
```

Types

core_role

Role type object, containing information about a role.

Field Name	Description
id - <code>core_id!</code>	Internal database id of the role.
name - <code>String</code>	Name of a role.
shortname - <code>String!</code>	Shortname of a role.
description - <code>String</code>	Description of a role.

EXAMPLE

```
{  
  "id": 9,  
  "name": "student",  
  "shortname": "student",  
  "description": "It's description of a student role."  
}
```

Types

core_role_role_reference

Input for identifying a role.

The role must be specified by providing one of the following:

- The role's internal database id
- The role's shortname

A role reference must uniquely identify a single role to be valid.

Input Field	Description
id - <code>core_id</code>	The internal id for the role
shortname - <code>String</code>	The shortname for the role

EXAMPLE

```
{"id": 5, "shortname": "learner"}
```

Types

core_sort_direction_enum

Options available when specifying the direction of a sort.

Enum Value	Description
ASC	Sort column in ascending order. This refers to 0, 1, 2, 3 for numbers and A, B, C, D for strings.
DESC	Sort column in descending order. This refers to 3, 2, 1, 0 for numbers and D, C, B, A for strings.

EXAMPLE

```
"ASC"
```

Types

core_sort_input

Input type for specifying sorting of arbitrary data.

Input Field	Description
column - String!	String containing either a single field or a comma-separated list of fields to be sorted by. Allowed fields are defined in the specific data provider and should be documented in the query, but typically column options will align with existing type field names.
direction - core_sort_direction_enum	The sort direction of the column or columns.

EXAMPLE

```
{"column": "timemodified", "direction": "ASC"}
```

Types

param_alphaext

Input parameter equivalent to PARAM_ALPHAEXT.

EXAMPLE

```
"Ab_cd-e"
```

Types

param_alphanumext

Input parameter equivalent to PARAM_ALPHANUMEXT.

EXAMPLE

```
"Abc_123-0"
```

Types

param_email

Input parameter equivalent to PARAM_EMAIL, " is converted to NULL.

EXAMPLE

```
"test@example.com"
```

Types

param_integer

Input parameter equivalent to PARAM_INT, " is converted to NULL.

```
EXAMPLE
```

```
123
```

Types

param_text

Input parameter equivalent to PARAM_TEXT.

```
EXAMPLE
```

```
"This is a string"
```

Types

param_username

Input parameter equivalent to PARAM_USERNAME, " is converted to NULL.

```
EXAMPLE
```

```
"testuser"
```

Audience

Queries

core_cohort_is_user_member

Return result that indicates the user is the member of the audience.

Response

Returns a [core_cohort_is_user_member_result!](#)

Arguments

Name	Description
user - core_user_user_reference!	Target user who is the member of the audience.
cohort - core_cohort_cohort_reference!	Target audience.

QUERY

```
query core_cohort_is_user_member(  
  $user: core_user_user_reference!,  
  $cohort: core_cohort_cohort_reference!  
) {  
  core_cohort_is_user_member(  
    user: $user,  
    cohort: $cohort  
  ) {  
    user_is_member  
  }  
}
```

VARIABLES

```
{  
  "user": core_user_user_reference,  
  "cohort": core_cohort_cohort_reference  
}
```

RESPONSE

```
{"data": {"core_cohort_is_user_member": {"user_is_member": true}}}
```

Queries

core_cohort_static_cohorts

Return a paginated list of static audiences in the system.

Response

Returns a [core_cohort_static_cohorts_result!](#)

Arguments

Name	Description
query - core_cohort_static_cohorts_query	Sort and pagination information to control the data returned.

QUERY

```
query core_cohort_static_cohorts($query: core_cohort_static_cohorts_query!)
  core_cohort_static_cohorts(query: $query) {
    items {
      ...core_cohortFragment
    }
    total
    next_cursor
  }
}
```

VARIABLES

```
{"query": core_cohort_static_cohorts_query}
```

RESPONSE

```
{
  "data": {
    "core_cohort_static_cohorts": {
      "items": [
        {
          "id": "3",
          "name": "Avengers",
          "description": "Description",
          "idnumber": "AUD0001",
          "type": "STATIC",
          "members_count": 2,
          "tags": {"rawname": "Education", "name": "education"}
        },
        {
          "id": "4",
          "name": "Math team",
          "description": "Math team description",
          "idnumber": "AUD0002",
          "type": "STATIC",
          "members_count": 3,
          "tags": {"rawname": "Education", "name": "education"}
        }
      ],
      "total": 47,
      "next_cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319"
    }
  }
}
```

Mutations

core_cohort_add_cohort_member

Add a given user to a target audience.

Response

Returns a `core_cohort_add_cohort_member_result!`

Arguments

Name	Description
<code>user</code> - <code>core_user_user_reference!</code>	Target user who will be added to the audience.
<code>cohort</code> - <code>core_cohort_cohort_reference!</code>	The audience to which the user should be added.

QUERY

```
mutation core_cohort_add_cohort_member(  
  $user: core_user_user_reference!,  
  $cohort: core_cohort_cohort_reference!  
) {  
  core_cohort_add_cohort_member(  
    user: $user,  
    cohort: $cohort  
  ) {  
    user {  
      ...core_userFragment  
    }  
    cohort {  
      ...core_cohortFragment  
    }  
    success  
    was_already_member  
  }  
}
```

VARIABLES

```
{  
  "user": core_user_user_reference,  
  "cohort": core_cohort_cohort_reference  
}
```

RESPONSE

```
{
  "data": {
    "core_cohort_add_cohort_member": {
      "user": core_user,
      "cohort": core_cohort,
      "success": true,
      "was_already_member": true
    }
  }
}
```

Mutations

core_cohort_remove_cohort_member

Remove a given member from a target audience.

Response

Returns a [core_cohort_remove_cohort_member_result!](#)

Arguments

Name	Description
user - core_user_user_reference!	Target user who will be removed from the audience.
cohort - core_cohort_cohort_reference!	The audience from which to remove the user.

QUERY

```
mutation core_cohort_remove_cohort_member(
  $user: core_user_user_reference!,
  $cohort: core_cohort_cohort_reference!
) {
  core_cohort_remove_cohort_member(
    user: $user,
    cohort: $cohort
  ) {
    user {
      ...core_userFragment
    }
    cohort {
      ...core_cohortFragment
    }
    success
    user_was_member
  }
}
```

```
}  
}
```

VARIABLES

```
{  
  "user": core_user_user_reference,  
  "cohort": core_cohort_cohort_reference  
}
```

RESPONSE

```
{  
  "data": {  
    "core_cohort_remove_cohort_member": {  
      "user": core_user,  
      "cohort": core_cohort,  
      "success": true,  
      "user_was_member": true  
    }  
  }  
}
```

Types

core_cohort

Audience type object, containing information about an audience.

Field Name	Description
id - core_id!	Internal database id of the audience.
name - String!	Audience name.
Arguments	
format - core_format	
idnumber - String	The id number of the audience.
Arguments	
format - core_format	
description - String	Audience description.

Field Name	Description
Arguments	
format - core_format	
type - core_cohort_type!	Audience type.
active - Boolean!	Whether the audience is active.
members_count - Int!	The number of audience members.
tags - [core_tag]!	The tags of static cohorts.

EXAMPLE

```
{
  "id": "2",
  "name": "Math team",
  "idnumber": "AUD0002",
  "description": "Math team description",
  "type": "STATIC",
  "active": true,
  "members_count": "5",
  "tags": [core\_tag]
}
```

Types

core_cohort_add_cohort_member_result

The result of the add_cohort_member mutation. This object contains:

- the user object that was requested
- the audience object to which the user should be added
- whether the operation to add the user to the audience was successful
- whether the user was already a member of the audience Note that if the user was already a member of the selected audience, the 'success' field will be 'False' since the user was not added as a result of this request.

Field Name	Description
user - core_user!	User object for the user that is added to the audience.
cohort - core_cohort!	Audience object.
success - Boolean!	Whether the user is added into audience by this request.
was_already_member - Boolean!	Whether the user was already a member of the audience. If so, this operation will technically have

Field Name	Description
	failed, but the end result is the same as if it had succeeded.

EXAMPLE

```
{
  "user": core_user,
  "cohort": core_cohort,
  "success": true,
  "was_already_member": true
}
```

Types

core_cohort_cohort_reference

Input for identifying a audience.

The user must be specified by providing one of the following:

- The audience's internal database id
- The audience's idnumber

Input Field	Description
id - <code>core_id</code>	Internal database id of the audience.
idnumber - <code>String</code>	The id number of the audience.

EXAMPLE

```
{"id": 5, "idnumber": "idnumber5"}
```

Types

core_cohort_is_user_member_result

Result returned from the core_cohort_is_user_member query.

Field Name	Description
user_is_member - <code>Boolean!</code>	Whether the user is the member of the target audience.

EXAMPLE

```
{"user_is_member": true}
```

Types

core_cohort_remove_cohort_member_result

Result returned from the `core_cohort_remove_cohort_member` mutation. Contains audience information and flags to let you know whether the member was removed from the audience.

Field Name	Description
<code>user</code> - core_user!	User object for the user that is removed from the audience.
<code>cohort</code> - core_cohort!	Audience object.
<code>success</code> - Boolean!	Whether the user was removed from the audience as a result of this request.
<code>user_was_member</code> - Boolean!	Whether the user was the member of the audience. If so, this operation will technically have failed, but the end result is the same as if it had succeeded.

EXAMPLE

```
{  
  "user": core_user,  
  "cohort": core_cohort,  
  "success": true,  
  "user_was_member": true  
}
```

Types

core_cohort_static_cohorts_filter

Input type for filtering static cohorts.

Input Field	Description
<code>tag_filter</code> - String	Raw name of a tag.
<code>since_timecreated</code> - core_date	Unix creation timestamp of tag including an integer timestamp, numeric string or an ISO-8601 string.

Input Field	Description
since_timemodified - core_date	Unix timemodification timestamp of tag including an integer timestamp, numeric string or an ISO-8601 string.

EXAMPLE

```
{
  "tag_filter": "education",
  "since_timecreated": "1696837540",
  "since_timemodified": "1696837540"
}
```

Types

core_cohort_static_cohorts_query

Input type used when querying for a list of audiences. Specifies pagination and sorting information that can impact the structure of the results.

Input Field	Description
pagination - core_pagination_input	Pagination information such as which page to return and the number of results requested.
sort - [core_sort_input!]	The sort order of the query.
filter - core_cohort_static_cohorts_filter	The filter for querying static cohorts.

EXAMPLE

```
{
  "pagination": core_pagination_input,
  "sort": [core_sort_input],
  "filter": core_cohort_static_cohorts_filter
}
```

Types

core_cohort_static_cohorts_result

Result returned from the core_cohort_static_cohorts query. Contains a page of results along with pagination information.

Field Name	Description
items - [core_cohort]!	Array of one page of audiences returned by the query.
total - Int!	Total number of users from this query (across all pages).
next_cursor - String!	Cursor to request the next set of results for this query.

EXAMPLE

```
{
  "items": [
    {
      "id": "3",
      "name": "Avengers",
      "description": "Description",
      "idnumber": "AUD0001",
      "type": "STATIC",
      "members_count": 2,
      "tags": {"rawname": "Education", "name": "education"}
    },
    {
      "id": "4",
      "name": "Math team",
      "description": "Math team description",
      "idnumber": "AUD0002",
      "type": "STATIC",
      "members_count": 3,
      "tags": {"rawname": "Education", "name": "education"}
    }
  ],
  "total": 47,
  "next_cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319"
}
```

Types

core_cohort_type

Constant for the audience type.

Enum Value	Description
STATIC	
DYNAMIC	

EXAMPLE

"STATIC"

Completion tracking

Queries

core_completion_get_course_completion

Query to get course completion data for a user in a course.

Response

Returns a `core_completion_get_course_completion_result!`

Arguments

Name	Description
query - <code>core_completion_get_course_completion_input!</code>	The parameter object for this query. This object requires a user reference and course reference.

QUERY

```
query core_completion_get_course_completion($query: core_completion_get_course_completion_get_course_completion(query: $query) {  
  user {  
    ...core_userFragment  
  }  
  course {  
    ...core_courseFragment  
  }  
  completion {  
    ...core_course_completionFragment  
  }  
}
```

VARIABLES

```
{"query": core_completion_get_course_completion_input}
```

RESPONSE

```
{
  "data": {
    "core_completion_get_course_completion": {
      "user": {"id": 4, "fullname": "John Smith"},
      "course": {"id": 4, "fullname": "Course 1"},
      "completion": {
        "id": 4,
        "status": "5",
        "timecompleted": "1696837540",
        "gradefinal": 6.5
      }
    }
  }
}
```

Types

core_completion_get_course_completion_input

The get_course_completion query request parameter object. This object requires both the course and user reference.

Input Field	Description
user - core_user_user_reference!	The user whose completion record is to be retrieved.
course - core_course_course_reference!	The course to which the completion record belongs.

EXAMPLE

```
{
  "user": {"id": "4", "idnumber": "user123"},
  "course": {"id": "4", "idnumber": "course123"}
}
```

Types

core_completion_get_course_completion_result

The results of a query to retrieve course completion data.

Field Name	Description
<code>user - core_user!</code>	The user who was specified in the query.
<code>course - core_course!</code>	The course that was specified in the query.
<code>completion - core_course_completion</code>	The course completion data for the user in the course, or NULL if no record was found.

EXAMPLE

```
{
  "user": {"id": 4, "fullname": "John Smith"},
  "course": {"id": 4, "fullname": "Course 1"},
  "completion": {
    "id": 4,
    "status": "5",
    "timecompleted": "1696837540",
    "gradefinal": 6.5
  }
}
```

Course Catalogue

Queries

`totara_catalog_filters`

Query to fetch available catalog filters and filter options.

Response

Returns a `totara_catalog_filters_result!`

Arguments

Name	Description
<code>input - totara_catalog_url_query_input</code>	

QUERY

```
query totara_catalog_filters($input: totara_catalog_url_query_input) {
  totara_catalog_filters(input: $input) {
    filters {
      ...totara_catalog_filterFragment
    }
  }
}
```

```
    browse_filter {
      ...totara_catalog_filterFragment
    }
    sort {
      ...totara_catalog_filterFragment
    }
  }
}
```

VARIABLES

```
{"input": totara_catalog_url_query_input}
```

RESPONSE

```
{
  "data": {
    "totara_catalog_filters": {
      "filters": [totara_catalog_filter],
      "browse_filter": totara_catalog_filter,
      "sort": totara_catalog_filter
    }
  }
}
```

Queries

totara_catalog_item_details

Query to get details panel content for a learning item.

Response

Returns a [totara_catalog_item_details_result!](#)

Arguments

Name	Description
------	-------------

input - totara_catalog_item_details_input!	
-------------------------------------------------------------------------------	--

QUERY

```
query totara_catalog_item_details($input: totara_catalog_item_details_input!) {
  totara_catalog_item_details(input: $input) {
    details {
      ...totara_catalog_item_detailsFragment
    }
  }
}
```

```
}  
}  
}
```

VARIABLES

```
{"input": totara_catalog_item_details_input}
```

RESPONSE

```
{  
  "data": {  
    "totara_catalog_item_details": {  
      "details": totara_catalog_item_details  
    }  
  }  
}
```

Queries

totara_catalog_items

Query to fetch a page of catalog learning items.

Response

Returns a [totara_catalog_items_result!](#)

Arguments

Name	Description
------	-------------

input - totara_catalog_url_query_input!	
------------------------------------------------------------	--

QUERY

```
query totara_catalog_items($input: totara_catalog_url_query_input!) {  
  totara_catalog_items(input: $input) {  
    items {  
      ...totara_catalog_itemFragment  
    }  
    limitfrom  
    maxcount  
    filters {  
      ...totara_catalog_filterFragment  
    }  
  }  
}
```

```
    browse_filter {
      ...totara_catalog_filterFragment
    }
    sort {
      ...totara_catalog_filterFragment
    }
    is_filtered
  endofrecords
}
}
```

VARIABLES

```
{"input": totara_catalog_url_query_input}
```

RESPONSE

```
{
  "data": {
    "totara_catalog_items": {
      "items": [totara_catalog_item],
      "limitfrom": "677c3ee51363c",
      "maxcount": 100,
      "filters": [totara_catalog_filter],
      "browse_filter": totara_catalog_filter,
      "sort": totara_catalog_filter,
      "is_filtered": true,
      "endofrecords": true
    }
  }
}
```

Queries

totara_catalog_search_filter_options

Query to look up the top 10 matching options for a specific filter. Useful for filters with a lot of options, to narrow down the number of options presented to the user.

Response

Returns a [totara_catalog_search_filter_options_result!](#)

Arguments

Name	Description
input - totara_catalog_search_filter_options_input!	

QUERY

```
query totara_catalog_search_filter_options($input: totara_catalog_search_
  totara_catalog_search_filter_options(input: $input) {
    filter_options
  }
}
```

VARIABLES

```
{"input": totara_catalog_search_filter_options_input}
```

RESPONSE

```
{
  "data": {
    "totara_catalog_search_filter_options": {
      "filter_options": ["basketball", "football"]
    }
  }
}
```

Queries

`totara_catalog_top_items_grouped`

Query to fetch top catalog learning items grouped by type.

Only ever fetches the first page of items of each type. Additional pages must be fetched using `totara_catalog_items` with the `objecttype` filter set appropriately.

Response

Returns a [totara_catalog_top_items_grouped_result!](#)

Arguments

Name	Description
input - totara_catalog_url_query_input!	

QUERY

```
query totara_catalog_top_items_grouped($input: totara_catalog_url_query_input!) {
  totara_catalog_top_items_grouped(input: $input) {
    groups {
      ...totara_catalog_top_items_groupedFragment
    }
  }
}
```

VARIABLES

```
{"input": totara_catalog_url_query_input}
```

RESPONSE

```
{
  "data": {
    "totara_catalog_top_items_grouped": {
      "groups": [totara_catalog_top_items_grouped]
    }
  }
}
```

Queries

totara_catalog_url_query_validation

Query to validate catalog URL query string or structured query JSON.

Response

Returns a [totara_catalog_url_query_validation_result!](#)

Arguments

Name	Description
------	-------------

<code>input</code> - totara_catalog_url_query_input!	
----------------------------------------------------------------------	--

QUERY

```
query totara_catalog_url_query_validation($input: totara_catalog_url_query_input!) {
  totara_catalog_url_query_validation(input: $input) {
    total_count
  }
}
```

```
    error
    error_message
  }
}
```

VARIABLES

```
{"input": totara_catalog_url_query_input}
```

RESPONSE

```
{
  "data": {
    "totara_catalog_url_query_validation": {
      "total_count": 7,
      "error": true,
      "error_message": "abc123"
    }
  }
}
```

Types

totara_catalog_filter

Catalog filter type, collects a named group of options with a defined key for use in catalog queries.

Field Name	Description
key - String!	Filter key that is catalog url query param eg ? course_type_panel[]=1.
title - String!	Filter title / label.
type - String!	Option type.
options - [totara_catalog_filter_option_union]!	Filter options.

EXAMPLE

```
{
  "key": "catalog_learning_type_panel",
  "title": "Learning type",
  "type": "multi",
```

```
"options": [totara_catalog_option]
}
```

Types

totara_catalog_filter_option_union

Union type allowing for different types of filter option to be used

Union Types

[totara_catalog_option](#)

[totara_catalog_tree_option](#)

EXAMPLE

```
totara_catalog_option
```

Types

totara_catalog_filters_result

Filters query result, lists the active filters and sorting options in the catalog.

Field Name	Description
<code>filters</code> - [totara_catalog_filter]	Active filters.
<code>browse_filter</code> - totara_catalog_filter	Filter selected as "Browse filter".
<code>sort</code> - totara_catalog_filter	Sort option.

EXAMPLE

```
{
  "filters": [totara_catalog_filter],
  "browse_filter": totara_catalog_filter,
  "sort": totara_catalog_filter
}
```

Types

totara_catalog_item

Catalog learning item type, used to display information about an individual learning item in a card.

Field Name	Description
<code>itemid</code> - core_id!	Catalog item database id. This is different from the item's actual course or resource id.
<code>type_label</code> - String!	Type of catalog item (singular), e.g. Course
<code>title</code> - String!	Item title.
<code>featured</code> - Boolean!	'Featured' is enabled or not for this item.
<code>redirecturl</code> - String!	Item URL. This URL may contain additional parameters that redirect the user back to the catalog.
<code>image_enabled</code> - Boolean!	Image is enabled or not for this item.
<code>image</code> - totara_catalog_item_image	Item image.
<code>logo</code> - totara_catalog_item_image	Item logo, if enabled.
<code>objecttype</code> - String!	Item's learning type.
<code>hero_data_text_enabled</code> - Boolean!	Hero data text is enabled or not for this item.
<code>hero_data_text</code> - String	Hero data text.
<code>hero_data_type</code> - String	Hero data type.
<code>hero_data_icon_enabled</code> - Boolean!	Hero data icon is enabled or not for this item.
<code>hero_data_icon</code> - totara_catalog_item_icon	Hero data icon.
<code>progress_bar_enabled</code> - Boolean!	Progress bar is enabled or not for this item.
<code>progress_bar</code> - totara_catalog_item_progress_bar	Progress bar.
<code>description_enabled</code> - Boolean!	Description is enabled or not for this item.
<code>description</code> - String	Item short description.
<code>text_placeholders_enabled</code> - Boolean!	Text placeholders are enabled or not for this item.
<code>text_placeholders</code> - [totara_catalog_text_placeholder]	Text placeholders.
<code>icon_placeholders_enabled</code> - Boolean!	Icon placeholders are enabled or not for this item.
<code>icon_placeholders</code> - [totara_catalog_item_icon]	Icon placeholders.

EXAMPLE

```
{  
  "itemid": 20,
```

```

"type_label": "xyz789",
"title": "Security essentials",
"featured": true,
"redirecturl": "http://localhost/course/view.php?id=115",
"image_enabled": true,
"image": totara_catalog_item_image,
"logo": totara_catalog_item_image,
"objecttype": "course",
"hero_data_text_enabled": true,
"hero_data_text": "hero",
"hero_data_type": "text",
"hero_data_icon_enabled": true,
"hero_data_icon": totara_catalog_item_icon,
"progress_bar_enabled": true,
"progress_bar": totara_catalog_item_progress_bar,
"description_enabled": true,
"description": "<p>The course description</p>",
"text_placeholders_enabled": true,
"text_placeholders": [totara_catalog_text_placeholder],
"icon_placeholders_enabled": true,
"icon_placeholders": [totara_catalog_item_icon]
}

```

Types

totara_catalog_item_details

Catalog learning item details type, used to display detailed information about the learning item in a panel.

Field Name	Description
<code>title</code> - String	Item title.
<code>manage_link</code> - totara_catalog_item_link	Manage link.
<code>details_link</code> - totara_catalog_item_details_link	Details link block.
<code>rich_text</code> - String	Main rich text content.
<div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> Arguments <hr/> <code>format</code> - core_format </div>	
<code>description</code> - String	Item short description.
<code>text_placeholders</code> - [totara_catalog_text_placeholder]	Text placeholders.
<code>icon_placeholders</code> - [totara_catalog_item_icon]	Icon placeholders.

EXAMPLE

```
{
  "title": "Swimming program",
  "manage_link": totara_catalog_item_link,
  "details_link": totara_catalog_item_details_link,
  "rich_text": "<div class=\"text_to_html\">Program summary here.</div>",
  "description": "abc123",
  "text_placeholders": [totara_catalog_text_placeholder],
  "icon_placeholders": [totara_catalog_item_icon]
}
```

Types

totara_catalog_item_details_input

Catalog learning item details input type.

Input Field	Description
itemid - core_id!	Item id.

EXAMPLE

```
{"itemid": 4}
```

Types

totara_catalog_item_details_link

Details link block in details panel.

Field Name	Description
description - String	Text to display next to button.
button - totara_catalog_item_link	Link button.

EXAMPLE

```
{
  "description": "xyz789",
  "button": totara_catalog_item_link
}
```

Types

totara_catalog_item_details_result

Catalog learning item details result.

Field Name	Description
details - totara_catalog_item_details!	Catalog item details.

EXAMPLE

```
{"details": totara_catalog_item_details}
```

Types

totara_catalog_item_icon

Catalog learning item icon type.

Field Name	Description
url - String	Icon image URL.
alt - String	Icon alt text.
icon - String	Icon name.

EXAMPLE

```
{  
  "url": "http://localhost/theme/image.php/inspire/totara_core/1732046708/  
  "alt": "Icon",  
  "icon": "abc123"  
}
```

Types

totara_catalog_item_image

Catalog learning item image type.

Field Name	Description
url - String!	Image URL.
alt - String!	Image alt text.

EXAMPLE

```
{  
  "url": "http://localhost/pluginfile.php/988/totara_program/images/224/10  
  "alt": "Image"  
}
```

Types

totara_catalog_item_link

Catalog learning item action link type.

Field Name	Description
label - String!	Button text.
url - String!	Action URL.

EXAMPLE

```
{"label": "How to swim", "url": "http://localhost/course/view.php?id=115"}
```

Types

totara_catalog_item_progress_bar

Catalog learning item progress bar type.

Field Name	Description
progress - Float	Learner's progress, out of 100

EXAMPLE

```
{"progress": 987.65}
```

Types

totara_catalog_items_result

Catalog learning items result. Contains a page of learning items which match a query, and metadata about the page and total result set.

Field Name	Description
<code>items</code> - <code>[totara_catalog_item]!</code>	Catalog learning items.
<code>limitfrom</code> - <code>String!</code>	The cursor to use in <code>query_structure</code> input to fetch the next page of items.
<code>maxcount</code> - <code>Int!</code>	The approximate total number of items in the result set. Due to visibility restrictions, this number may not be an accurate total.
<code>filters</code> - <code>[totara_catalog_filter]</code>	Active filters.
<code>browse_filter</code> - <code>totara_catalog_filter</code>	Filter selected as "Browse filter".
<code>sort</code> - <code>totara_catalog_filter</code>	Sort options.
<code>is_filtered</code> - <code>Boolean!</code>	Flag to indicate whether the query has any filters applied or not.
<code>endofrecords</code> - <code>Boolean!</code>	Flag to indicate if there are any more records to load.

EXAMPLE

```
{
  "items": [totara_catalog_item],
  "limitfrom": "677c3ee51363c",
  "maxcount": 100,
  "filters": [totara_catalog_filter],
  "browse_filter": totara_catalog_filter,
  "sort": totara_catalog_filter,
  "is_filtered": true,
  "endofrecords": true
}
```

Types

totara_catalog_option

Generic catalog filter option type.

Field Name	Description
<code>id - String!</code>	Option id.
<code>label - String!</code>	Option label.
<code>active - Boolean!</code>	Option is active or not, in other words, it's selected/default or not.

EXAMPLE

```
{"id": "Courses", "label": "course", "active": true}
```

Types

`totara_catalog_search_filter_options_input`

Search filter options input type.

Input Field	Description
<code>search_term - param_text!</code>	User search term.
<code>filter_key - String!</code>	Filter key.

EXAMPLE

```
{"search_term": "text_to_search", "filter_key": "tag_panel_1"}
```

Types

`totara_catalog_search_filter_options_result`

Search filter options result type.

Field Name	Description
<code>filter_options - [String]!</code>	Filter options.

EXAMPLE

```
{"filter_options": ["basketball", "football"]}
```

Types

totara_catalog_text_placeholder

Catalog learning item text placeholder type.

Field Name	Description
data_exists - Boolean!	Data exists or not.
data - String	Data.
show_label - Boolean	Show label or not.
label - String	label.

EXAMPLE

```
{
  "data_exists": true,
  "data": "abc123",
  "show_label": true,
  "label": "abc123"
}
```

Types

totara_catalog_top_items_grouped

Grouped items type, a collection of learning items and the type associated with them.

Field Name	Description
objecttype - String!	Object type.
items - [totara_catalog_item]!	Group of items.

EXAMPLE

```
{
  "objecttype": "Certification",
  "items": [totara_catalog_item]
}
```

Types

totara_catalog_top_items_grouped_result

Top grouped items result type, used to display catalog learning items grouped by type.

Field Name	Description
groups - [totara_catalog_top_items_grouped]	Item groups.

EXAMPLE

```
{"groups": [totara_catalog_top_items_grouped]}
```

Types

totara_catalog_tree_option

Catalog tree filter option type, used for hierarchical options.

Field Name	Description
id - String!	Node id.
parentid - String	Node parent id, might be null if top-level.
label - String!	Node name.
active - Boolean!	Option is active or not, in other words, it's selected/default or not.

EXAMPLE

```
{  
  "id": 3,  
  "parentid": "abc123",  
  "label": "Miscellaneous",  
  "active": true  
}
```

Types

totara_catalog_url_query_input

Parameters used to define catalog search query.

Use either `query_string` or `query_structure`.

The format of this input matches the URL query parameters used with the Totara 'grid' or 'explore' catalogs, one of which must be enabled to use this API.

Input Field	Description
query_string - String	<p>URL query string. Query string from any existing catalog search can be used here. Use this option for the first page of results only.</p> <p>The list of expected query keys for query_string search:</p> <ul style="list-style-type: none"> • Active filter aliases • objecttype • perpageload • orderbykey
query_structure - String	<p>URL query structure as a JSON blob of key-value pairs. Use this option if you need pagination.</p> <p>The list of expected query keys for query_structure search:</p> <ul style="list-style-type: none"> • Active filter aliases • objecttype • perpageload • orderbykey • limitfrom (pagination cursor; only for pages 2 through n)

EXAMPLE

```
{
  "query_string": "catalog_learning_type_panel[0]=course&course_type_panel[0]=course",
  "query_structure": "{\"catalog_cat_browse\":\"1259\",\"catalog_learning_type_panel[0]\": \"course\"}"
}
```

Types

totara_catalog_url_query_validation_result

URL query validation result type, provides metadata about a given query.

Field Name	Description
total_count - Int!	Total number of learning items.
error - Boolean!	The flag to indicate the URL query has error or not.
error_message - String	Error message.

EXAMPLE

```
{
  "total_count": 7,
  "error": true,
  "error_message": "xyz789"
}
```

Customfields

Types

totara_customfield_definition

A single custom field definition. Represents details about the custom field. For example, identifying information (such as fullname or shortname) as well as defining information (such as the default value and type)

Field Name	Description
<code>id</code> - core_id!	The custom field definition ID.
<code>fullname</code> - String	Custom field full name.
<code>shortname</code> - String	Custom field short name.
<code>type</code> - totara_customfield_type!	The type of custom field. Defined by an enum of available formats the user can choose from when setting up custom fields in Totara.
<code>description</code> - String	Custom field description.
<code>raw_default_value</code> - String	The raw default value of the custom field without any transformation. This is the raw representation of the data. See totara_customfield_type for more information on specific implementations.
<code>default_value</code> - totara_customfield_value_type	Custom field default value.

EXAMPLE

```
{
  "id": 4,
  "fullname": "xyz789",
  "shortname": "abc123",
  "type": "CHECKBOX",
  "description": "abc123",
}
```

```
"raw_default_value": "abc123",
"default_value": totara_customfield_value_type_generic
}
```

Types

`totara_customfield_field`

A single custom field object. This object contains value of the custom field (both the raw value stored in the database, and the value dependant on the custom field type). We also provide information about the custom field through the definition property, including the name, shortname, type and description).

Custom fields are user-defined extra fields in Totara to store additional information not covered by the usual fields a component has.

Custom fields are used in many different components in Totara including courses, seminars and evidence.

Field Name	Description
<code>definition - totara_customfield_definition!</code>	The custom field's definition. This includes identifying information (such as fullname or shortname) as well as defining information (such as the default value and type).
<code>raw_value - String</code>	The raw value of the custom field without any transformation. This is the raw representation of the data. If no value is set for the related item, then <code>definition.raw_default_value</code> is used.
<code>value - totara_customfield_value_type</code>	<p>The value set against the related item. If no value is set for the related item, then <code>definition.default_value</code> is used.</p> <p>For example, if we have a custom field 'Custom Field A' for courses with a default value of 'My default value' and we have a course 'Course 1', but no data has been set for the custom field on the course, Then the value for Course 1 will be 'My default value'. However, if we then edit Course 1 and set the value for Custom Field A for the course to 'My new value', the value returned will be 'My new value'.</p>

EXAMPLE

```
{
  "definition": totara_customfield_definition,
  "raw_value": "xyz789",
}
```

```
"value": totara_customfield_value_type_generic
}
```

Types

totara_customfield_input

Input type representing a single custom field for a particular type. Please note: setting custom fields may require certain capabilities

Input Field

Description

shortname - [String!](#)

The shortname of the custom field.

Note: if the same custom field is provided multiple times in a single request they will be applied in order so the last one will be used.

Represents the value for the custom field. The format of this data will depend on the field type of the field. See below for a list of types that are supported.

Types and their expected inputs:

data - [String](#)

- checkbox: 0 (unchecked) or 1 (checked).
- menu: A string exactly matching one of the menu options (case sensitive).
- text: Plain text.
- datetime: A string in the format YYYY-MM-DD or YYYY-MM-DD-HH-MM-SS (depending on if the field uses the 'Include time?' option). If the field defines a year range, the input will be validated to ensure it falls within that range. If only YYYY-MM-DD is provided for a field where 'Include time?' is enabled, 00:00 will be used for the time. Stored as a unix timestamp.
- integer: A string representing an integer value E.g. 10
- decimal: A string representing a decimal value E.g. 10.59

Setting data to null will delete the existing value for that custom field. This will set the value back to the default value.

Note: We currently do not support file, textarea, url, multiselect, location. If one of this is used, an exception will be thrown stating an unsupported datatype was used.

EXAMPLE

```
{
  "shortname": "abc123",
  "data": "abc123"
}
```

Types

`totara_customfield_type`

Enum of the different types of custom fields. These relate to the different types of custom fields available to the user when creating custom fields in Totara.

Enum Value	Description
CHECKBOX	Checkbox type. The raw value is stored as a 0 (for false) or a 1 (for true).
MULTISELECT	Multiselect type. The raw value defines a string JSON object with hashes of the options as keys. An example of this structure would look like: <pre>{ "an_md5_hash_for_the_option": { "option": "my option value", -- "icon": "", -- The icon display "default": "0", -- Whether the "delete": "0", -- Whether the v } }</pre>
FILE	File types are currently unsupported. A custom field with the type FILE will always return null for value, raw_value, raw_default_value and default_value properties.
DATETIME	Datetime type. The raw value is stored as a UNIX timestamp.
TEXTAREA	Textarea type. The raw value can be different depending on the text editor used.

For example, if Weka (JSON_EDITOR) was used, the raw value for this will be a JSON object in the format the JSON_EDITOR expects. However, if

Enum Value

Description

the Atto editor was used, the raw value may include HTML. These two editors are not the only options for editor. The output is determined on the particular editor used, which varies depending on user preference.

A single editor is also not guaranteed for a single custom field's uses. Depending on a user's editor preferences, different editors may have been used for different uses of the textarea custom field.

For example, let's say there's two courses; CourseA and CourseB and both of these courses have a custom field "MyTextarea", and two users; UserAtto and UserWeka. If UserAtto has the Atto editor set as their editor preference, and UserWeka, Weka., then it's possible for a textarea custom field (MyTextarea) on CourseA to use Atto when first edited by UserAtto and for CourseB to use Weka when edited by UserWeka. However, if UserAtto then edits CourseB, they will see and use Weka, and vice versa. This is a limitation of the textarea custom field.

URL

URL type. The raw value can either be a JSON value or a plain URL.

- The raw value will be a JSON string when the custom field's value has been set on the item (e.g. the custom field has been set on a course).
- The raw value will be a plain text URL when the value has not been set on the item, and is following the custom field definition's default value.
- The raw default value will always be a plain text URL, so long as a value is set.

LOCATION

Location type. For type location, the default value (including raw_default_value) will be null, regardless of configuration.

The raw value is a JSON object representing both location and display properties. An example is provided below.

```
{
  "address": "186 Willis Street", -- A display pr
  "size": "medium", -- A display prope
  "view": "map", -- A display property
  "zoom": 12, -- A display property
```

Enum Value	Description
	<pre>"display": "address", -- A displa "coordinates": { -- The geographi "latitude": -41.2917599, "longitude": 174.7704544 }</pre>
MENU	Menu type. The type allows for a single choice of many values for the custom field. The raw value will be the value selected.
TEXT	Text type.
INTEGER	Integer type.
DECIMAL	Decimal type.
UNKNOWN	Unknown type. This type is used for types which aren't defined in the tye totara_customfield_type enum.

EXAMPLE

"CHECKBOX"

Types

totara_customfield_value_type

Defines the different supported types available for custom fields. If a specific type has not been defined for custom fields, the totara_customfield_type_generic type will be used. When requesting fields on types used in a union it is a good idea to use aliases for the properties (e.g. value is a field on a datetime type and on a text type, specifying an alias for the field will help ensure you get back the expected field for the expected type).

Note: custom fields with type 'TEXTAREA' will utilise the totara_customfield_type_generic type due to there currently being a lack of support for formatting editor content in Totara.

Example of use in a query:

```
custom_fields {
  value {
    __typename
    ... on totara_customfield_value_type_generic {
      raw_value
    }
    ... on totara_customfield_value_type_multiselect {
```

```

        options {
            option
        }
    }
    ...on totara_customfield_value_type_datetime {
        datetime_value: value (format: ISO8601)
    }
    ... on totara_customfield_value_type_text {
        text_value: value
    }
}
}

```

Union Types

[totara_customfield_value_type_generic](#)
[totara_customfield_value_type_checkbox](#)
[totara_customfield_value_type_text](#)
[totara_customfield_value_type_multiselect](#)
[totara_customfield_value_type_datetime](#)
[totara_customfield_value_type_integer](#)
[totara_customfield_value_type_decimal](#)
[totara_customfield_value_type_location](#)
[totara_customfield_value_type_url](#)
[totara_customfield_value_type_menu](#)

EXAMPLE

```
totara_customfield_value_type_generic
```

Types

totara_customfield_value_type_checkbox

The checkbox custom field type.

Field Name

Description

checked - [Boolean](#)

Whether or not the option is checked.

EXAMPLE

```
{"checked": true}
```

Types

totara_customfield_value_type_datetime

The datetime custom field type.

Field Name	Description
value - core_date	Formatted value of the date time value. Raw value is stored as a UNIX timestamp.
	Formats as <code>TIMESTAMP</code> by default.

Arguments

format - [core_date_format](#)

EXAMPLE

```
{"value": "2022-04-17"}
```

Types

totara_customfield_value_type_decimal

The decimal custom field type.

Field Name	Description
value - Float	The value of the decimal.

EXAMPLE

```
{"value": 987.65}
```

Types

totara_customfield_value_type_generic

Generic custom field type. Used to handle unknown custom field types.

Field Name	Description
raw_value - String	The raw value of the custom field. Raw string representation of the data stored.

EXAMPLE

```
{"raw_value": "xyz789"}
```

Types

totara_customfield_value_type_integer

The integer custom field type.

Field Name	Description
value - Int	The value of the integer.

EXAMPLE

```
{"value": 123}
```

Types

totara_customfield_value_type_location

The location custom field type.

For type location, the default value (including raw_default_value) will be null, regardless of configuration.

Field Name	Description
address - String	A user editable textarea field specifying an address.
zoom - Int	A display property for Google maps. Defines the level of zoom when displaying Google maps.
size - totara_customfield_value_type_location_size	A display property for Google maps. Defines the size of the Google maps canvas.
view - totara_customfield_value_type_location_view	A display property for Google maps. Defines the type of view Google maps should display.

Field Name	Description
<code>display - totara_customfield_value_type_location_display</code>	A display property for the custom field. Defines which fields to show for the custom field (maps, address or both).
<code>coordinates - totara_customfield_value_type_location_coordinates</code>	The geographical coordinates (longitude and latitude) for a given location.

EXAMPLE

```
{
  "address": "xyz789",
  "zoom": 987,
  "size": "SMALL",
  "view": "MAP",
  "display": "BOTH",
  "coordinates": totara_customfield_value_type_location_coordinates
}
```

Types

`totara_customfield_value_type_location_coordinates`

Defines the geographical coordinates for the location custom field.

Field Name	Description
<code>longitude - Float</code>	Longitude is a geographic coordinate that specifies the east–west position of a point on the surface of the Earth
<code>latitude - Float</code>	Latitude is a geographic coordinate that specifies the north-south position of a point on the surface of the Earth

EXAMPLE

```
{"longitude": 987.65, "latitude": 123.45}
```

Types

`totara_customfield_value_type_location_display`

Defines the content displayed for the location custom field.

Enum Value	Description
BOTH	
MAP	
ADDRESS	

EXAMPLE

"BOTH"

Types

totara_customfield_value_type_location_size

Defines the Google maps display size for the location custom field.

Enum Value	Description
SMALL	
MEDIUM	
LARGE	

EXAMPLE

"SMALL "

Types

totara_customfield_value_type_location_view

Defines the Google maps view options for the location custom field.

Enum Value	Description
MAP	
SATELLITE	
HYBRID	

EXAMPLE

"MAP"

Types

totara_customfield_value_type_menu

The menu custom field type.

Field Name	Description
option - String	The value of the option selected.

EXAMPLE

```
{"option": "xyz789"}
```

Types

totara_customfield_value_type_multiselect

Multiselect custom field.

- When options have been selected for a particular item, this will return an array of the selected options.
- For default values, this will return null due to a limitation of how multiselect stores default options.

Field Name	Description
options - [totara_customfield_value_type_multiselect_option]	The options for the multiselect field.

EXAMPLE

```
{  
  "options": [  
    totara_customfield_value_type_multiselect_option  
  ]  
}
```

Types

totara_customfield_value_type_multiselect_option

A single option value for a multiselect.

Field Name	Description
option - String	The value of the option.

EXAMPLE

```
{"option": "abc123"}
```

Types

totara_customfield_value_type_text

The text custom field type.

Field Name	Description
value - String	The value of the text field.

EXAMPLE

```
{"value": "abc123"}
```

Types

totara_customfield_value_type_url

The URL custom field type.

Field Name	Description
url - String	The URL value of the custom field.
text - String	The text value of the custom field.
text - String	Due to a limitation of how this value is stored on custom field definitions, this returns null for default values and values which use the default.

EXAMPLE

```
{  
  "url": "xyz789",  
  "text": "xyz789"  
}
```

Enrol

Types

core_enrol

Represents an enrolment instance.

Field Name	Description
id - core_id!	The unique identifier for the enrolment instance.
enrol - String!	The enrolment method or type as a string.
status - core_enrol_status_enum!	The status of the enrolment instance.
course - core_course!	The course associated with the enrolment instance.
sortorder - Int!	The sort order of the enrolment instance.

EXAMPLE

```
{
  "id": 9,
  "enrol": "totara_program",
  "status": "ENROL_INSTANCE_ENABLED",
  "course": {"id": 2, "idnumber": "abc"},
  "sortorder": "3"
}
```

Types

core_enrol_status_enum

Enumerates the possible enrolment status values for an enrolment instance.

Enum Value	Description
ENROL_INSTANCE_ENABLED	The enrolment instance is enabled.
ENROL_INSTANCE_DISABLED	The enrolment instance is disabled.
ENROL_INSTANCE_DELETED	The enrolment instance has been deleted.

EXAMPLE

Hierarchies

Types

totara_hierarchy_framework

Hierarchy framework interface. Defines standard fields available in all hierarchy frameworks. A framework is a collection of hierarchy items arranged in a tree.

Field Name	Description
<code>id</code> - <code>core_id!</code>	Internal database id of the framework.
<code>fullname</code> - <code>String!</code>	Full name of the framework.
<div>Arguments</div> <hr/> <code>format</code> - <code>core_format</code> Output format this field should be formatted for.	

totara_hierarchy_framework Types

[totara_hierarchy_organisation_framework](#)

[totara_hierarchy_position_framework](#)

[hierarchy_organisation_framework](#)

[hierarchy_position_framework](#)

EXAMPLE

```
{
  "id": 7,
  "fullname": "Example hierarchy framework",
  "idnumber": "HFW123",
  "shortname": "example_hierarchy_framework",
  "description": "This is an example hierarchy framework."
}
```

Types

totara_hierarchy_item

Hierarchy item interface. Defines standard fields available in all hierarchy items. A hierarchy item is a single node in a tree-like hierarchy of items.

Field Name	Description
<code>id</code> - core_id!	Internal database id of the hierarchy item.
<code>fullname</code> - String!	Full name of the hierarchy item.
<hr/>	
Arguments	
<hr/>	
<code>format</code> - core_format	
Output format this field should be formatted for.	
<hr/>	
<code>idnumber</code> - String	Unique reference used to represent the hierarchy item across multiple systems.
<hr/>	
<code>shortname</code> - String	The short name of the hierarchy item. Only used as additional information if admin setting 'showhierarchysshortnames' is enabled.

Arguments

`format` - [core_format](#)

Output format this field should be formatted for.

Field Name	Description
<code>description</code> - String	Rich-text description of the hierarchy item.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - core_format</p> <p>Output format this field should be formatted for.</p> </div>	
<code>frameworkid</code> - core_id	The internal database id of the hierarchy framework this item belongs to.
<code>framework</code> - totara_hierarchy_framework	The hierarchy framework this item belongs to.
<code>path</code> - String!	An ordered list of parent database ids for this hierarchy item, indicating the hierarchy of parent items. Includes the current item and separated by forward slashes, e.g. item with id=6 might be: /1/4/6 This is a computed value.
<code>visible</code> - Boolean!	Flag determining whether the hierarchy item is visible to end users or not.
<code>parentid</code> - core_id	The id of this hierarchy item's immediate parent in the framework hierarchy. Set to 0 if this is a top-level item.
<code>parent</code> - totara_hierarchy_item	The parent item in the hierarchy. Null if a top-level item.
<code>children</code> - [totara_hierarchy_item!]!	Collection of hierarchy items that are immediately below this one in the framework hierarchy. Empty if this item has no children.
<code>typeid</code> - core_id	The id of the hierarchy item type, or 0 if this is a generic item.
<code>type</code> - totara_hierarchy_type	Hierarchy item type object.
<code>timemodified</code> - core_date!	Timestamp of when the hierarchy item type was last modified. Available since Totara 19.0.1.

Arguments

`format` - [core_date_format](#)

[totara_hierarchy_item](#) Types

[totara_hierarchy_organisation](#)

[totara_hierarchy_position](#)

[hierarchy_organisation_organisation](#)

[hierarchy_position_position](#)

EXAMPLE

```
{
  "id": 9,
  "fullname": "Example hierarchy item",
  "idnumber": "HI567",
  "shortname": "example_hierarchy_item",
  "description": "This is an example hierarchy item.",
  "frameworkid": 5,
  "framework": {"id": 8, "fullname": "Example hierarchy framework"},
  "path": "/1/2/45",
  "visible": true,
  "parentid": 4,
  "parent": {"id": 16, "fullname": "Example hierarchy item"},
  "children": [
    {"id": 5, "fullname": "Example hierarchy item"},
    {"id": 7, "fullname": "Another example hierarchy item"}
  ],
  "typeid": 14,
  "type": {"id": 3, "fullname": "Example hierarchy type"},
  "timemodified": "2022-04-17"
}
```

Types

totara_hierarchy_organisation

Organisation item object.

The organisation structure defines the regions, departments, groups, areas, or teams that a company has defined.

Field Name	Description
<code>id</code> - <code>core_id!</code>	Internal database id of the organisation.
<code>fullname</code> - <code>String!</code>	Full name of the organisation.
Arguments	
<code>format</code> - <code>core_format</code>	Output format this field should be formatted for.
<code>idnumber</code> - <code>String</code>	Unique reference used to represent the organisation across multiple systems.
<code>shortname</code> - <code>String</code>	The short name of the organisation. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.

Field Name	Description
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
<p>description - String</p>	Rich-text description of the organisation.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for. Please note: RAW format is only available to users with the capability 'totara/hierarchy:updateorganisation' in the system context, as its purpose is for editing.</p>	
<p>frameworkid - core_id</p>	The internal database id of the organisation framework this organisation belongs to. Please note: viewing this organisation's frameworkid requires the capability 'totara/hierarchy:vieworganisationframeworks'
<p>framework - totara_hierarchy_organisation_framework</p>	The organisation framework this organisation belongs to. Please note: viewing this organisation's framework requires the capability 'totara/hierarchy:vieworganisationframeworks'
<p>path - String!</p>	An ordered list of parent database ids for this organisation, indicating the hierarchy of parent items. Includes the current item and separated by forward slashes, e.g. item with id=6 might be: /1/4/6 This is a computed value.
<p>visible - Boolean!</p>	Flag determining whether the organisation is visible to end users or not.
<p>parentid - core_id</p>	The id of this organisation's immediate parent in the framework hierarchy. Set to 0 if this is a top-level organisation.
<p>parent - totara_hierarchy_organisation</p>	The parent organisation in the hierarchy. Null if a top-level item.
<p>children - [totara_hierarchy_organisation!]!</p>	Collection of organisations that are immediately below this one in the framework hierarchy. Empty if this organisation has no children.
<p>typeid - core_id</p>	The id of the organisation type, or 0 if this is a generic organisation.
<p>type - totara_hierarchy_organisation_type</p>	Organisation type object.

Field Name	Description
timemodified - core_date!	Timestamp of when the organisation type was last modified. Available since Totara 19.0.1.

Arguments

format - [core_date_format](#)

EXAMPLE

```
{
  "id": 27,
  "fullname": "Sales and Marketing",
  "idnumber": "SM1",
  "shortname": "sales",
  "description": "Sales and marketing are based in our US office.",
  "frameworkid": 6,
  "framework": {"id": 7, "fullname": "Global company organisation structure"},
  "path": "/4/22/27",
  "visible": true,
  "parentid": 4,
  "parent": {"id": 22, "fullname": "Head Office"},
  "children": [
    {"id": 45, "fullname": "US sales team"},
    {"id": 37, "fullname": "Europe sales team"}
  ],
  "typeid": 7,
  "type": {"id": 3, "fullname": "Organisational units"},
  "timemodified": "2022-04-17"
}
```

Types

totara_hierarchy_organisation_framework

Organisation framework object. An organisation framework is a collection of organisations arranged in a tree.

Field Name	Description
id - core_id!	Internal database id of the organisation framework.
fullname - String!	Full name of the organisation framework.

Arguments

format - [core_format](#)

Field Name	Description
Output format this field should be formatted for.	
idnumber - String	Unique reference used to represent the organisation framework across multiple systems.
shortname - String	Short name of the organisation framework, only used if admin setting 'showhierarchyshortnames' is enabled.

Arguments

format - [core_format](#)

Output format this field should be formatted for.

description - String	Rich-text description of the organisation framework.
---------------------------------------------	------------------------------------------------------

Arguments

format - [core_format](#)

Output format this field should be formatted for.

EXAMPLE

```
{
  "id": 7,
  "fullname": "Global company organisation structure",
  "idnumber": "MAIN1",
  "shortname": "global_structure",
  "description": "This is the main organisation framework for organisation"
}
```

Types

totara_hierarchy_organisation_reference

Input for identifying an organisation.

The organisation must be identified by providing one of the following:

- The organisation's internal database id
- The organisation's idnumber

An organisation reference must uniquely identify a single organisation to be valid.

Input Field	Description
<code>id - core_id</code>	Identify an organisation by internal database id.
<code>idnumber - String</code>	Identify an organisation by its idnumber. idnumber is a unique reference used to represent the organisation across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "org1"}
```

Types

totara_hierarchy_organisation_type

Organisation type object. An organisation type is a user-defined classification which can be assigned to organisations leading to certain custom fields being available on that organisation.

Field Name	Description
<code>id - core_id!</code>	Internal database id of the organisation type.
<code>fullname - String!</code>	Organisation type full name.

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>idnumber - String</code>	Unique reference used to represent the organisation type across multiple systems.
<code>shortname - String</code>	The short name of the organisation type. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>description - String</code>	Rich-text description of the organisation type.
-----------------------------------	-------------------------------------------------

Arguments

`format - core_format`

Output format this field should be formatted for.

EXAMPLE

```
{
  "id": 3,
  "fullname": "Organisational units",
  "idnumber": "ORG1",
  "shortname": "org_units",
  "description": "This type defines organisation units with specific busin
}
```

Types

totara_hierarchy_position

Position item object.

The position structure defines the job roles that a company has defined.

Field Name	Description
<code>id</code> - core_id!	Internal database id of the position.
<code>fullname</code> - String!	Full name of the position.
<div>Arguments</div> <hr/> <code>format</code> - core_format Output format this field should be formatted for.	

Field Name	Description
	Output format this field should be formatted for. Please note: RAW format is only available to users with the capability 'totara/hierarchy:updateorganisation' in the system context, as its purpose is for editing.
frameworkid - core_id	The internal database id of the position framework this position belongs to. Please note: viewing this position's frameworkid requires the capability 'totara/hierarchy:viewpositionframeworks'
framework - totara_hierarchy_position_framework	The position framework this position belongs to. Please note: viewing this position's framework requires the capability 'totara/hierarchy:viewpositionframeworks'
path - String!	An ordered list of parent database ids for this position, indicating the hierarchy of parent items. Includes the current item and separated by forward slashes, e.g. item with id=6 might be: /1/4/6 This is a computed value.
visible - Boolean!	Flag determining whether the position is visible to users or not.
parentid - core_id	The id of this position's immediate parent in the framework hierarchy. Set to 0 if this is a top-level position.
parent - totara_hierarchy_position	The parent position in the hierarchy. Null if a top-level item.
children - [totara_hierarchy_position!]!	Collection of positions that are immediately below this one in the framework hierarchy. Empty if this position has no children.
typeid - core_id	The id of the position type, or 0 if this is a generic position.
type - totara_hierarchy_position_type	Position type object.
timemodified - core_date!	Timestamp of when the position type was last modified. Available since Totara 19.0.1.

Arguments

format - [core_date_format](#)

EXAMPLE

```
{
  "id": 27,
  "fullname": "Senior project manager",
  "idnumber": "PM3",
  "shortname": "senior_project_manager",
```

```

"description": "A grade 3 project manager.",
"frameworkid": 6,
"framework": {"id": 7, "fullname": "Active company roles"},
"path": "/4/22/27",
"visible": true,
"parentid": 4,
"parent": {"id": 22, "fullname": "Project Manager"},
"children": [],
"typeid": 7,
"type": {"id": 3, "fullname": "Managerial roles"},
"timemodified": "2022-04-17"
}

```

Types

totara_hierarchy_position_framework

Position framework object. A position framework is a collection of positions arranged in a tree.

Field Name	Description
<code>id</code> - <code>core_id!</code>	Internal database id of the position framework.
<code>fullname</code> - <code>String!</code>	Full name of the position framework.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - <code>core_format</code></p> <p>Output format this field should be formatted for.</p> </div>	
<code>idnumber</code> - <code>String</code>	Unique reference used to represent the position framework across multiple systems.
<code>shortname</code> - <code>String</code>	Short name of the position framework, only used if admin setting 'showhierarchysortnames' is enabled.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - <code>core_format</code></p> <p>Output format this field should be formatted for.</p> </div>	
<code>description</code> - <code>String</code>	Rich-text description of the position framework.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - <code>core_format</code></p> <p>Output format this field should be formatted for.</p> </div>	

Field Name	Description
<code>positions - [totara_hierarchy_position!]</code>	Positions that belong to this position framework.

EXAMPLE

```
{
  "id": 7,
  "fullname": "Active company roles",
  "idnumber": "ACTIVEROLES",
  "shortname": "active_roles",
  "description": "This is the main position framework for active positions",
  "positions": [totara_hierarchy_position]
}
```

Types

`totara_hierarchy_position_reference`

Input for identifying a position.

The position must be identified by providing one of the following:

- The position's internal database id
- The position's idnumber

A position reference must uniquely identify a single position to be valid.

Input Field	Description
<code>id - core_id</code>	Identify a position by internal database id.
<code>idnumber - String</code>	Identify a position by its idnumber. idnumber is a unique reference used to represent the position across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "pos1"}
```

Types

`totara_hierarchy_position_type`

Position type object. A position type is a user-defined classification which can be assigned to positions leading to certain custom fields being available on that position.

Field Name	Description
<code>id - core_id!</code>	Internal database id of the position type.
<code>fullname - String!</code>	Position type full name.

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>idnumber - String</code>	Unique reference used to represent the position type across multiple systems.
<code>shortname - String</code>	The short name of the position type. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>description - String</code>	Rich-text description of the position type.
---------------------------------------------------	---------------------------------------------

Arguments

`format - core_format`

Output format this field should be formatted for.

EXAMPLE

```
{
  "id": 3,
  "fullname": "Managerial roles",
  "idnumber": "MAN1",
  "shortname": "manager_roles",
  "description": "This type defines positions with line management respons"
}
```

Types

`totara_hierarchy_type`

Hierarchy type interface. Defines standard fields available in all hierarchy types. A hierarchy type is a user-defined classification which can be assigned to hierarchy items leading to certain custom fields being available on that item.

Field Name	Description
<code>id - core_id!</code>	Internal database id of the hierarchy type.
<code>fullname - String!</code>	Type full name.
<div>Arguments</div> <hr/> <code>format - core_format</code> Output format this field should be formatted for.	
<code>idnumber - String</code>	Unique reference used to represent the type across multiple systems.
<code>shortname - String</code>	The short name of the type. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.
<div>Arguments</div> <hr/> <code>format - core_format</code> Output format this field should be formatted for.	
<code>description - String</code>	Rich-text description of the type.
<div>Arguments</div> <hr/> <code>format - core_format</code> Output format this field should be formatted for.	

`totara_hierarchy_type` Types

[totara_hierarchy_organisation_type](#)

[totara_hierarchy_position_type](#)

[hierarchy_organisation_type](#)

[hierarchy_position_type](#)

EXAMPLE

```
{  
  "id": 3,
```

```
"fullname": "Example hierarchy type",
"idnumber": "HTY123",
"shortname": "example_hierarchy_type",
"description": "This is an example hierarchy type."
}
```

IntelliData

Queries

bi_intellidata_data_files

Returns the result of a data files schema query.

Response

Returns a [bi_intellidata_data_files_result!](#)

Arguments

Name	Description
------	-------------

input - bi_intellidata_data_files_input	
------------------------------------------------------------	--

QUERY

```
query bi_intellidata_data_files($input: bi_intellidata_data_files_input) {
  bi_intellidata_data_files(input: $input) {
    items {
      ...bi_intellidata_data_files_fileFragment
    }
    metadata {
      ...bi_intellidata_data_files_metadataFragment
    }
  }
}
```

VARIABLES

```
{"input": bi_intellidata_data_files_input}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_data_files": {
      "items": [bi_intellidata_data_files_file],
      "metadata": bi_intellidata_data_files_metadata
    }
  }
}
```

Queries

bi_intellidata_dbschema_custom

Returns the result of the custom database schema query.

Response

Returns a [bi_intellidata_dbschema_custom_result!](#)

QUERY

```
query bi_intellidata_dbschema_custom {
  bi_intellidata_dbschema_custom {
    items {
      ...bi_intellidata_dbschema_custom_tableFragment
    }
  }
}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_dbschema_custom": {
      "items": [bi_intellidata_dbschema_custom_table]
    }
  }
}
```

Queries

bi_intellidata_dbschema_unified

Returns the fields schema for unified tables, i.e. the list and fields structure for all Required Tables.

Response

Returns a [bi_intellidata_dbschema_unified_result!](#)

QUERY

```
query bi_intellidata_dbschema_unified {
  bi_intellidata_dbschema_unified {
    items {
      ...bi_intellidata_dbschema_unified_tableFragment
    }
  }
}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_dbschema_unified": {
      "items": [bi_intellidata_dbschema_unified_table]
    }
  }
}
```

Queries

bi_intellidata_export_logs

Fetches the intellidata plugin export logs information, i.e. for the migration / export logs.

Response

Returns a [bi_intellidata_export_logs_result!](#)

QUERY

```
query bi_intellidata_export_logs {
  bi_intellidata_export_logs {
    items {
      ...bi_intellidata_export_logs_logFragment
    }
  }
}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_export_logs": {
```

```
    "items": [bi_intellidata_export_logs_log]
  }
}
```

Queries

bi_intellidata_live_data

Returns the results of a query for live data records.

Response

Returns a [bi_intellidata_live_data_result!](#)

Arguments

Name	Description
<code>input</code> - bi_intellidata_live_data_input!	Fetches live data for a specific datatype. Allowed datatypes are: 'roles', 'categories', 'courses'.

QUERY

```
query bi_intellidata_live_data($input: bi_intellidata_live_data_input!) {
  bi_intellidata_live_data(input: $input) {
    data
  }
}
```

VARIABLES

```
{"input": bi_intellidata_live_data_input}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_live_data": {
      "data": "xyz789"
    }
  }
}
```

Queries

bi_intellidata_plugin_config

Fetches Totara and plugin version and configuration information.

Response

Returns a [bi_intellidata_plugin_config_result!](#)

QUERY

```
query bi_intellidata_plugin_config {
  bi_intellidata_plugin_config {
    moodleconfig {
      ...bi_intellidata_moodleconfigFragment
    }
    cronconfig {
      ...bi_intellidata_cronconfigFragment
    }
    pluginconfig {
      ...bi_intellidata_pluginconfigFragment
    }
    pluginversion
  }
}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_plugin_config": {
      "moodleconfig": bi_intellidata_moodleconfig,
      "cronconfig": [bi_intellidata_cronconfig],
      "pluginconfig": bi_intellidata_pluginconfig,
      "pluginversion": "2023010612"
    }
  }
}
```

Queries

bi_intellidata_validate_credentials

Validates IntelliBoard keys and setup.

Response

Returns a [bi_intellidata_validate_credentials_result!](#)

Arguments

Name	Description
input - bi_intellidata_validate_credentials_input!	The input data for validating credentials.

QUERY

```
query bi_intellidata_validate_credentials($input: bi_intellidata_validate_credentials_validate_credentials(input: $input) {  
  data  
}
```

VARIABLES

```
{"input": bi_intellidata_validate_credentials_input}
```

RESPONSE

```
{"data": {"bi_intellidata_validate_credentials": {"data": "ok"}}
```

Mutations

bi_intellidata_enable_processing

This endpoint is for enabling/disabling Data processing on the Totara side via the IntelliBoard plugin.

Response

Returns a [bi_intellidata_enable_processing_result!](#)

Arguments

Name	Description
input - bi_intellidata_enable_processing_input!	The input data for the enable_processing request.

QUERY

```
mutation bi_intellidata_enable_processing($input: bi_intellidata_enable_processing_input!) {
  bi_intellidata_enable_processing(input: $input) {
    data
  }
}
```

VARIABLES

```
{"input": bi_intellidata_enable_processing_input}
```

RESPONSE

```
{"data": {"bi_intellidata_enable_processing": {"data": "ok"}}
```

Mutations

bi_intellidata_export_data

This endpoint processes exporting data from the 'temp' location to storage.

Response

Returns a [bi_intellidata_export_data_result!](#)

QUERY

```
mutation bi_intellidata_export_data {
  bi_intellidata_export_data {
    items {
      ...bi_intellidata_data_files_fileFragment
    }
    metadata {
      ...bi_intellidata_data_files_metadataFragment
    }
  }
}
```

RESPONSE

```
{
  "data": {
    "bi_intellidata_export_data": {
      "items": [bi_intellidata_data_files_file],
      "metadata": bi_intellidata_data_files_metadata
    }
  }
}
```

```
}  
}
```

Mutations

bi_intellidata_set_datatype

Adds or updates (resets) new custom table to the export mechanism.

Response

Returns a [bi_intellidata_set_datatype_result!](#)

Arguments

Name	Description
------	-------------

input - bi_intellidata_set_datatype_input!	
-------------------------------------------------------------------------------	--

QUERY

```
mutation bi_intellidata_set_datatype($input: bi_intellidata_set_datatype_!  
  bi_intellidata_set_datatype(input: $input) {  
  data  
  }  
}
```

VARIABLES

```
{"input": bi_intellidata_set_datatype_input}
```

RESPONSE

```
{  
  "data": {  
    "bi_intellidata_set_datatype": {"data": "Datatype successfully added."  
  }  
}
```

Types

bi_intellidata_cronconfig

Cron configure type that maps into table 'task_scheduled'

Field Name	Description
<code>id</code> - <code>Int!</code>	Id
<code>component</code> - <code>String</code>	Component
<code>classname</code> - <code>String</code>	Classname
<code>lastruntime</code> - <code>core_date</code>	Last run time
<div style="border: 1px solid #ccc; padding: 5px;"><p>Arguments</p><hr/><p><code>format</code> - <code>core_date_format</code></p></div>	
<code>nextruntime</code> - <code>core_date</code>	Next run time
<div style="border: 1px solid #ccc; padding: 5px;"><p>Arguments</p><hr/><p><code>format</code> - <code>core_date_format</code></p></div>	
<code>blocking</code> - <code>String</code>	Blocking
<code>minute</code> - <code>String</code>	Minute
<code>hour</code> - <code>String</code>	Hour
<code>day</code> - <code>String</code>	Day
<code>month</code> - <code>String</code>	Month
<code>dayofweek</code> - <code>String</code>	Day of week
<code>faildelay</code> - <code>String</code>	Fail delay
<code>disabled</code> - <code>Int</code>	Disabled
<code>customised</code> - <code>Int</code>	Customised

EXAMPLE

```
{
  "id": 123,
  "component": "bi_intellidata",
  "classname": "\\bi_intellidata\\task\\migration_task",
  "lastruntime": "2022-04-17",
  "nextruntime": "2022-04-17",
  "blocking": "0",
  "minute": "1",
  "hour": "1",
```

```
"day": "*",
"month": "*",
"dayofweek": "*",
"faildelay": "0",
"disabled": "1",
"customised": "0"
}
```

Types

bi_intellidata_data_files_file

Type representing a collection of data files.

Field Name	Description
datatype - String!	Data type filter.
files - [bi_intellidata_data_files_file_instance]!	List of files.
migration_files - [bi_intellidata_data_files_file_instance]!	List of migration files.

EXAMPLE

```
{
  "datatype": "abc123",
  "files": [bi_intellidata_data_files_file_instance],
  "migration_files": [
    bi_intellidata_data_files_file_instance
  ]
}
```

Types

bi_intellidata_data_files_file_instance

Type representing a single data file.

Field Name	Description
filename - String!	File name.
filearea - String!	File area.

Field Name	Description
itemid - Int!	Item ID.
filepath - String!	File path.
mimetype - String	MIME type.
filesize - Int!	File size.
timemodified - core_date	Time last modified.

Arguments

format - **core_date_format**

isexternalfile - Boolean!	Indicates whether the file is external or not.
fileurl - String!	File URL.
repositorytype - String	Repository type, if the file is external.

EXAMPLE

```
{
  "filename": "d19c2ecf20245b4d.zip",
  "filearea": "userlogins",
  "itemid": 123,
  "filepath": "/",
  "mimetype": "application/zip",
  "filesize": 208,
  "timemodified": 1678110847,
  "isexternalfile": true,
  "fileurl": "http://totara/api/pluginfile.php/1/bi_intellidata/userlogins",
  "repositorytype": "xyz789"
}
```

Types

bi_intellidata_data_files_input

Input data for fetching data files.

Input Field	Description
timestart - core_date	Start time.
timeend - core_date	End time.
datatype - String	Data type filter.

EXAMPLE

```
{
  "timestart": "2022-04-17",
  "timeend": "2022-04-17",
  "datatype": "abc123"
}
```

Types

bi_intellidata_data_files_metadata

Metadata type for data files.

Field Name	Description
lastmigrationdate - core_date!	Last migration date.
Arguments	
format - core_date_format	
lastexportdate - core_date!	Last export date.
Arguments	
format - core_date_format	
pluginversion - String!	Plugin version.

EXAMPLE

```
{
  "lastmigrationdate": 1678836325,
  "lastexportdate": 1678110847,
  "pluginversion": 2023010612
}
```

Types

bi_intellidata_data_files_result

Type representing the result of a data_files query.

Field Name	Description
items - [bi_intellidata_data_files_file]!	List of data files.
metadata - bi_intellidata_data_files_metadata!	Metadata for the data files.

EXAMPLE

```
{
  "items": [bi_intellidata_data_files_file],
  "metadata": bi_intellidata_data_files_metadata
}
```

Types

bi_intellidata_dbschema_custom_result

A type that represents the result of a custom database schema query.

Field Name	Description
items - [bi_intellidata_dbschema_custom_table]!	

EXAMPLE

```
{"items": [bi_intellidata_dbschema_custom_table]}
```

Types

bi_intellidata_dbschema_custom_table

A type that represents a custom table in a database schema.

Field Name	Description
name - String!	The name of the table.
fields - [bi_intellidata_dbschema_custom_table_field]!	The fields that the table has.
plugintype - String	The plugin type.
plugin - String	The plugin name.

Field Name	Description
keys - [bi_intellidata_dbschema_custom_table_key]!	The keys that the table has.

EXAMPLE

```
{
  "name": "course_sections",
  "fields": [bi_intellidata_dbschema_custom_table_field],
  "plugintype": "course",
  "plugin": "core",
  "keys": [bi_intellidata_dbschema_custom_table_key]
}
```

Types

bi_intellidata_dbschema_custom_table_field

A type that represents a field in a custom table of a database schema.

Field Name	Description
type - String!	The data type of the field.
name - String!	The name of the field.
has_default - Boolean	Whether the field has default value.
max_length - Int	The maximum length of the field.
primary_key - Boolean	Whether the field is primary key.
not_null - Boolean	Whether the field cannot be null.
auto_increment - Boolean	Whether the field can auto increment.
default_value - String	The default value of the field.

EXAMPLE

```
{
  "type": "text",
  "name": "name",
  "has_default": true,
  "max_length": 20,
  "primary_key": true,
  "not_null": true,
  "auto_increment": true,
}
```

```
"default_value": "xyz789"
}
```

Types

bi_intellidata_dbschema_custom_table_key

A type that represents a table key in a custom table of a database schema.

Field Name	Description
name - <code>String!</code>	The type of key, eg. primary.
fields - <code>[String!]!</code>	The fields that the key is related to.
reftable - <code>String</code>	The table the key refers to
reffields - <code>[String]!</code>	The fields the key refers to

EXAMPLE

```
{"name": "primary", "fields": ["id"], "reftable": "users", "reffields": [
```

Types

bi_intellidata_dbschema_unified_result

A type that represents the result of a unified database schema query.

Field Name	Description
items - <code>[bi_intellidata_dbschema_unified_table]!</code>	

EXAMPLE

```
{"items": [bi_intellidata_dbschema_unified_table]}
```

Types

bi_intellidata_dbschema_unified_table

A type that represents a unified table in a database schema.

Field Name	Description
name - String!	The name of the entity that the table represents.
fields - [bi_intellidata_dbschema_unified_table_field!]!	The fields that the table has.

EXAMPLE

```
{
  "name": "users",
  "fields": [bi_intellidata_dbschema_unified_table_field]
}
```

Types

bi_intellidata_dbschema_unified_table_field

A type that represents a field in a unified table of a database schema.

Field Name	Description
field_name - String!	The name of the field.
type - String!	The data type of the field.
description - String	A description of the field.
default - String	The default value of the field.
null - Boolean	Whether the field can be null.

EXAMPLE

```
{
  "field_name": "username",
  "type": "RAW",
  "description": "Username description",
  "default": "abc123",
  "null": true
}
```

Types

bi_intellidata_enable_processing_input

The parameters to go in a request to enable_processing.

Input Field	Description
status - Int!	1 or 0. 1 - Enable, 0 - Disable.
callbackurl - String	The URL to send a callback request to IntelliBoard Next.

EXAMPLE

```
{"status": 1, "callbackurl": "https://mycallback.com?key=1234"}
```

Types

bi_intellidata_enable_processing_result

The response after a request was made to set the enable/disable status of the IntelliBoard plugin.

Field Name	Description
data - String!	If the enable setting was successful, this should be 'ok'.

EXAMPLE

```
{"data": "ok"}
```

Types

bi_intellidata_export_data_result

Type representing the result of a `export_data` mutation.

Field Name	Description
items - [bi_intellidata_data_files_file]!	List of data files.
metadata - bi_intellidata_data_files_metadata!	Metadata for the data files.

EXAMPLE

```
{
  "items": [bi_intellidata_data_files_file],
}
```

```
"metadata": bi_intellidata_data_files_metadata
}
```

Types

bi_intellidata_export_logs_log

Details about the Intelliboard integration plugin export logs.

Field Name	Description
<code>id</code> - <code>core_id!</code>	ID identifying the <code>bi_intellidata_export_log</code> .
<code>datatype</code> - <code>String</code>	The data source for the export log, usually a table.
<code>last_exported_time</code> - <code>core_date</code>	The most recent time the log was exported.
Arguments	
<code>format</code> - <code>core_date_format</code>	
<code>last_exported_id</code> - <code>core_id</code>	The maximum ID in the data source when the log was last exported.
<code>migrated</code> - <code>Int</code>	The migration status value, i.e. 0 = inprogress, 1 = completed, 2 = pending.
<code>timestart</code> - <code>core_date</code>	The time the migration task began.
Arguments	
<code>format</code> - <code>core_date_format</code>	
<code>recordsmigrated</code> - <code>Int</code>	Count of migrated records.
<code>recordscount</code> - <code>Int</code>	Count of records.
<code>tabletype</code> - <code>Int</code>	The type the plugin assigns the data source / table, i.e. required = 0, optional = 1, logs = 2.
<code>timecreated</code> - <code>core_date</code>	The time the <code>export_log</code> record was created. This may be null and not recorded.
Arguments	
<code>format</code> - <code>core_date_format</code>	
<code>timemodified</code> - <code>core_date</code>	The time the <code>export_log</code> record was altered. This may be null and not recorded.
Arguments	

Field Name	Description
format - core_date_format	
usermodified - Int	The ID of the user who altered the export_log record.

EXAMPLE

```
{
  "id": 4,
  "datatype": "xyz789",
  "last_exported_time": 1678110847,
  "last_exported_id": 12345,
  "migrated": 1,
  "timestart": 1678110847,
  "recordsmigrated": 123,
  "recordscount": 456,
  "tabletype": 1,
  "timecreated": 1678110847,
  "timemodified": 1678110847,
  "usermodified": 9876
}
```

Types

bi_intellidata_export_logs_result

Contains a list of export log record items.

Field Name	Description
items - [bi_intellidata_export_logs_log]!	Export log records.

EXAMPLE

```
{"items": [bi\_intellidata\_export\_logs\_log]}
```

Types

bi_intellidata_live_data_input

The input data for querying by datatype.

Input Field	Description
datatype - String!	The 'data source', usually a database table, for querying. Datatypes allowed are: 'roles', 'categories', 'courses' .

EXAMPLE

```
{"datatype": "abc123"}
```

Types

bi_intellidata_live_data_result

Response type for a live data query.

Field Name	Description
data - String!	A JSON-encoded string representing live data records.

EXAMPLE

```
{"data": "xyz789"}
```

Types

bi_intellidata_moodleconfig

Moodle configure type, containing basic moodle information

Field Name	Description
version - String!	Moodle version
release - String!	Moodle release date
dbtype - String!	Database type
moodleworkplace - Boolean	Moodle workplace exists or not
totaraversion - String!	Totara version

EXAMPLE

```
{
  "version": "2022110800.03",
```

```
"release": "3.4.9 (Build: 20190513)",
"dbtype": "pgsql",
"moodleworkplace": "0",
"totaraversion": "17.3"
}
```

Types

bi_intellidata_plugin_config_result

Plugin config result type, containing moodle config, bi_intellidata config and cron config.

Field Name	Description
moodleconfig - bi_intellidata_moodleconfig!	Moodle configure
cronconfig - [bi_intellidata_cronconfig]!	Cron configure
pluginconfig - bi_intellidata_pluginconfig!	bi_intellidata plugin configure
pluginversion - String!	Plugin version

EXAMPLE

```
{
  "moodleconfig": bi_intellidata_moodleconfig,
  "cronconfig": [bi_intellidata_cronconfig],
  "pluginconfig": bi_intellidata_pluginconfig,
  "pluginversion": "2023010612"
}
```

Types

bi_intellidata_pluginconfig

The Intelliboard plugin's configuration settings.

Field Name	Description
title - String!	Configuration title.
items - [bi_intellidata_pluginconfig_setting!]	Configuration settings.

EXAMPLE

```
{
  "title": "title",
  "items": [bi_intellidata_pluginconfig_setting]
}
```

Types

bi_intellidata_pluginconfig_setting

An Intelliboard plugin configuration setting.

Field Name	Description
type - String!	Type
subtype - String!	Subtype that represents setting, eg. checkbox.
title - String!	Title
name - String!	Name
grouptitle - String	Group title for each setting section name. Note: this could be null.
value - String!	Setting value. If the value is an array it will be encoded into a string. If the subtype is 'checkbox', the value will be 1 or 0.

EXAMPLE

```
{
  "type": "setting",
  "subtype": "select",
  "title": "title",
  "name": "name",
  "grouptitle": "xyz789",
  "value": "standard"
}
```

Types

bi_intellidata_set_datatype_input

Trigger to export files after inserting custom table. 0 = do not process migration, 1 = create an ad-hoc task to process export with the next cron run. Default value is 0.

Input Field	Description
datatypes - [String!]!	Custom table name(s)
exportfiles - Int	Trigger to export files after inserting custom table 0 - do not process migration 1 - create ad-hoc task to process export with next cron run Default value is 0
callbackurl - String	Callback URL to send request when export completed. Working only if not empty and exportfiles = 1

EXAMPLE

```
{
  "datatypes": ["abc123"],
  "exportfiles": 1,
  "callbackurl": "https://mycallback.com?key=1234"
}
```

Types

bi_intellidata_set_datatype_result

The result of attempting to set a custom datatype.

Field Name	Description
data - String!	The operation result message.

EXAMPLE

```
{"data": "Datatype successfully added."}
```

Types

bi_intellidata_validate_credentials_input

The input data required for validating credentials.

Input Field	Description
code - String!	The code containing the client identifier.

EXAMPLE

```
{"code": "xyz789"}
```

Types

bi_intellidata_validate_credentials_result

A type that represents the result of validating credentials, containing data about whether the validation was successful or not.

Field Name	Description
data - String!	The result data from the validation operation.

EXAMPLE

```
{"data": "ok"}
```

Job assignments

Queries

totara_job_job_assignment

Return a single job_assignment from the system. Multitenancy restrictions apply.

Response

Returns a [totara_job_job_assignment_result!](#)

Arguments

Name	Description
target_job - totara_job_job_assignment_reference!	The job assignment that we want to retrieve.

QUERY

```
query totara_job_job_assignment($target_job: totara_job_job_assignment_re
  totara_job_job_assignment(target_job: $target_job) {
    job_assignment {
      ...totara_job_job_assignmentFragment
    }
  }
found
```

```
}  
}
```

VARIABLES

```
{"target_job": totara_job_job_assignment_reference}
```

RESPONSE

```
{  
  "data": {  
    "totara_job_job_assignment": {  
      "job_assignment": totara_job_job_assignment,  
      "found": true  
    }  
  }  
}
```

Queries

totara_job_job_assignments

Return a paginated list of job_assignments in the system. Multitenancy restrictions apply.

Response

Returns a [totara_job_job_assignments_result!](#)

Arguments

Name	Description
------	-------------

query -
[totara_job_job_assignments_query](#)

QUERY

```
query totara_job_job_assignments($query: totara_job_job_assignments_query!)  
  totara_job_job_assignments(query: $query) {  
    items {  
      ...totara_job_job_assignmentFragment  
    }  
    total  
    next_cursor
```

```
}  
}
```

VARIABLES

```
{"query": totara_job_job_assignments_query}
```

RESPONSE

```
{  
  "data": {  
    "totara_job_job_assignments": {  
      "items": [totara_job_job_assignment],  
      "total": 987,  
      "next_cursor": "xyz789"  
    }  
  }  
}
```

Mutations

totara_job_create_job_assignment

Creates a new job assignment.

Response

Returns a `totara_job_create_job_assignment_result!`

Arguments

Name	Description
<code>input</code> - <code>totara_job_create_job_assignment_input!</code>	Input object specifying data for creating the new job assignment.

QUERY

```
mutation totara_job_create_job_assignment($input: totara_job_create_job_assignment_input!) {  
  totara_job_create_job_assignment(input: $input) {  
    job_assignment {  
      ...totara_job_job_assignmentFragment  
    }  
  }  
}
```

```
}  
}
```

VARIABLES

```
{"input": totara_job_create_job_assignment_input}
```

RESPONSE

```
{  
  "data": {  
    "totara_job_create_job_assignment": {  
      "job_assignment": totara_job_job_assignment  
    }  
  }  
}
```

Mutations

totara_job_delete_job_assignment

Mutation to delete a job assignment.

Response

Returns a [totara_job_delete_job_assignment_result!](#)

Arguments

Name	Description
target_job - totara_job_job_assignment_reference!	Input provided for identifying the job assignment.

QUERY

```
mutation totara_job_delete_job_assignment($target_job: totara_job_job_assignment_reference!) {  
  totara_job_delete_job_assignment(target_job: $target_job) {  
    job_assignment_id  
  }  
}
```

VARIABLES

```
{"target_job": totara_job_job_assignment_reference}
```

RESPONSE

```
{"data": {"totara_job_delete_job_assignment": {"job_assignment_id": 4}}}
```

Mutations

totara_job_update_job_assignment

Updates a specific target job assignment with new properties.

Response

Returns a [totara_job_update_job_assignment_result!](#)

Arguments

Name	Description
target_job - totara_job_job_assignment_reference!	The job assignment that should be updated.
input - totara_job_update_job_assignment_input!	Input object specifying data for updating the job assignment.

QUERY

```
mutation totara_job_update_job_assignment(  
  $target_job: totara_job_job_assignment_reference!,  
  $input: totara_job_update_job_assignment_input!  
) {  
  totara_job_update_job_assignment(  
    target_job: $target_job,  
    input: $input  
  ) {  
    job_assignment {  
      ...totara_job_job_assignmentFragment  
    }  
  }  
}
```

VARIABLES

```
{  
  "target_job": totara_job_job_assignment_reference,  
}
```

```
"input": totara_job_update_job_assignment_input
}
```

RESPONSE

```
{
  "data": {
    "totara_job_update_job_assignment": {
      "job_assignment": {
        "id": 7,
        "idnumber": "PM3",
        "fullname": "Senior project manager",
        "user": {"fullname": "John Smith"}
      }
    }
  }
}
```

Types

totara_job_create_job_assignment_input

Input data for creating a job assignment.

Input Field	Description
<code>idnumber</code> - String!	A unique reference used to represent the job assignment across multiple systems. If the reference value is not unique, this can be combined with the <code>job_assignment</code> id and <code>user_id</code> in queries to behave like a unique compound reference value in order to refer to a single job_assignment.
<code>fullname</code> - String	The full name of the job assignment.
<code>shortname</code> - String	The short name of the job assignment.
<code>user</code> - core_user_user_reference!	The user to create the job assignment for.
<code>start_date</code> - core_date	Date that the user started in this job assignment.
<code>end_date</code> - core_date	Date that the user ends in this job assignment.
<code>position</code> - totara_hierarchy_position_reference	Position (job role) that applies to this job assignment.
<code>organisation</code> - totara_hierarchy_organisation_reference	Organisation in which the user holds this job assignment.

Input Field	Description
manager - totara_job_job_assignment_reference	Job assignment that defines the management relationship.
appraiser - core_user_user_reference	Appraiser for this assignment.
temp_manager - totara_job_job_assignment_reference	Job assignment that defines the temporary management relationship.
temp_manager_expiry_date - core_date	Date the temporary manager assignment will expire.

EXAMPLE

```
{
  "idnumber": "abc123",
  "fullname": "abc123",
  "shortname": "abc123",
  "user": core_user_user_reference,
  "start_date": "2022-04-17",
  "end_date": "2022-04-17",
  "position": totara_hierarchy_position_reference,
  "organisation": totara_hierarchy_organisation_reference,
  "manager": totara_job_job_assignment_reference,
  "appraiser": core_user_user_reference,
  "temp_manager": totara_job_job_assignment_reference,
  "temp_manager_expiry_date": "2022-04-17"
}
```

Types

[totara_job_create_job_assignment_result](#)

Mutation result type returned after creating a job assignment. Returns information about the job assignment that was created.

Field Name	Description
job_assignment - totara_job_job_assignment!	The newly created job assignment.

EXAMPLE

```
{"job_assignment": totara_job_job_assignment}
```

Types

totara_job_delete_job_assignment_result

Mutation result type returned after deletion of a job assignment.

Field Name	Description
<code>job_assignment_id - core_id!</code>	Database id of the job assignment.

EXAMPLE

```
{"job_assignment_id": 4}
```

Types

totara_job_job_assignment

Job assignment type object, containing information about an individual job assignment.

Field Name	Description
<code>id - core_id!</code>	Internal database id of the job assignment.
<code>userid - core_id!</code>	User id of the user the job assignment is associated with.
<code>user - core_user!</code>	User record of the user the job assignment is associated with.
<code>fullname - String</code>	The full name of the job assignment. Used when job assignment is displayed and for selecting job assignments in dialogs.

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>shortname - String</code>	The short name of the job assignment. Only used as additional information.
---------------------------------	----------------------------------------------------------------------------

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>idnumber - String!</code>	Used when syncing job assignment data from external sources. Must be unique for each user (but two users could have the same job assignment)
---------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------

Field Name	Description
	idnumber). If the reference value is not unique, this can be combined with the job_assignment id and user_id in queries to behave like a unique compound reference value in order to refer to a single job_assignment.
description - String	Rich-text description of the job assignment.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p> </div>	
	Date that the user started in this job assignment. This date can be used in dynamic audience rules.
startdate - core_date	This date is not used by the system to determine if the job assignment is active.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - core_date_format</p> <p>Output date format this field should be formatted using.</p> </div>	
	Date that the user ends in this job assignment. This date can be used in dynamic audience rules.
enddate - core_date	This date is not used by the system to determine if the job assignment is active.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p>format - core_date_format</p> <p>Output date format this field should be formatted using.</p> </div>	
positionid - core_id	Internal database id of the position for this assignment.
position - totara_hierarchy_position	Position (job role) object for this job assignment.
organisationid - core_id	Internal database id of the organisation for this job assignment.
organisation - totara_hierarchy_organisation	Organisation object for this job assignment.
managerjaid - core_id	Internal database id of the manager job assignment.

Field Name	Description
managerja - totara_job_job_assignment	Job assignment object that defines the management relationship.
tempmanagerjaid - core_id	Internal database id of the temporary manager job assignment. Temporary manager has the same rights as a normal manager, until the expiry date is reached.
tempmanagerja - totara_job_job_assignment	Job assignment object that defines the temporary management relationship.
tempmanagerexpirydate - core_date	Date the temporary manager assignment will expire.

Arguments

format - [core_date_format](#)

Output date format this field should be formatted using.

appraiserid - core_id	Internal database id of the user assigned as the appraiser for this job assignment.
appraiser - core_user	Appraiser user object for this job assignment.
staffcount - Int!	Number of staff assigned to the user via this job assignment. E.g. the number of other job assignments which list this assignment as 'managerja'.
tempstaffcount - Int!	Number of staff assigned to the user to temporarily manage via this job assignment. E.g. the number of other job assignments which list this assignment as 'tempmanagerja'.
timemodified - core_date	The date/time the job assignment was last modified

Arguments

format - [core_date_format](#)

EXAMPLE

```
{
  "id": 4,
  "userid": 4,
  "user": {"username": "joe.smith"},
  "fullname": "Senior Project Manager",
  "shortname": "senior_pm",
  "idnumber": "PM3",
  "description": "Grade 3 project manager.",
}
```

```

"startdate": "2022-04-17",
"enddate": "2022-04-17",
"positionid": 4,
"position": {"idnumber": "pos1"},
"organisationid": 4,
"organisation": {"idnumber": "org2"},
"managerjaid": 4,
"managerja": {"idnumber": "managerjob1"},
"tempmanagerjaid": 4,
"tempmanagerja": {"idnumber": "tempmanagerjob1"},
"tempmanagerexpirydate": "2022-04-17",
"appraiserid": 4,
"appraiser": {"email": "appraiser@example.com"},
"staffcount": 10,
"tempstaffcount": 3,
"timemodified": "2022-04-17"
}

```

Types

totara_job_job_assignment_reference

Input for identifying a job assignment.

The job assignment must be identified by providing one of the following:

- The job assignment's internal database id
- The job assignment's idnumber

A job assignment reference must uniquely identify a single job assignment record to be valid.

Input Field	Description
id - core_id	Identify a job assignment by internal database id.
	Identify a job assignment by its idnumber.
idnumber - String	Idnumber is a unique reference used to represent a job assignment across multiple systems. If the reference value is not unique, this can be combined with the job_assignment id and user_id in queries to behave like a unique compound reference value in order to refer to a single job_assignment.
user - core_user_user_reference	The user who has received the job_assignment.

EXAMPLE

```
{
  "id": 4,
  "idnumber": "job1",
  "user": core_user_user_reference
}
```

Types

totara_job_job_assignment_result

Query result type returned after querying a single job assignment.

Field Name	Description
job_assignment - totara_job_job_assignment	The found job assignment.
found - Boolean!	Whether the job assignment requested is available to the viewing user

EXAMPLE

```
{
  "job_assignment": totara_job_job_assignment,
  "found": true
}
```

Types

totara_job_job_assignments_query

Information on paginating and sorting the results.

Input Field	Description
pagination - core_pagination_input	Pagination information such as which page to return and the number of results requested.
sort - [core_sort_input!]	The sort order of the query. Allowed entity fields for the sort column are 'id', 'userid', 'shortname', 'startdate', 'enddate', 'position', 'organisation', 'managerjaid', 'tempmanagerjaid', 'tempmanagerexpirydate', 'appraiserid', 'staffcount', 'tempstaffcount'.
filters - totara_job_job_assignments_query_filters	An input object for specifying criteria to narrow down query results of job assignments.

EXAMPLE

```
{
  "pagination": core_pagination_input,
  "sort": [core_sort_input],
  "filters": totara_job_job_assignments_query_filters
}
```

Types

totara_job_job_assignments_query_filters

Input Field	Description
<code>since_timemodified</code> - core_date	Filters retrieved job assignments to those modified on or since the given date.

EXAMPLE

```
{"since_timemodified": "2022-04-17"}
```

Types

totara_job_job_assignments_result

Result returned from the `totara_job_job_assignments` query. Contains a page of results along with pagination information.

Field Name	Description
<code>items</code> - [totara_job_job_assignment]	Array of one page of <code>job_assignments</code> returned by the query.
<code>total</code> - Int!	Total number of <code>job_assignments</code> from this query (across all pages).
<code>next_cursor</code> - String!	Cursor to request the next set of results for this query.

EXAMPLE

```
{
  "items": [totara_job_job_assignment],
  "total": 123,
  "next_cursor": "xyz789"
}
```

Types

totara_job_update_job_assignment_input

The input data for updating the job assignment.

Input Field	Description
idnumber - String	Unique reference used to represent the job assignment across multiple systems. If the reference value is not unique, this can be combined with the job_assignment id and user_id in queries to behave like a unique compound reference value in order to refer to a single job_assignment.
fullname - String	The full name of the job assignment.
shortname - String	The short name of the job assignment.
start_date - core_date	Date that the user started in this job assignment.
end_date - core_date	Date that the user ends in this job assignment.
position - totara_hierarchy_position_reference	Position (job role) for the user.
organisation - totara_hierarchy_organisation_reference	Organisation in which the user works.
manager - totara_job_job_assignment_reference	Job assignment that defines the management relationship.
appraiser - core_user_user_reference	Appraiser for this assignment.
temp_manager - totara_job_job_assignment_reference	Job assignment that defines the temporary management relationship.
temp_manager_expiry_date - core_date	Date the temporary manager assignment will expire.

EXAMPLE

```
{
  "idnumber": "PM3",
  "fullname": "Senior Project Manager",
  "shortname": "abc123",
  "start_date": "2020-07-15",
  "end_date": "2025-03-26",
  "position": {"idnumber": "pos1"},
  "organisation": {"idnumber": "org1"},
  "manager": {"idnumber": "managerjob1"},
}
```

```
"appraiser": {"username": "jane.doe"},
"temp_manager": {"idnumber": "tempmanagerjob1"},
"temp_manager_expiry_date": "2025-04-16"
}
```

Types

totara_job_update_job_assignment_result

Mutation result type returned after updating a job assignment. Returns the job assignment that was updated.

Field Name	Description
<code>job_assignment - totara_job_job_assignment!</code>	The updated job assignment.

EXAMPLE

```
{
  "job_assignment": {
    "id": 7,
    "idnumber": "PM3",
    "fullname": "Senior project manager",
    "user": {"fullname": "John Smith"}
  }
}
```

Manual enrolments

Mutations

enrol_manual_enrol_user

Add a manual enrolment into the target course for the target user. The API User role must be able to assign the corresponding role to the enrolled user.

Response

Returns an `enrol_manual_enrol_user_result!`

Arguments

Name	Description
<code>input - enrol_manual_enrol_user_input!</code>	Input object specifying data for enrolling the user.

QUERY

```
mutation enrol_manual_enrol_user($input: enrol_manual_enrol_user_input!) {
  enrol_manual_enrol_user(input: $input) {
    user {
      ...core_userFragment
    }
    course {
      ...core_courseFragment
    }
    role {
      ...core_roleFragment
    }
    enrolment {
      ...core_user_enrolmentFragment
    }
    success
    was_already_enrolled
  }
}
```

VARIABLES

```
{"input": enrol_manual_enrol_user_input}
```

RESPONSE

```
{
  "data": {
    "enrol_manual_enrol_user": {
      "user": {"id": 12, "idnumber": "123"},
      "course": {"id": 22, "idnumber": "222"},
      "role": {"id": 5, "shortname": "learner"},
      "enrolment": {
        "id": 122,
        "timestart": "1698227823",
        "timeend": "1698227824",
        "status": "ENROL_USER_ACTIVE"
      },
      "success": true,
      "was_already_enrolled": true
    }
  }
}
```

Mutations

enrol_manual_unenrol_user

Remove any manual enrolments for the target user from the target course. If multiple manual enrolments (with different roles) exist for the same user, they will all be removed. Does not affect other enrolment types, such as self-enrolment.

Response

Returns an [enrol_manual_unenrol_user_result!](#)

Arguments

Name	Description
input - enrol_manual_unenrol_user_input!	Input object specifying data for unenrolling the user.

QUERY

```
mutation enrol_manual_unenrol_user($input: enrol_manual_unenrol_user_input!){
  enrol_manual_unenrol_user(input: $input) {
    user {
      ...core_userFragment
    }
    course {
      ...core_courseFragment
    }
    success
    user_had_manual_enrolment
  }
}
```

VARIABLES

```
{"input": enrol_manual_unenrol_user_input}
```

RESPONSE

```
{
  "data": {
    "enrol_manual_unenrol_user": {
      "user": {"id": 12, "idnumber": "123"},
      "course": {"id": 22, "idnumber": "222"},
      "success": true,
      "user_had_manual_enrolment": true
    }
  }
}
```

```
}  
}
```

Types

enrol_manual_enrol_user_input

Input required when enrolling a user.

Input Field	Description
user - core_user_user_reference!	The target user to enrol in the course.
course - core_course_course_reference!	The target course to enrol the user in.
role - core_role_role_reference	The role for the enrolment. If no role is provided then the default role for this enrol will be used. In most cases that would be the Learner role.
timestart - core_date	The start date for the enrolment.
timeend - core_date	The end date for the enrolment.
suspended - Boolean	Whether the enrolment is suspended.

EXAMPLE

```
{  
  "user": core_user_user_reference,  
  "course": core_course_course_reference,  
  "role": core_role_role_reference,  
  "timestart": "2022-04-17",  
  "timeend": "2022-04-17",  
  "suspended": true  
}
```

Types

enrol_manual_enrol_user_result

Result object for mutation 'enrol_manual_enrol_user'.

Field Name	Description
user - core_user!	User object for the user that is enrolled into the course.

Field Name	Description
course - core_course!	Course object.
role - core_role!	Role object.
enrolment - core_user_enrolment!	User enrolment object.
success - Boolean!	Whether the enrolment was successful.
was_already_enrolled - Boolean!	Whether the user was already enrolled in the course.

EXAMPLE

```
{
  "user": {"id": 12, "idnumber": "123"},
  "course": {"id": 22, "idnumber": "222"},
  "role": {"id": 5, "shortname": "learner"},
  "enrolment": {
    "id": 122,
    "timestart": "1698227823",
    "timeend": "1698227824",
    "status": "ENROL_USER_ACTIVE"
  },
  "success": true,
  "was_already_enrolled": true
}
```

Types

enrol_manual_unenrol_user_input

Input required when unenrolling a user.

Input Field	Description
user - core_user_user_reference!	Target user who will be unenrolled from the course.
course - core_course_course_reference!	Target course.

EXAMPLE

```
{
  "user": core_user_user_reference,
  "course": core_course_course_reference
}
```

Types

enrol_manual_unenrol_user_result

Result object for mutation enrol_manual_unenrol_user.

Field Name	Description
user - core_user!	User object for the user that is unenrolled from the course.
course - core_course!	Course object.
success - Boolean!	Whether all manual enrolments on this course have been removed for the user. Note that the user may still be enrolled via other methods.
user_had_manual_enrolment - Boolean!	Whether the user had any manual enrolments on the course before this mutation.

EXAMPLE

```
{
  "user": {"id": 12, "idnumber": "123"},
  "course": {"id": 22, "idnumber": "222"},
  "success": true,
  "user_had_manual_enrolment": true
}
```

Multitenancy

Types

totara_tenant_tenant_reference

Input for identifying a tenant.

The tenant must be specified by providing one of the following:

- The tenant's internal database id
- The tenant's idnumber

A tenant reference must uniquely identify a single tenant to be valid.

Input Field	Description
id - core_id	Identify a tenant by the internal database id of the tenant.

Input Field

Description

idnumber - [String](#)

Identify a tenant by the idnumber of the tenant. This is a unique reference used to represent the tenant across multiple systems.

EXAMPLE

```
{"id": 5, "idnumber": "tenant1"}
```

Organisation Hierarchies

Queries

hierarchy_organisation_organisation

Return a specific organisation.

Response

Returns a [hierarchy_organisation_organisation_result!](#)

Arguments

Name

Description

reference -

[hierarchy_organisation_organisation_reference!](#)

Input object specifying data for the existing organisation.

QUERY

```
query hierarchy_organisation_organisation($reference: hierarchy_organisation_organisation_reference!) {
  hierarchy_organisation_organisation(reference: $reference) {
    found
    organisation {
      ...hierarchy_organisation_organisationFragment
    }
  }
}
```

VARIABLES

```
{
  "reference": hierarchy_organisation_organisation_reference
}
```

RESPONSE

```
{
  "data": {
    "hierarchy_organisation_organisation": {
      "found": true,
      "organisation": hierarchy_organisation_organisation
    }
  }
}
```

Queries

hierarchy_organisation_organisations

Query to retrieve organisations.

Response

Returns a [hierarchy_organisation_organisations_result!](#)

Arguments

Name	Description
------	-------------

query - hierarchy_organisation_organisations_query	
-----------------------------------------------------------------------	--

QUERY

```
query hierarchy_organisation_organisations($query: hierarchy_organisation_organisations_query!) {
  hierarchy_organisation_organisations(query: $query) {
    items {
      ...hierarchy_organisation_organisationFragment
    }
    total
    next_cursor
  }
}
```

VARIABLES

```
{"query": hierarchy_organisation_organisations_query}
```

RESPONSE

```
{
  "data": {
    "hierarchy_organisation_organisations": {
      "items": [hierarchy_organisation_organisation],
      "total": 987,
      "next_cursor": "xyz789"
    }
  }
}
```

Mutations

hierarchy_organisation_create_organisation

Creates a new organisation. Please note: creating organisations requires the capability 'totara/hierarchy:createorganisation'

Response

Returns a [hierarchy_organisation_organisation_created_result!](#)

Arguments

Name	Description
<code>input</code> - hierarchy_organisation_create_organisation_input!	Input object specifying data for creating the new organisation.

QUERY

```
mutation hierarchy_organisation_create_organisation($input: hierarchy_organisation_create_organisation_input!) {
  hierarchy_organisation_create_organisation(input: $input) {
    organisation {
      ...hierarchy_organisation_organisationFragment
    }
  }
}
```

VARIABLES

```
{
  "input": hierarchy_organisation_create_organisation_input
}
```

RESPONSE

```
{
  "data": {
    "hierarchy_organisation_create_organisation": {
      "organisation": hierarchy_organisation_organisation
    }
  }
}
```

Mutations

hierarchy_organisation_delete_organisation

Mutation to delete an organisation. Please note: deleting organisations requires the capability 'totara/hierarchy:deleteorganisation'

Response

Returns a [hierarchy_organisation_organisation_deleted_result!](#)

Arguments

Name	Description
target_organisation - hierarchy_organisation_organisation_reference!	Input provided for identifying the organisation.

QUERY

```
mutation hierarchy_organisation_delete_organisation($target_organisation:
  hierarchy_organisation_delete_organisation(target_organisation: $target_
  organisation_id
  )
}
```

VARIABLES

```
{
  "target_organisation": hierarchy_organisation_organisation_reference
}
```

RESPONSE

```
{"data": {"hierarchy_organisation_delete_organisation": {"organisation_id"
```

Mutations

hierarchy_organisation_update_organisation

Updates a specific target organisation with new properties. Please note: updating organisations requires the capability 'totara/hierarchy:updateorganisation'

Response

Returns a [hierarchy_organisation_organisation_updated_result!](#)

Arguments

Name	Description
target_organisation - hierarchy_organisation_organisation_reference!	The organisation that should be updated.
input - hierarchy_organisation_update_organisation_input!	Input object specifying data for updating the organisation.

QUERY

```
mutation hierarchy_organisation_update_organisation(  
  $target_organisation: hierarchy_organisation_organisation_reference!,  
  $input: hierarchy_organisation_update_organisation_input!  
) {  
  hierarchy_organisation_update_organisation(  
    target_organisation: $target_organisation,  
    input: $input  
  ) {  
    organisation {  
      ...hierarchy_organisation_organisationFragment  
    }  
  }  
}
```

VARIABLES

```
{  
  "target_organisation": hierarchy_organisation_organisation_reference,
```

```
"input": hierarchy_organisation_update_organisation_input
}
```

RESPONSE

```
{
  "data": {
    "hierarchy_organisation_update_organisation": {
      "organisation": hierarchy_organisation_organisation
    }
  }
}
```

Types

hierarchy_organisation_create_organisation_input

Input provided when creating an organisation.

Input Field	Description
fullname - String!	Full name of the organisation.
idnumber - String	Unique reference used to represent the organisation across multiple systems.
description - String	Rich-text description of the organisation.
framework - hierarchy_organisation_framework_reference!	The organisation framework this organisation belongs to.
parent - hierarchy_organisation_organisation_reference	The parent organisation in the hierarchy.
type - hierarchy_organisation_type_reference	Organisation type.
visible - Boolean	Flag determining whether the organisation is visible to end users or not. If not given, will default to visible.
custom_fields - [totara_customfield_input!]	Optional array of organisation type custom field data (each array element describes a single custom field value for this organisation).

EXAMPLE

```

{
  "fullname": "abc123",
  "idnumber": "xyz789",
  "description": "abc123",
  "framework": hierarchy_organisation_framework_reference,
  "parent": hierarchy_organisation_organisation_reference,
  "type": hierarchy_organisation_type_reference,
  "visible": true,
  "custom_fields": [totara_customfield_input]
}

```

Types

hierarchy_organisation_framework

Organisation framework object. An organisation framework is a collection of organisations arranged in a tree.

Field Name	Description
<code>id</code> - core_id!	Internal database id of the organisation framework.
<code>fullname</code> - String!	Full name of the organisation framework.

Arguments

`format` - [core_format](#)

Output format this field should be formatted for.

<code>idnumber</code> - String	Unique reference used to represent the organisation framework across multiple systems.
<code>shortname</code> - String	Short name of the organisation framework, only used if admin setting 'showhierarchyshortnames' is enabled.

Arguments

`format` - [core_format](#)

Output format this field should be formatted for.

<code>description</code> - String	Rich-text description of the organisation framework.
---------------------------------------------------	------------------------------------------------------

Field Name	Description
Arguments	
<code>format - core_format</code>	
	Output format this field should be formatted for.

EXAMPLE

```
{
  "id": 4,
  "fullname": "abc123",
  "idnumber": "abc123",
  "shortname": "xyz789",
  "description": "xyz789"
}
```

Types

hierarchy_organisation_framework_reference

Input for identifying an organisation framework.

Input Field	Description
<code>id - core_id</code>	Identify an organisation framework by internal database id.
<code>idnumber - String</code>	Identify an organisation framework by its idnumber. <code>idnumber</code> is a unique reference used to represent the organisation framework across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "abc123"}
```

Types

hierarchy_organisation_organisation

Organisation item object.

The organisation structure defines the regions, departments, groups, areas, or teams that a company has defined.

Field Name	Description
<code>id - core_id!</code>	Internal database id of the organisation.
<code>fullname - String!</code>	Full name of the organisation.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format - core_format</code></p> <p>Output format this field should be formatted for.</p> </div>	
<code>idnumber - String</code>	Unique reference used to represent the organisation across multiple systems.
<code>shortname - String</code>	The short name of the organisation. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format - core_format</code></p> <p>Output format this field should be formatted for.</p> </div>	
<code>description - String</code>	Rich-text description of the organisation.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format - core_format</code></p> <p>Output format this field should be formatted for. Please note: RAW format is only available to users with the capability 'totara/hierarchy:updateorganisation' in the system context, as its purpose is for editing.</p> </div>	
<code>frameworkid - core_id</code>	The internal database id of the organisation framework this organisation belongs to. Please note: viewing this organisation's frameworkid requires the capability 'totara/hierarchy:vieworganisationframeworks'
<code>framework - hierarchy_organisation_framework</code>	The organisation framework this organisation belongs to. Please note: viewing this organisation's framework requires the capability 'totara/hierarchy:vieworganisationframeworks'
<code>path - String!</code>	An ordered list of parent database ids for this organisation, indicating the hierarchy of parent items. Includes the current item and separated by forward slashes, e.g. item with id=6 might be: /1/4/6 This is a computed value.
<code>parentid - core_id</code>	The id of this organisation's immediate parent in the framework hierarchy. Set to 0 if this is a top-level organisation.

Field Name	Description
parent - hierarchy_organisation_organisation	The parent organisation in the hierarchy. Null if a top-level item.
children - [hierarchy_organisation_organisation!]	Collection of organisations that are immediately below this one in the framework hierarchy. Empty if this organisation has no children.
typeid - core_id	The id of the organisation type, or 0 if this is a generic organisation.
type - hierarchy_organisation_type	Organisation type object.
timemodified - core_date!	Timestamp of when the hierarchy item type was last modified. Available since Totara 19.0.1.
<div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>Arguments</p> <hr/> <p>format - core_date_format</p> </div>	
visible - Boolean!	Flag determining whether the organisation is visible to end users or not.
custom_fields - [totara_customfield_field!]	Custom fields relating to the organisation

EXAMPLE

```
{
  "id": 4,
  "fullname": "xyz789",
  "idnumber": "abc123",
  "shortname": "abc123",
  "description": "abc123",
  "frameworkid": 4,
  "framework": hierarchy_organisation_framework,
  "path": "xyz789",
  "parentid": 4,
  "parent": hierarchy_organisation_organisation,
  "children": [hierarchy_organisation_organisation],
  "typeid": 4,
  "type": hierarchy_organisation_type,
  "timemodified": "2022-04-17",
  "visible": true,
  "custom_fields": [totara_customfield_field]
}
```

Types

[hierarchy_organisation_organisation_created_result](#)

Mutation result type returned after creation of a new organisation. Returns the organisation that was created.

Field Name	Description
organisation - hierarchy_organisation_organisation	Organisation object for the organisation that was created as part of the operation.

EXAMPLE

```
{"organisation": hierarchy_organisation_organisation}
```

Types

hierarchy_organisation_organisation_deleted_result

Mutation result type returned after deletion of an organisation. Returns the id of deleted organisation if it was deleted successfully.

Field Name	Description
organisation_id - core_id!	Internal database id of the organisation that was just deleted from the system.

EXAMPLE

```
{"organisation_id": 4}
```

Types

hierarchy_organisation_organisation_reference

Input for identifying an organisation.

Input Field	Description
id - core_id	Identify an organisation by internal database id.
idnumber - String	Identify an organisation by its idnumber. <code>idnumber</code> is a unique reference used to represent the organisation across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "abc123"}
```

Types

hierarchy_organisation_organisation_result

Query result type returned when querying a single organisation.

Field Name	Description
found - Boolean!	Whether the organisation was found. This may return false if the record doesn't exist within Totara or if the API client doesn't have access to the record.
organisation - hierarchy_organisation_organisation	Organisation object for the organisation that was requested as part of the operation.

EXAMPLE

```
{
  "found": true,
  "organisation": hierarchy_organisation_organisation
}
```

Types

hierarchy_organisation_organisation_updated_result

Mutation result type returned after updates to an existing organisation. Returns the organisation that was updated.

Field Name	Description
organisation - hierarchy_organisation_organisation	Organisation object for the organisation that was updated as part of the operation.

EXAMPLE

```
{"organisation": hierarchy_organisation_organisation}
```

Types

hierarchy_organisation_organisations_filter

Organisation filter

Properties to filter out organisations.

Input Field	Description
<code>framework_id</code> - param_integer	The internal ID of a framework to filter on.
<code>parent_id</code> - param_integer	The internal ID of a parent organisation to filter on.
<code>ids</code> - [param_integer!]	A list of internal IDs of organisations to filter on.
<code>name</code> - param_text	Filter based on the name of the organisations.
<code>type_id</code> - param_integer	The internal ID of a type to filter on.
<code>since_timemodified</code> - core_date	The timestamp of when the organisation was last modified. Handles the following formats: integer timestamp, numeric string or an ISO-8601 string. Available since Totara 19.0.1.

EXAMPLE

```
{
  "framework_id": 123,
  "parent_id": 123,
  "ids": [123],
  "name": "This is a string",
  "type_id": 123,
  "since_timemodified": "2022-04-17"
}
```

Types

hierarchy_organisation_organisations_query

Input type for the organisations query.

Input Field	Description
<code>pagination</code> - core_pagination_input	The pagination object contains information about the number of items to be returned, and the page offset to start from.
<code>sort</code> - [core_sort_input!]	The sort order for the query. The query is currently only able to sort by one order.
<code>filters</code> - hierarchy_organisation_organisations_filter	The filters for querying organisations.

EXAMPLE

```
{
  "pagination": core_pagination_input,
  "sort": [core_sort_input],
  "filters": hierarchy_organisation_organisations_filter
}
```

Types

hierarchy_organisation_organisations_result

Organisation result

Result type for a collection of organisations.

Field Name	Description
items - [hierarchy_organisation_organisation!]!	Array of one page of organisations returned by the query.
total - Int!	Total number of organisations from this query (across all pages).
next_cursor - String!	Cursor to request the next set of results for this query.

EXAMPLE

```
{
  "items": [hierarchy_organisation_organisation],
  "total": 987,
  "next_cursor": "abc123"
}
```

Types

hierarchy_organisation_type

Organisation type object. An organisation type is a user-defined classification which can be assigned to organisations leading to certain custom fields being available on that organisation.

Field Name	Description
id - core_id!	Internal database id of the organisation type.
fullname - String!	Organisation type full name.

Field Name	Description
<p>Arguments</p> <hr/> <p><code>format</code> - <code>core_format</code></p> <p>Output format this field should be formatted for.</p>	
<code>idnumber</code> - <code>String</code>	Unique reference used to represent the organisation type across multiple systems.
<code>shortname</code> - <code>String</code>	The short name of the organisation type. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.
<p>Arguments</p> <hr/> <p><code>format</code> - <code>core_format</code></p> <p>Output format this field should be formatted for.</p>	
<code>description</code> - <code>String</code>	Rich-text description of the organisation type.
<p>Arguments</p> <hr/> <p><code>format</code> - <code>core_format</code></p> <p>Output format this field should be formatted for.</p>	

EXAMPLE

```
{
  "id": 4,
  "fullname": "xyz789",
  "idnumber": "abc123",
  "shortname": "xyz789",
  "description": "xyz789"
}
```

Types

hierarchy_organisation_type_reference

Input for identifying an organisation type.

Input Field	Description
<code>id</code> - <code>core_id</code>	Identify an organisation type by internal database id.

Input Field	Description
idnumber - String	Identify an organisation type by its idnumber. idnumber is a unique reference used to represent the organisation type across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "abc123"}
```

Types

hierarchy_organisation_update_organisation_input

Input provided when updating an organisation.

Input Field	Description
fullname - String	Full name of the organisation.
idnumber - String	Unique reference used to represent the organisation across multiple systems.
description - String	Rich-text description of the organisation.
framework - hierarchy_organisation_framework_reference	The organisation framework this organisation belongs to.
parent - hierarchy_organisation_organisation_reference	The parent organisation in the hierarchy.
type - hierarchy_organisation_type_reference	Organisation type.
visible - Boolean	Flag determining whether the organisation is visible to end users or not.
custom_fields - [totara_customfield_input!]	Optional array of organisation type custom field data (each array element describes a single custom field value for this organisation).

EXAMPLE

```
{
  "fullname": "abc123",
  "idnumber": "xyz789",
  "description": "abc123",
  "framework": hierarchy_organisation_framework_reference,
  "parent": hierarchy_organisation_organisation_reference,
  "type": hierarchy_organisation_type_reference,
}
```

```
"visible": true,  
"custom_fields": [totara_customfield_input]  
}
```

Position Hierarchies

Queries

hierarchy_position_position

Return a specific position.

Response

Returns a [hierarchy_position_position_result!](#)

Arguments

Name	Description
reference - hierarchy_position_position_reference!	Input object specifying data for the existing position.

QUERY

```
query hierarchy_position_position($reference: hierarchy_position_position!  
  hierarchy_position_position(reference: $reference) {  
  found  
  position {  
    ...hierarchy_position_positionFragment  
  }  
}
```

VARIABLES

```
{"reference": hierarchy_position_position_reference}
```

RESPONSE

```
{  
  "data": {
```

```
"hierarchy_position_position": {
  "found": true,
  "position": hierarchy_position_position
}
}
```

Queries

hierarchy_position_positions

Return a list of positions.

Response

Returns a [hierarchy_position_positions_result!](#)

Arguments

Name	Description
------	-------------

query - hierarchy_position_positions_query	
---------------------------------------------------------------	--

QUERY

```
query hierarchy_position_positions($query: hierarchy_position_positions_query!) {
  hierarchy_position_positions(query: $query) {
    items {
      ...hierarchy_position_positionFragment
    }
    total
    next_cursor
  }
}
```

VARIABLES

```
{"query": hierarchy_position_positions_query}
```

RESPONSE

```
{
  "data": {
    "hierarchy_position_positions": {
      "items": [hierarchy_position_position],
    }
  }
}
```

```
"total": 987,  
  "next_cursor": "xyz789"  
}  
}  
}
```

Mutations

hierarchy_position_create_position

Creates a new position. Please note: creating positions requires the capability 'totara/hierarchy:createposition'

Response

Returns a [hierarchy_position_position_created_result!](#)

Arguments

Name	Description
input - hierarchy_position_create_position_input!	Input object specifying data for creating the new position.

QUERY

```
mutation hierarchy_position_create_position($input: hierarchy_position_create_position_create_position_input!) {  
  hierarchy_position_create_position(input: $input) {  
    position {  
      ...hierarchy_position_positionFragment  
    }  
  }  
}
```

VARIABLES

```
{"input": hierarchy_position_create_position_input}
```

RESPONSE

```
{  
  "data": {  
    "hierarchy_position_create_position": {
```

```
    "position": hierarchy_position_position
  }
}
```

Mutations

hierarchy_position_delete_position

Mutation to delete an position. Please note: deleting positions requires the capability 'totara/hierarchy:deleteposition'

Response

Returns a [hierarchy_position_position_deleted_result!](#)

Arguments

Name	Description
target_position - hierarchy_position_position_reference !	Input provided for identifying the position.

QUERY

```
mutation hierarchy_position_delete_position($target_position: hierarchy_position_position_reference!) {
  hierarchy_position_delete_position(target_position: $target_position) {
    position_id
  }
}
```

VARIABLES

```
{"target_position": hierarchy_position_position_reference}
```

RESPONSE

```
{"data": {"hierarchy_position_delete_position": {"position_id": 4}}}
```

Mutations

hierarchy_position_update_position

Updates a specific target position with new properties. Please note: updating positions requires the capability 'totara/hierarchy:updateposition'

Response

Returns a [hierarchy_position_position_updated_result](#)

Arguments

Name	Description
target_position - hierarchy_position_position_reference!	The position that should be updated.
input - hierarchy_position_update_position_input!	Input object specifying data for updating the position.

QUERY

```
mutation hierarchy_position_update_position(  
  $target_position: hierarchy_position_position_reference!,  
  $input: hierarchy_position_update_position_input!  
) {  
  hierarchy_position_update_position(  
    target_position: $target_position,  
    input: $input  
  ) {  
    position {  
      ...hierarchy_position_positionFragment  
    }  
  }  
}
```

VARIABLES

```
{  
  "target_position": hierarchy_position_position_reference,  
  "input": hierarchy_position_update_position_input  
}
```

RESPONSE

```
{  
  "data": {  
    "hierarchy_position_update_position": {  
      "position": hierarchy_position_position  
    }  
  }  
}
```

```
}  
}  
}
```

Types

hierarchy_position_create_position_input

Input provided when creating a position.

Input Field	Description
fullname - String!	Full name of the position.
idnumber - String	Unique reference used to represent the position across multiple systems.
description - String	Rich-text description of the position.
framework - hierarchy_position_framework_reference!	The position framework this position belongs to.
parent - hierarchy_position_position_reference	The parent position in the hierarchy.
type - hierarchy_position_type_reference	Position type.
visible - Boolean	Flag determining whether the position is visible to end users or not. If not given, will default to visible.
custom_fields - [totara_customfield_input!]	Optional array of position type custom field data (each array element describes a single custom field value for this position).

EXAMPLE

```
{  
  "fullname": "abc123",  
  "idnumber": "xyz789",  
  "description": "abc123",  
  "framework": hierarchy_position_framework_reference,  
  "parent": hierarchy_position_position_reference,  
  "type": hierarchy_position_type_reference,  
  "visible": true,  
  "custom_fields": [totara_customfield_input]  
}
```

Types

hierarchy_position_framework

Position framework object. A position framework is a collection of positions arranged in a tree.

Field Name	Description
<code>id</code> - core_id!	Internal database id of the position framework.
<code>fullname</code> - String!	Full name of the position framework.
<hr/> <p>Arguments</p> <hr/> <p><code>format</code> - core_format</p> <p>Output format this field should be formatted for.</p> <hr/>	
<code>idnumber</code> - String	Unique reference used to represent the position framework across multiple systems.
<code>shortname</code> - String	Short name of the position framework, only used if admin setting 'showhierarchysortnames' is enabled.
<hr/> <p>Arguments</p> <hr/> <p><code>format</code> - core_format</p> <p>Output format this field should be formatted for.</p> <hr/>	
<code>description</code> - String	Rich-text description of the position framework.
<hr/> <p>Arguments</p> <hr/> <p><code>format</code> - core_format</p> <p>Output format this field should be formatted for.</p> <hr/>	

EXAMPLE

```
{
  "id": 4,
  "fullname": "abc123",
  "idnumber": "abc123",
  "shortname": "abc123",
  "description": "abc123"
}
```

Types

hierarchy_position_framework_reference

Input for identifying a position framework.

Input Field	Description
<code>id - core_id</code>	Identify a position framework by internal database id.
<code>idnumber - String</code>	Identify a position framework by its idnumber. <code>idnumber</code> is a unique reference used to represent the position framework across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "xyz789"}
```

Types

hierarchy_position_position

Position item object.

The position structure defines the job roles that a company has defined.

Field Name	Description
<code>id - core_id!</code>	Internal database id of the position.
<code>fullname - String!</code>	Full name of the position.

Arguments

`format - core_format`

Output format this field should be formatted for.

<code>idnumber - String</code>	Unique reference used to represent the position across multiple systems.
<code>shortname - String</code>	The short name of the position. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.

Arguments

Field Name	Description
format - core_format	
Output format this field should be formatted for.	
description - String	Rich-text description of the position.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for. Please note: RAW format is only available to users with the capability 'totara/hierarchy:updateposition' in the system context, as its purpose is for editing.</p>	
frameworkid - core_id	The internal database id of the position framework this position belongs to. Please note: viewing this position's frameworkid requires the capability 'totara/hierarchy:viewpositionframeworks'
framework - hierarchy_position_framework	The position framework this position belongs to. Please note: viewing this position's framework requires the capability 'totara/hierarchy:viewpositionframeworks'
path - String!	An ordered list of parent database ids for this position, indicating the hierarchy of parent items. Includes the current item and separated by forward slashes, e.g. item with id=6 might be: /1/4/6 This is a computed value.
parentid - core_id	The id of this position's immediate parent in the framework hierarchy. Set to 0 if this is a top-level position.
parent - hierarchy_position_position	The parent position in the hierarchy. Null if a top-level item.
children - [hierarchy_position_position!]!	Collection of positions that are immediately below this one in the framework hierarchy. Empty if this position has no children.
typeid - core_id	The id of the position type, or 0 if this is a generic position.
type - hierarchy_position_type	Position type object.
timemodified - core_date!	Timestamp of when the position was last modified. Available since Totara 19.0.1.
<p>Arguments</p> <hr/> <p>format - core_date_format</p>	

Field Name	Description
visible - Boolean!	Flag determining whether the position is visible to users or not.
custom_fields - [totara_customfield_field!]	Custom fields relating to the position

EXAMPLE

```
{
  "id": 4,
  "fullname": "abc123",
  "idnumber": "abc123",
  "shortname": "xyz789",
  "description": "xyz789",
  "frameworkid": 4,
  "framework": hierarchy_position_framework,
  "path": "abc123",
  "parentid": 4,
  "parent": hierarchy_position_position,
  "children": [hierarchy_position_position],
  "typeid": 4,
  "type": hierarchy_position_type,
  "timemodified": "2022-04-17",
  "visible": true,
  "custom_fields": [totara_customfield_field]
}
```

Types

hierarchy_position_position_created_result

Mutation result type returned after creation of a new position. Returns the position that was created.

Field Name	Description
position - hierarchy_position_position	Position object for the position that was created as part of the operation.

EXAMPLE

```
{"position": hierarchy_position_position}
```

Types

hierarchy_position_position_deleted_result

Mutation result type returned after deletion of a position. Returns the id of deleted position if it was deleted successfully.

Field Name	Description
<code>position_id - core_id!</code>	Internal database id of the position that was just deleted from the system.

EXAMPLE

```
{"position_id": 4}
```

Types

hierarchy_position_position_reference

Input for identifying a position.

Input Field	Description
<code>id - core_id</code>	Identify a position by internal database id.
<code>idnumber - String</code>	Identify a position by its idnumber. <code>idnumber</code> is a unique reference used to represent the position across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "abc123"}
```

Types

hierarchy_position_position_result

Represents a single position result.

Field Name	Description
<code>found - Boolean</code>	Whether or not the position is found. This may return false if the record doesn't exist within Totara or if the API client doesn't have access to the record.
<code>position - hierarchy_position_position</code>	The found position. This will be null if the position couldn't be found.

EXAMPLE

```
{"found": true, "position": hierarchy_position_position}
```

Types

hierarchy_position_position_updated_result

Mutation result type returned after updates to an existing position. Returns the position that was updated.

Field Name	Description
position - hierarchy_position_position	Position object for the position that was updated as part of the operation.

EXAMPLE

```
{"position": hierarchy_position_position}
```

Types

hierarchy_position_positions_filter

Position filter

Properties to filter out positions.

Input Field	Description
framework_id - param_integer	The internal ID of a framework to filter on.
parent_id - param_integer	The internal ID of a parent position to filter on.
ids - [param_integer!]	A list of internal IDs of positions to filter on.
name - param_text	Filter based on the name of the positions.
type_id - param_integer	The internal ID of a type to filter on.
since_timemodified - core_date	The timestamp of when the position was last modified. Handles the following formats: integer timestamp, numeric string or an ISO-8601 string. Available since Totara 19.0.1.

EXAMPLE

```
{
  "framework_id": 123,
  "parent_id": 123,
  "ids": [123],
  "name": "This is a string",
  "type_id": 123,
  "since_timemodified": "2022-04-17"
}
```

Types

hierarchy_position_positions_query

Input type for the positions query.

Input Field	Description
pagination - core_pagination_input	The pagination object contains information about the number of items to be returned, and the page offset to start from.
sort - [core_sort_input!]	The sort order for the query. The query is currently only able to sort by one order.
filters - hierarchy_position_positions_filter	The filters for querying positions.

EXAMPLE

```
{
  "pagination": core_pagination_input,
  "sort": [core_sort_input],
  "filters": hierarchy_position_positions_filter
}
```

Types

hierarchy_position_positions_result

Position result

Result type for a collection of positions.

Field Name	Description
items - [hierarchy_position_position!]!	Array of one page of positions returned by the query.

Field Name	Description
<code>total</code> - Int!	Total number of positions from this query (across all pages).
<code>next_cursor</code> - String!	Cursor to request the next set of results for this query.

EXAMPLE

```
{
  "items": [hierarchy_position_position],
  "total": 123,
  "next_cursor": "abc123"
}
```

Types

hierarchy_position_type

Position type object. A position type is a user-defined classification which can be assigned to positions leading to certain custom fields being available on that position.

Field Name	Description
<code>id</code> - core_id!	Internal database id of the position type.
<code>fullname</code> - String!	Position type full name.

Arguments

`format` - [core_format](#)

Output format this field should be formatted for.

<code>idnumber</code> - String	Unique reference used to represent the position type across multiple systems.
<code>shortname</code> - String	The short name of the position type. Only used as additional information if admin setting 'showhierarchyshortnames' is enabled.

Arguments

`format` - [core_format](#)

Output format this field should be formatted for.

<code>description</code> - String	Rich-text description of the position type.
---------------------------------------------------	---------------------------------------------

Arguments

Field Name	Description
format - core_format	Output format this field should be formatted for.

EXAMPLE

```
{
  "id": 4,
  "fullname": "xyz789",
  "idnumber": "xyz789",
  "shortname": "abc123",
  "description": "xyz789"
}
```

Types

hierarchy_position_type_reference

Input for identifying a position type.

Input Field	Description
id - core_id	Identify a position type by internal database id.
idnumber - String	Identify a position type by its idnumber. <code>idnumber</code> is a unique reference used to represent the position type across multiple systems.

EXAMPLE

```
{"id": 4, "idnumber": "abc123"}
```

Types

hierarchy_position_update_position_input

Input provided when updating a position.

Input Field	Description
fullname - String	Full name of the position.
idnumber - String	Unique reference used to represent the position across multiple systems.

Input Field	Description
<code>description</code> - String	Rich-text description of the position.
<code>framework</code> - hierarchy_position_framework_reference	The position framework this position belongs to.
<code>parent</code> - hierarchy_position_position_reference	The parent position in the hierarchy.
<code>type</code> - hierarchy_position_type_reference	Position type.
<code>visible</code> - Boolean	Flag determining whether the position is visible to end users or not.
<code>custom_fields</code> - [totara_customfield_input!]	Optional array of position type custom field data (each array element describes a single custom field value for this position).

EXAMPLE

```
{
  "fullname": "xyz789",
  "idnumber": "abc123",
  "description": "xyz789",
  "framework": hierarchy_position_framework_reference,
  "parent": hierarchy_position_position_reference,
  "type": hierarchy_position_type_reference,
  "visible": true,
  "custom_fields": [totara_customfield_input]
}
```

Tag

Types

core_tag

Tag type object, containing information about a tag.

Field Name	Description
<code>name</code> - String!	Name of tag.
<code>rawname</code> - String!	Rawname of tag.

EXAMPLE

```
{"name": "education", "rawname": "Education"}
```

Totara goals

Queries

perform_goal_get_goal_tasks

A query to get the tasks for a specified goal.

Response

Returns a [perform_goal_get_goal_tasks_result!](#)

Arguments

Name	Description
goal_reference - perform_goal_reference!	Input value(s) for identifying a goal: id and/or id_number.

QUERY

```
query perform_goal_get_goal_tasks($goal_reference: perform_goal_reference!){
  perform_goal_get_goal_tasks(goal_reference: $goal_reference) {
    goal_id
    tasks {
      ...perform_goal_goal_taskFragment
    }
    total
    completed
  }
}
```

VARIABLES

```
{"goal_reference": perform_goal_reference}
```

RESPONSE

```

{
  "data": {
    "perform_goal_get_goal_tasks": {
      "goal_id": 4,
      "tasks": [perform_goal_goal_task],
      "total": 987,
      "completed": 987
    }
  }
}

```

Queries

perform_goal_user_goals

A query to fetch the available goals for a user.

Response

Returns a [perform_goal_user_goals_result!](#)

Arguments

Name	Description
input - perform_goal_user_goals_input!	Filter and pagination parameters for the user goals query.

QUERY

```

query perform_goal_user_goals($input: perform_goal_user_goals_input!) {
  perform_goal_user_goals(input: $input) {
    items {
      ...perform_goal_user_goals_result_itemFragment
    }
    total
    next_cursor
    has_goals
    success
    errors {
      ...perform_goal_operation_errorFragment
    }
  }
}

```

VARIABLES

```
{"input": perform_goal_user_goals_input}
```

RESPONSE

```
{
  "data": {
    "perform_goal_user_goals": {
      "items": [perform_goal_user_goals_result_item],
      "total": 123,
      "next_cursor": "abc123",
      "has_goals": true,
      "success": true,
      "errors": perform_goal_operation_error
    }
  }
}
```

Queries

perform_goal_view_goal

A query to fetch information about a single goal.

Response

Returns a [perform_goal_view_result!](#)

Arguments

Name	Description
goal_reference - perform_goal_reference!	Input value(s) for identifying a goal: id and/or id_number.

QUERY

```
query perform_goal_view_goal($goal_reference: perform_goal_reference!) {
  perform_goal_view_goal(goal_reference: $goal_reference) {
    goal {
      ...perform_goal_goalFragment
    }
    raw {
      ...perform_goal_goal_raw_dataFragment
    }
    permissions {
      ...perform_goal_permissionsFragment
    }
  }
}
```

```
}  
}
```

VARIABLES

```
{"goal_reference": perform_goal_reference}
```

RESPONSE

```
{  
  "data": {  
    "perform_goal_view_goal": {  
      "goal": perform_goal_goal,  
      "raw": perform_goal_goal_raw_data,  
      "permissions": perform_goal_permissions  
    }  
  }  
}
```

Mutations

perform_goal_create_goal

An operation to create a new goal, containing the information needed to initialise it for use.

Response

Returns a [perform_goal_crud_result!](#)

Arguments

Name	Description
input - perform_goal_create_input!	Values provided to create a new goal.

QUERY

```
mutation perform_goal_create_goal($input: perform_goal_create_input!) {  
  perform_goal_create_goal(input: $input) {  
    goal {  
      ...perform_goal_goalFragment  
    }  
    raw {  
      ...perform_goal_goal_raw_dataFragment  
    }  
  }  
}
```

```
permissions {
  ...perform_goal_permissionsFragment
}
```

VARIABLES

```
{"input": perform_goal_create_input}
```

RESPONSE

```
{
  "data": {
    "perform_goal_create_goal": {
      "goal": perform_goal_goal,
      "raw": perform_goal_goal_raw_data,
      "permissions": perform_goal_permissions
    }
  }
}
```

Mutations

perform_goal_create_goal_task

An operation to create a new goal task, containing the information needed to initialise it for use.

Response

Returns a [perform_goal_task_upsert_result!](#)

Arguments

Name	Description
input - perform_goal_task_create_input!	Values provided to create a new goal task.

QUERY

```
mutation perform_goal_create_goal_task($input: perform_goal_task_create_in
  perform_goal_create_goal_task(input: $input) {
    goal_task {
      ...perform_goal_goal_taskFragment
    }
    success
    errors {
```

```
    ...perform_goal_task_operation_errorFragment
  }
}
```

VARIABLES

```
{"input": perform_goal_task_create_input}
```

RESPONSE

```
{
  "data": {
    "perform_goal_create_goal_task": {
      "goal_task": perform_goal_goal_task,
      "success": true,
      "errors": perform_goal_task_operation_error
    }
  }
}
```

Mutations

perform_goal_delete_goal

An operation to delete an existing goal.

Response

Returns a [perform_goal_delete_result!](#)

Arguments

Name	Description
goal_reference - perform_goal_reference!	Input value(s) for identifying a goal.

QUERY

```
mutation perform_goal_delete_goal($goal_reference: perform_goal_reference!
  perform_goal_delete_goal(goal_reference: $goal_reference) {
  success
}
```

VARIABLES

```
{"goal_reference": perform_goal_reference}
```

RESPONSE

```
{"data": {"perform_goal_delete_goal": {"success": true}}}
```

Mutations

perform_goal_delete_goal_task

An operation to delete an existing goal task.

Response

Returns a [perform_goal_task_delete_result!](#)

Arguments

Name	Description
goal_task_reference - perform_goal_task_reference!	Input value(s) for identifying a goal task.

QUERY

```
mutation perform_goal_delete_goal_task($goal_task_reference: perform_goal_task_reference!) {
  perform_goal_delete_goal_task(goal_task_reference: $goal_task_reference) {
    success
  }
}
```

VARIABLES

```
{"goal_task_reference": perform_goal_task_reference}
```

RESPONSE

```
{"data": {"perform_goal_delete_goal_task": {"success": true}}}
```

Mutations

perform_goal_set_goal_task_progress

An operation to set the progress of a single goal task.

Response

Returns a [perform_goal_task_upsert_result!](#)

Arguments

Name	Description
input - perform_goal_task_set_progress_input!	Input value for setting a goal task's progress.

QUERY

```
mutation perform_goal_set_goal_task_progress($input: perform_goal_task_set_goal_task_set_progress_input!) {
  perform_goal_set_goal_task_progress(input: $input) {
    goal_task {
      ...perform_goal_goal_taskFragment
    }
    success
    errors {
      ...perform_goal_task_operation_errorFragment
    }
  }
}
```

VARIABLES

```
{"input": perform_goal_task_set_progress_input}
```

RESPONSE

```
{
  "data": {
    "perform_goal_set_goal_task_progress": {
      "goal_task": perform_goal_goal_task,
      "success": true,
      "errors": perform_goal_task_operation_error
    }
  }
}
```

Mutations

perform_goal_update_goal

An operation to modify an existing goal's information.

Response

Returns a [perform_goal_crud_result!](#)

Arguments

Name	Description
goal_reference - perform_goal_reference!	Input value(s) for identifying a goal.
input - perform_goal_update_input!	Values provided for updating a goal.

QUERY

```
mutation perform_goal_update_goal(  
  $goal_reference: perform_goal_reference!,  
  $input: perform_goal_update_input!  
) {  
  perform_goal_update_goal(  
    goal_reference: $goal_reference,  
    input: $input  
  ) {  
    goal {  
      ...perform_goal_goalFragment  
    }  
    raw {  
      ...perform_goal_goal_raw_dataFragment  
    }  
    permissions {  
      ...perform_goal_permissionsFragment  
    }  
  }  
}
```

VARIABLES

```
{  
  "goal_reference": perform_goal_reference,  
  "input": perform_goal_update_input  
}
```

RESPONSE

```
{
  "data": {
    "perform_goal_update_goal": {
      "goal": perform_goal_goal,
      "raw": perform_goal_goal_raw_data,
      "permissions": perform_goal_permissions
    }
  }
}
```

Mutations

perform_goal_update_goal_task

An operation to update a goal task.

Response

Returns a [perform_goal_task_upsert_result!](#)

Arguments

Name	Description
input - perform_goal_task_update_input!	Values provided to create a new goal task.

QUERY

```
mutation perform_goal_update_goal_task($input: perform_goal_task_update_in
  perform_goal_update_goal_task(input: $input) {
    goal_task {
      ...perform_goal_goal_taskFragment
    }
    success
    errors {
      ...perform_goal_task_operation_errorFragment
    }
  }
}
```

VARIABLES

```
{"input": perform_goal_task_update_input}
```

RESPONSE

```
{
  "data": {
    "perform_goal_update_goal_task": {
      "goal_task": perform_goal_goal_task,
      "success": true,
      "errors": perform_goal_task_operation_error
    }
  }
}
```

Mutations

perform_goal_update_progress

An operation to update an existing goal's progress information, i.e. the goal status and current_value.

Response

Returns a [perform_goal_update_progress_result!](#)

Arguments

Name	Description
goal_reference - perform_goal_reference!	Input value(s) for identifying a goal.
input - perform_goal_update_progress_input!	Values to update a goal's progress with.

QUERY

```
mutation perform_goal_update_progress(
  $goal_reference: perform_goal_reference!,
  $input: perform_goal_update_progress_input!
) {
  perform_goal_update_progress(
    goal_reference: $goal_reference,
    input: $input
  ) {
    current_value
    status {
      ...perform_goal_statusFragment
    }
  }
}
```

VARIABLES

```
{
  "goal_reference": perform_goal_reference,
  "input": perform_goal_update_progress_input
}
```

RESPONSE

```
{
  "data": {
    "perform_goal_update_progress": {
      "current_value": 987.65,
      "status": perform_goal_status
    }
  }
}
```

Types

perform_goal_create_input

Input provided when creating a goal. Some fields are required.

Input Field	Description
context_id - core_id	Parent context ID of the goal object.
owner - core_user_user_reference	The user that 'owns' the goal, i.e. is responsible for managing it. Usually the owner will be the same person as the user who initially created the goal. If this field is not included, the default owner will be set to the user making the request.
user - core_user_user_reference	The user who is the subject of the goal, i.e. who the goal is applied to.
name - String!	The display name for the goal.
id_number - String	An external identifier for the goal.
description - String	Comments or a statement about the goal.
start_date - core_date!	The date the goal will commence.
target_type - String	The type of the target for the goal, e.g. date, percentage, currency, number, manual. Current target type is 'date'.

Input Field	Description
target_date - core_date!	The date the goal is expected to be completed/closed.
target_value - Float!	The expected value for achieving the goal.
current_value - Float	The value towards the goal currently achieved so far.
status - String	The current status of the goal (currently supports the 'not_started', 'in_progress', 'completed', 'cancelled').
plugin_name - String	The module/plugin name (currently supports the 'basic' plugin).

EXAMPLE

```
{
  "context_id": 4,
  "owner": core_user_user_reference,
  "user": core_user_user_reference,
  "name": "Display knowledge of 5 sales methods",
  "id_number": "sf_10133",
  "description": "The learner must show knowledge of 5 sales methods.",
  "start_date": 1696289455,
  "target_type": "date",
  "target_date": 1701513055,
  "target_value": 100,
  "current_value": 987.65,
  "status": "in_progress",
  "plugin_name": "basic"
}
```

Types

perform_goal_crud_result

The result returned from a goal create/update/delete operation.

Field Name	Description
goal - perform_goal_goal!	Goal type object, containing information about a personal goal.
raw - perform_goal_goal_raw_data!	Goal metadata from a goal create/update/delete operation.
permissions - perform_goal_permissions!	Permissions for the current user to manipulate goals.

EXAMPLE

```
{
  "goal": perform_goal_goal,
  "raw": perform_goal_goal_raw_data,
  "permissions": perform_goal_permissions
}
```

Types

perform_goal_delete_result

The result returned from a goal delete operation.

Field Name	Description
success - Boolean!	Whether the current goal successfully deleted.

EXAMPLE

```
{"success": true}
```

Types

perform_goal_get_goal_tasks_result

The result returned from a get goal tasks operation.

Field Name	Description
goal_id - core_id!	The database ID of the parent goal.
tasks - [perform_goal_goal_task!]	The list of goal_task items associated with a goal.
total - Int!	Total number of tasks in this goal.
completed - Int!	Total number of <i>completed</i> tasks in this goal.

EXAMPLE

```
{
  "goal_id": 4,
  "tasks": [perform_goal_goal_task],
  "total": 123,
  "completed": 123
}
```

Types

perform_goal_goal

Goal type object, containing information about a personal or team goal. All available fields are listed. No fields are required.

Field Name	Description
<code>id</code> - core_id	Internal database ID of the goal.
<code>context_id</code> - core_id	Parent context ID of the goal object.
<code>category_id</code> - core_id	Perform goal category, associates a category name with a goaltypesubplugin.
<code>owner</code> - core_user	The user that 'owns' the goal, i.e. is responsible for managing it. Usually the owner will be the same person as the user who initially created the goal.
<code>user</code> - core_user	The user who is the subject of the goal, i.e. who the goal is applied to.
<code>name</code> - String	The display name for the goal.
<code>id_number</code> - String	An external identifier for the goal.
<code>description</code> - String	Comments or a statement about the goal.

Arguments

`format` - [core_format](#)

<code>assignment_type</code> - String!	Goal assignment type, e.g. self, manager, etc.
<code>plugin_name</code> - String!	The goaltypesubplugin used to implement this goal.
<code>start_date</code> - core_date	The date the goal will commence.

Arguments

`format` - [core_date_format](#)

<code>target_type</code> - String	The type of the target for the goal, e.g. date, percentage, currency, number, manual.
<code>target_date</code> - core_date	The date the goal is expected to be completed/closed.

Arguments

`format` - [core_date_format](#)

Field Name	Description
<code>target_value</code> - <code>Float</code>	The expected value for achieving the goal.
<code>current_value</code> - <code>Float</code>	The value towards the goal currently achieved so far.
<code>current_value_updated_at</code> - <code>core_date</code>	The date the <code>current_value</code> was most recently updated.
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - <code>core_date_format</code> </div>	
<code>status</code> - <code>perform_goal_status</code>	The current status of the goal.
<code>status_updated_at</code> - <code>core_date</code>	The date the status was most recently updated.
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - <code>core_date_format</code> </div>	
<code>closed_at</code> - <code>core_date</code>	The date the goal was closed, or null if the goal is still active.
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - <code>core_date_format</code> </div>	
<code>created_at</code> - <code>core_date</code>	The date the goal record was first created.
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - <code>core_date_format</code> </div>	
<code>updated_at</code> - <code>core_date</code>	The date the goal record was last updated.
<div style="border: 1px solid #ccc; padding: 5px;"> Arguments <hr/> <code>format</code> - <code>core_date_format</code> </div>	
<code>comment_count</code> - <code>Int!</code>	The number of comments for the goal.
<code>goal_tasks_metadata</code> - <code>perform_goal_goal_tasks_metadata!</code>	Tasks meta information for this goal.

EXAMPLE

```
{
  "id": 123456,
```

```

"context_id": 4,
"category_id": 1,
"owner": {"id": 5, "username": "system_manager_user1"},
"user": {"id": 3, "username": "jane_smith"},
"name": "Show understanding of Sales methods",
"id_number": "sf_4067",
"description": "The learner must show knowledge of 5 sales methods.",
"assignment_type": "xyz789",
"plugin_name": "xyz789",
"start_date": "26/07/2023",
"target_type": "date",
"target_date": "26/08/2023",
"target_value": 100,
"current_value": 25,
"current_value_updated_at": "3/07/2023",
"status": {"id": "not_started", "label": "Not started"},
"status_updated_at": "26/08/2023",
"closed_at": "26/08/2023",
"created_at": "26/07/2023",
"updated_at": "26/08/2023",
"comment_count": 987,
"goal_tasks_metadata": perform_goal_goal_tasks_metadata
}

```

Types

perform_goal_goal_raw_data

Goal metadata from a goal create/retrieve/update/delete operation.

Field Name	Description
<code>available_statuses</code> - [perform_goal_status!]	Allowed goal status values.
<code>description</code> - <code>String</code>	Associated goal description.

Arguments

`format` - `core_format`

EXAMPLE

```

{
  "available_statuses": [perform_goal_status],
  "description": "xyz789"
}

```

Types

perform_goal_goal_task

Represents a goal task value.

Field Name	Description
<code>id</code> - core_id!	Internal database ID of the task.
<code>goal_id</code> - core_id!	Internal database ID of the parent goal.
<code>description</code> - String	Task description.
<div style="border: 1px solid #ccc; padding: 5px;"><p>Arguments</p><hr/><p><code>format</code> - core_format</p></div>	
<code>completed_at</code> - core_date	The date the task was completed.
<div style="border: 1px solid #ccc; padding: 5px;"><p>Arguments</p><hr/><p><code>format</code> - core_date_format</p></div>	
<code>created_at</code> - core_date!	The date the task was first created.
<div style="border: 1px solid #ccc; padding: 5px;"><p>Arguments</p><hr/><p><code>format</code> - core_date_format</p></div>	
<code>updated_at</code> - core_date	The date the task was last updated.
<div style="border: 1px solid #ccc; padding: 5px;"><p>Arguments</p><hr/><p><code>format</code> - core_date_format</p></div>	
<code>resource_exists</code> - Boolean	Indicates whether the associated resource (e.g. course) exists. If this is false and <code>resource_id</code> is not null, it means the resource originally existed but has now been deleted.
<code>resource_can_view</code> - Boolean!	Indicates whether the resource is viewable by the current user.
<code>resource</code> - perform_goal_goal_task_resource	Associated resource if any.

```

{
  "id": 123456,
  "goal_id": 123456,
  "description": "I have to study \"Information security\" course",
  "completed_at": "26/07/2023",
  "created_at": "26/07/2023",
  "updated_at": "26/08/2023",
  "resource_exists": true,
  "resource_can_view": true,
  "resource": perform_goal_goal_task_resource
}

```

Types

perform_goal_goal_task_resource

Represents a goal task resource.

Field Name	Description
<code>id</code> - core_id!	Internal database ID of the resource.
<code>task_id</code> - core_id!	Internal database ID of the parent task.
<code>resource_id</code> - core_id!	Points to the id in the table of the given resource type.
<code>resource_type</code> - Int!	Numerical code for the resource type.
<code>created_at</code> - core_date!	The date the resource was first created.

Arguments

`format` - [core_date_format](#)

`resource_custom_data` - [String](#) Extra data about the goal_task resource, e.g. a course resource.

EXAMPLE

```

{
  "id": 4,
  "task_id": 4,
  "resource_id": 4,
  "resource_type": 123,
  "created_at": "2022-04-17",
  "resource_custom_data": "xyz789"
}

```

Types

perform_goal_goal_tasks_metadata

Tasks meta information for a goal.

Field Name	Description
total_count - <code>Int!</code>	Total number of tasks for a goal.
completed_count - <code>Int!</code>	Number of completed tasks for a goal.

EXAMPLE

```
{"total_count": 123, "completed_count": 123}
```

Types

perform_goal_operation_error

Describes an error that occurred when doing a goal related query/mutation.

Field Name	Description
code - <code>String</code>	An error code representing this particular class of error.
message - <code>String</code>	Human readable string describing the error.

EXAMPLE

```
{  
  "code": "xyz789",  
  "message": "abc123"  
}
```

Types

perform_goal_permissions

Permissions for the current user to manipulate goals.

Field Name	Description
can_manage - Boolean!	Whether the current user can create/update/delete goals.
can_update_status - Boolean!	Whether the current user can update a goal's status.

EXAMPLE

```
{"can_manage": true, "can_update_status": true}
```

Types

perform_goal_reference

Input for identifying a goal.

The goal must be specified by providing one of the following:

- The goal's internal database id
- The user's id_number

A goal reference must uniquely identify a single goal to be valid.

Input Field	Description
id - core_id	The database ID of the goal.
id_number - String	An external identifier for the goal.

EXAMPLE

```
{"id": 4, "id_number": "xyz789"}
```

Types

perform_goal_status

Represents a goal status value.

Field Name	Description
id - String	Status code.
label - String	Status display label.

EXAMPLE

```
{"id": "not_started", "label": "Not started"}
```

Types

perform_goal_task_create_input

Input provided when creating a goal task. Some fields are required.

Input Field	Description
goal_id - core_id!	Internal database ID of the parent goal.
description - param_text	Goal task description.
resource_id - core_id	Points to the id in the table of the given resource type.
resource_type - Int	Numerical code for the resource type corresponding to constants in the back end code.

EXAMPLE

```
{  
  "goal_id": 4,  
  "description": "I have to study \"Information security\" course",  
  "resource_id": 4123,  
  "resource_type": 1  
}
```

Types

perform_goal_task_delete_result

The result returned from a goal task delete operation.

Field Name	Description
success - Boolean!	Whether the current goal task successfully deleted.

EXAMPLE

```
{"success": true}
```

Types

perform_goal_task_operation_error

Describes an error that occurred when doing a goal task related query/mutation.

Field Name	Description
code - <code>String</code>	An error code representing this particular class of error.
message - <code>String</code>	Human readable string describing the error.

EXAMPLE

```
{  
  "code": "xyz789",  
  "message": "abc123"  
}
```

Types

perform_goal_task_reference

Input for identifying a goal task.

The task must be specified by providing one of these:

- The task's internal database id (if present, always ignores the position value and goal id below).
- Its position in a parent goal's list. Note this value ranges from 0 (earliest created task) to <list size - 1> (latest created task).

Input Field	Description
goal_id - <code>core_id</code>	The database ID of the parent goal.
id - <code>core_id</code>	The database ID of the task.
ordinal - <code>Int</code>	The task's position in the parent goal's list.

EXAMPLE

```
{"goal_id": 4, "id": 4, "ordinal": 987}
```

Types

perform_goal_task_set_progress_input

Input provided when setting a goal task's progress.

Input Field	Description
goal_task_reference - perform_goal_task_reference!	Input value(s) for identifying a goal task.
completed - Int!	The task completion status: 0 = false, anything else = true.

EXAMPLE

```
{
  "goal_task_reference": perform_goal_task_reference,
  "completed": 1
}
```

Types

perform_goal_task_update_input

Input provided when updating a goal task.

Input Field	Description
goal_task_reference - perform_goal_task_reference!	Input value(s) for identifying a goal task.
description - param_text	Goal task description.
resource_id - core_id	Points to the id in the table of the given resource type.
resource_type - Int	Numerical code for the resource type corresponding to constants in the back end code.

EXAMPLE

```
{
  "goal_task_reference": perform_goal_task_reference,
  "description": "This is a string",
  "resource_id": 4,
  "resource_type": 987
}
```

Types

perform_goal_task_upsert_result

The result returned from a goal task create/update/set progress operation.

Field Name	Description
goal_task - perform_goal_goal_task	Task type object, containing information about a goal task.
success - Boolean!	Indicates if the operation succeeded.
errors - perform_goal_task_operation_error	Error messages if the operation failed.

EXAMPLE

```
{
  "goal_task": perform_goal_goal_task,
  "success": true,
  "errors": perform_goal_task_operation_error
}
```

Types

perform_goal_update_input

Input provided when updating a goal. No field is required.

Input Field	Description
name - String	The display name for the goal.
id_number - String	An external identifier for the goal.
description - String	Comments or a statement about the goal.
start_date - core_date	The date the goal will commence.
target_type - String	The type of the target for the goal, e.g. date, percentage, currency, number, manual.
target_date - core_date	The date the goal is expected to be completed/closed.
target_value - Float	The expected value for achieving the goal.

EXAMPLE

```
{
  "name": "Display knowledge of 5 sales methods",
  "id_number": "sf_10133",
  "description": "The learner must show knowledge of 5 sales methods.",
  "start_date": 1696289455,
```

```
"target_type": "date",
"target_date": 1701513055,
"target_value": 100
}
```

Types

perform_goal_update_progress_input

Values to update a goal's progress with.

Input Field	Description
current_value - Float!	The value towards the goal currently achieved so far.
status - String!	The current status of the goal.

EXAMPLE

```
{"current_value": 25, "status": "in_progress"}
```

Types

perform_goal_update_progress_result

Values returned from a perform_goal_update_progress operation.

Field Name	Description
current_value - Float	The value towards the goal currently achieved so far.
status - perform_goal_status!	The current status of the goal.

EXAMPLE

```
{"current_value": 123.45, "status": perform_goal_status}
```

Types

perform_goal_user_goals_filter

Filters input provided when querying available goals for a user.

Input Field	Description
user - core_user_user_reference	The user whose goals will be fetched. If left out, defaults to the logged-in user.
search - param_text	A search term for searching the goals
status - [param_alphaext!]	A list of status codes to filter for. The status codes correspond to the status class names.

EXAMPLE

```
{
  "user": core_user_user_reference,
  "search": "This is a string",
  "status": ["Ab_cd-e"]
}
```

Types

perform_goal_user_goals_input

Input provided when querying available goals for a user.

Input Field	Description
filters - perform_goal_user_goals_filter	Filters when querying available goals for a user.
options - perform_goal_user_goals_options	Options (e.g. sorting) when querying available goals for a user.
pagination - core_pagination_input	Standard pagination input.

EXAMPLE

```
{
  "filters": perform_goal_user_goals_filter,
  "options": perform_goal_user_goals_options,
  "pagination": core_pagination_input
}
```

Types

perform_goal_user_goals_options

Options input provided when querying available goals for a user.

Input Field	Description
<code>sort_by</code> - param_alphaext	Sort by a given field. Possible values are <code>created_at</code> , <code>target_date</code> , <code>most_complete</code> , <code>least_complete</code> .

EXAMPLE

```
{"sort_by": "Ab_cd-e"}
```

Types

perform_goal_user_goals_result

The result returned from a query to find the available goals for a user.

Field Name	Description
<code>items</code> - [perform_goal_user_goals_result_item]	Retrieved personal goals.
<code>total</code> - Int!	Total number of personal goals retrieved across all pages.
<code>next_cursor</code> - String!	Pointer to next page of personal goals.
<code>has_goals</code> - Boolean!	Flag indicating if the user has at least one goal (independent of filter input).
<code>success</code> - Boolean!	Indicates if the operation succeeded.
<code>errors</code> - perform_goal_operation_error	Error messages if the operation failed.

EXAMPLE

```
{
  "items": [perform_goal_user_goals_result_item],
  "total": 123,
  "next_cursor": "abc123",
  "has_goals": true,
  "success": true,
  "errors": perform_goal_operation_error
}
```

Types

perform_goal_user_goals_result_item

Information on one goal.

Field Name	Description
goal - perform_goal_goal!	Retrieved goal.
permissions - perform_goal_permissions!	Permissions for the goal with respect to the currently logged-in user.

EXAMPLE

```
{
  "goal": perform_goal_goal,
  "permissions": perform_goal_permissions
}
```

Types

perform_goal_view_result

The result returned from a goal view operation.

Field Name	Description
goal - perform_goal_goal	Goal type object, containing information about a personal or team goal.
raw - perform_goal_goal_raw_data	Goal metadata from a goal retrieve operation.
permissions - perform_goal_permissions	Permissions for the current user to manipulate goals.

EXAMPLE

```
{
  "goal": perform_goal_goal,
  "raw": perform_goal_goal_raw_data,
  "permissions": perform_goal_permissions
}
```

Troubleshooting diagnostics

Queries

tool_diagnostic_get_providers

Get all providers.

Response

Returns `[tool_diagnostic_provider!]!`

QUERY

```
query tool_diagnostic_get_providers {
  tool_diagnostic_get_providers {
    id
    name
    description
    enabled
    whitelist
  }
}
```

RESPONSE

```
{
  "data": {
    "tool_diagnostic_get_providers": [
      {
        "id": "xyz789",
        "name": "abc123",
        "description": "abc123",
        "enabled": true,
        "whitelist": "xyz789"
      }
    ]
  }
}
```

Mutations

`tool_diagnostic_run_diagnostics`

Run providers and return location to file created.

Response

Returns a `tool_diagnostic_run_diagnostics_result!`

Arguments

Name	Description
------	-------------

excluded_provider_ids -
[param_alphanumext!]

QUERY

```
mutation tool_diagnostic_run_diagnostics($excluded_provider_ids: [param_alphanumext!]!){
  tool_diagnostic_run_diagnostics(excluded_provider_ids: $excluded_provider_ids){
    download_url
    download_filesize
  }
}
```

VARIABLES

```
{"excluded_provider_ids": ["Abc_123-0"]}
```

RESPONSE

```
{
  "data": {
    "tool_diagnostic_run_diagnostics": {
      "download_url": "abc123",
      "download_filesize": 123
    }
  }
}
```

Types

tool_diagnostic_provider

Information about a provider.

Field Name	Description
id	- String!
name	- String!
description	- String!
enabled	- Boolean!

Field Name

Description

whitelist - [String!](#)

EXAMPLE

```
{
  "id": "xyz789",
  "name": "xyz789",
  "description": "xyz789",
  "enabled": true,
  "whitelist": "abc123"
}
```

Types

tool_diagnostic_run_diagnostics_result

Result about running the diagnostics

Field Name

Description

download_url - [String!](#)

download_filesize - [Int!](#)

EXAMPLE

```
{
  "download_url": "xyz789",
  "download_filesize": 123
}
```

User

Queries

core_user_user

Return a specific user in the system.

Response

Returns a [core_user_user_result!](#)

Arguments

Name	Description
reference - core_user_user_reference!	Input object specifying data for the existing user.

QUERY

```
query core_user_user($reference: core_user_user_reference!) {
  core_user_user(reference: $reference) {
    user {
      ...core_userFragment
    }
  }
}
```

VARIABLES

```
{"reference": core_user_user_reference}
```

RESPONSE

```
{
  "data": {
    "core_user_user": {
      "user": {
        "id": 4,
        "username": "jsmith",
        "email": "jsmith@example.com",
        "firstname": "John",
        "lastname": "Smith",
        "custom_fields": [
          {"shortname": "employeeid", "data": "EM12345", "data_type": "TEXT"}
        ]
      }
    }
  }
}
```

Queries

core_user_users

Return a paginated list of users in the system. Deleted users are excluded and multitenancy restrictions apply. By default, only active (non-suspended) users are included.

Response

Returns a `core_user_users_result!`

Arguments

Name	Description
query - <code>core_user_users_query</code>	Sort and pagination information to control the data returned.

QUERY

```
query core_user_users($query: core_user_users_query) {  
  core_user_users(query: $query) {  
    items {  
      ...core_userFragment  
    }  
    total  
    next_cursor  
  }  
}
```

VARIABLES

```
{"query": core_user_users_query}
```

RESPONSE

```
{  
  "data": {  
    "core_user_users": {  
      "items": [  
        {  
          "username": "test",  
          "firstname": "Test",  
          "lastname": "User",  
          "email": "testuser@example.com"  
        },  
        {  
          "username": "another",  
          "firstname": "Another",  
          "lastname": "User",  
          "email": "anotheruser@example.com"  
        }  
      ],  
      "total": 47,  
      "next_cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319"
```

```
}  
  }  
}
```

Mutations

core_user_create_user

Create a new user.

Response

Returns a [core_user_user_result!](#)

Arguments

Name	Description
input - core_user_create_user_input!	Input object specifying data for the new user.

QUERY

```
mutation core_user_create_user($input: core_user_create_user_input!) {  
  core_user_create_user(input: $input) {  
    user {  
      ...core_userFragment  
    }  
  }  
}
```

VARIABLES

```
{"input": core_user_create_user_input}
```

RESPONSE

```
{  
  "data": {  
    "core_user_create_user": {  
      "user": {  
        "id": 4,  
        "username": "jsmith",  
        "email": "jsmith@example.com",  
        "firstname": "John",  
        "lastname": "Smith",  
      }  
    }  
  }  
}
```

```
    "custom_fields": [
      { "shortname": "employeeid", "data": "EM12345", "data_type": "TEXT" }
    ]
  }
}
```

Mutations

core_user_delete_user

Delete the target user. Note: a service account making a request cannot delete its own user account.

Response

Returns a [core_user_delete_user_result!](#)

Arguments

Name	Description
target_user - core_user_user_reference!	Target user which is being deleted.

QUERY

```
mutation core_user_delete_user($target_user: core_user_user_reference!) {
  core_user_delete_user(target_user: $target_user) {
    user_id
  }
}
```

VARIABLES

```
{"target_user": core_user_user_reference}
```

RESPONSE

```
{"data": {"core_user_delete_user": {"user_id": 4}}}
```

Mutations

core_user_update_user

Update the specified target user with new properties. Note: a service account making a request cannot suspend its own user account.

Response

Returns a [core_user_user_result!](#)

Arguments

Name	Description
target_user - core_user_user_reference!	User to be updated.
input - core_user_update_user_input!	Input object specifying data for updating the user.

QUERY

```
mutation core_user_update_user(  
  $target_user: core_user_user_reference!,  
  $input: core_user_update_user_input!  
) {  
  core_user_update_user(  
    target_user: $target_user,  
    input: $input  
  ) {  
    user {  
      ...core_userFragment  
    }  
  }  
}
```

VARIABLES

```
{  
  "target_user": core_user_user_reference,  
  "input": core_user_update_user_input  
}
```

RESPONSE

```
{  
  "data": {  
    "core_user_update_user": {  
      "user": {  
        "id": 4,  

```

```

    "username": "jsmith",
    "email": "jsmith@example.com",
    "firstname": "John",
    "lastname": "Smith",
    "custom_fields": [
      {"shortname": "employeeid", "data": "EM12345", "data_type": "TEXT"}
    ]
  }
}
}
}
}

```

Types

core_user

User type object, containing information about an individual user.

Field Name	Description
<code>id</code> - core_id!	Internal database id of the user.
<code>idnumber</code> - String	The id number of the user.
<code>username</code> - String	Username for the user. This field is used in combination with the password for logging in with some authentication types.

Arguments

`format` - [core_format](#)

Output format this field should be formatted for.

<code>fullname</code> - String!	Full name of the user. This field combines other name fields as specified by the site display settings.
-------------------------------------------------	---------------------------------------------------------------------------------------------------------

Arguments

`format` - [core_format](#)

<code>firstname</code> - String	First name of the user.
-------------------------------------------------	-------------------------

Arguments

`format` - [core_format](#)

Field Name	Description
Output format this field should be formatted for.	
lastname - String	Last name of the user.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
middlename - String	Middle name of the user.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
alternatename - String	Alternate name of the user.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
firstnamephonetic - String	Phonetic spelling of the user's first name.
lastnamephonetic - String	Phonetic spelling of the user's last name.
email - String	Email address of the user.
address - String	The user's address.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
phone1 - String	Primary phone number of the user.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
phone2 - String	Secondary phone number of the user.

Field Name	Description
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
<p>department - String</p>	Department the user belongs to.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
<p>institution - String</p>	Institution (organisation) the user belongs to.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
<p>city - String</p>	City where the user is located.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for.</p>	
<p>country - String</p>	Country of the user, displayed as a string, for example 'New Zealand'.
<p>description - String</p>	Description field for the user's profile.
<p>Arguments</p> <hr/> <p>format - core_format</p> <p>Output format this field should be formatted for. Note: if the system setting 'profilesforenrolleduseronly' is enabled, the profile description of a user who is not yet enrolled in any course will be hidden. Description could also be hidden until a user has at least one role assignment. (This is to prevent misuse of user profile information by spammers.)</p>	
<p>descriptionformat - core_format</p>	Format of the description field, as stored in the database.
<p>profileurl - String</p>	Profile url can be null if the viewer cannot view the target user's profile.

Field Name	Description
<code>profileimageurl</code> - String	URL pointing to the main version of the user's profile image.
<code>profileimageurlsmall</code> - String	URL pointing to a small version of the user's profile image.
<code>profileimagealt</code> - String	Alternate text description of the profile image, for accessibility purposes.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - core_format Output format this field should be formatted for.</p> </div>	
<code>lang</code> - String	Language of the user, for example 'en_us'.
<code>theme</code> - String	Theme setting for the user or an empty string if the user is using the site default theme.
<code>suspended</code> - Boolean	Whether this user is marked as suspended. Suspended users are unable to log in.
<code>timezone</code> - String	<p>Timezone of the user, as a string, for example 'Pacific/Auckland'.</p> <p>Note that if the admin setting 'forcetimezone' is configured, its value will be used here instead.</p>
<code>interests</code> - String	Comma-separated string of interests the user has or null if no interests are assigned.
<code>firstaccess</code> - core_date	Timestamp of when this user first accessed the site (or null if never).
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - core_date_format Date format that firstaccess should be returned in.</p> </div>	
<code>lastaccess</code> - core_date	Timestamp of when this user last accessed the site (or null if never).
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format</code> - core_date_format Date format that lastaccess should be returned in.</p> </div>	
<code>url</code> - String	URL for the user's profile.
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> </div>	

Field Name	Description
format - <code>core_format</code>	Output format this field should be formatted for.
skype - <code>String</code>	Skype username for the user's profile.
Arguments	
format - <code>core_format</code>	Output format this field should be formatted for.
custom_fields - <code>[core_user_custom_field!]</code>	Custom fields are additional fields that can be defined by a Site Administrator to allow additional specific data to be connected to users. Hidden and restricted custom fields require 'moodle/user:viewalldetails' capability.
job_assignments - <code>[totara_job_job_assignment!]</code>	This user's job assignments.

EXAMPLE

```
{
  "id": 4,
  "idnumber": "RW1234",
  "username": "rebecca.woodhouse",
  "fullname": "Rebecca Woodhouse",
  "firstname": "Rebecca",
  "lastname": "Woodhouse",
  "middlename": "Jane",
  "alternatename": "Becky",
  "firstnamephonetic": "xyz789",
  "lastnamephonetic": "xyz789",
  "email": "rebecca.woodhouse@example.com",
  "address": "1 Civic Square, London, England",
  "phone1": "+64 4 555 1234",
  "phone2": "+64 21 555 1234",
  "department": "Marketing",
  "institution": "Acme Inc.",
  "city": "London",
  "country": "United Kingdom",
  "description": "I have an interest in agile project management and softw",
  "descriptionformat": "PLAIN",
  "profileurl": "xyz789",
  "profileimageurl": "http://localhost/totara/server/theme/image.php/ventu",
  "profileimageurlsmall": "http://localhost/totara/server/theme/image.php",
  "profileimagealt": "Picture of Rebecca Woodhouse",
  "lang": "en",
}
```

```

"theme": "ventura",
"suspended": true,
"timezone": "Europe/London",
"interests": "xyz789",
"firstaccess": "2022-04-17",
"lastaccess": "2022-04-17",
"url": "https://www.example.com/~rwoodhouse/",
"skype": "rebecca.woodhouse",
"custom_fields": [
  {"shortname": "worklocation", "data": "Remote", "data_type": "TEXT"},
  {"shortname": "employeeid", "data": "456789", "data_type": "TEXT"}
],
"job_assignments": [
  {
    "idnumber": "my_id1243",
    "fullname": "Sales Manager",
    "shortname": "SM1",
    "description": "This is a description",
    "userid": 1234567,
    "positionid": 543421
  }
]
}

```

Types

core_user_create_user_input

Input provided when creating a user.

Input Field	Description
username - <code>String!</code>	Username for the user. This field is used in combination with the password for logging in with some authentication types. Username must be unique on the site.
email - <code>param_email!</code>	Email address of the user. Must be unique unless 'allowaccountssameemail' setting is enabled (not recommended).
password - <code>String</code>	Initial password for user. This will only be used with certain internal auth methods such as 'manual'. This field is required unless you set <code>generate_password</code> to true OR use external auth methods for your site.
firstname - <code>String!</code>	First name of the user.
lastname - <code>String!</code>	Last name of the user.

Input Field	Description
idnumber - String	The id number of the user.
firstnamephonetic - String	Optional phonetic spelling of the user's first name.
lastnamephonetic - String	Optional phonetic spelling of the user's last name.
middlename - String	Middle name of the user.
alternatename - String	Alternate name of the user.
city - String	City where the user is located. If not provided the system will use the admin setting 'defaultcity' if set.
description - String	Rich-text description field for the user's profile.
descriptionformat - core_format	Format of the description field for the user's profile. See core_format for options.
url - String	URL for the user's profile.
skype - String	Skype username for the user's profile.
institution - String	Institution (organisation) the user belongs to, for the user's profile. This is not connected to job assignment organisation.
department - String	Department the user belongs to.
phone1 - String	Primary phone number of the user.
phone2 - String	Secondary phone number of the user.
address - String	The user's address.
country - String	ISO-3166 two-letter country code of the user, e.g. 'NZ'. Country code list: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2 .
timezone - String	<p>Timezone of the user, as a string, for example 'Pacific/Auckland'. If not provided the value from the admin setting 'timezone' will be used instead. We recommend using location-based timezones rather than UTC+1 etc. A list of available timezones can be found at https://www.php.net/manual/en/timezones.php.</p> <p>Note that if the admin setting 'forcetimezone' is configured, any value provided here will be stored but not displayed/used - the forcetimezone value will be used instead.</p>
lang - String	Language of the user, for example 'en_us'. Valid options depend on the site's installed language packs.

Input Field	Description
theme - String	Theme setting for the user. If the Totara system setting for 'allow user themes' is enabled, then a valid theme name will be accepted from this request, e.g. 'ventura'. If the system setting is disabled, then the theme value will be ignored.
auth - String	Authentication plugin for this user, for example 'manual'. The available options will depend on which authentication plugins are installed and enabled on the site.
calendartype - String	Calendar type for this user, for example 'gregorian'. The available options will depend on which calendar plugins are installed and enabled on the site.
emailstop - Boolean	Whether this user should be completely prevented from receiving any email notifications. Default is false.
suspended - Boolean	Whether this user should be marked as suspended. Suspended users are unable to log in. Default is false.
tenant - totara_tenant_tenant_reference	A reference to a tenant that the user should be assigned to. The API client must be permitted to assign the specified tenant. If not given the user will be automatically assigned based on the API client's tenant, or to the System. Ignored if multitenancy is disabled.
custom_fields - [core_user_custom_field_input!]	Optional array of custom profile field data (each array element describes a single custom field value for this user). Hidden and restricted custom fields require 'moodle/user:viewalldetails' capability.
force_password_change - Boolean	Prompt the user to change their password on their next login.
generate_password - Boolean	Generate a random password for the user and send it to them via email.

EXAMPLE

```
{
  "username": "rebecca.woodhouse",
  "email": "rebecca.woodhouse@example.com",
  "password": "supersecret",
  "firstname": "Rebecca",
  "lastname": "Woodhouse",
  "idnumber": "RW1234",
  "firstnamephonetic": "reh BEH kuh",
  "lastnamephonetic": "WUUD hows",
```

```

"middlename": "Jane",
"alternatename": "Becky",
"city": "London",
"description": "I have an interest in agile project management and softw
"descriptionformat": "PLAIN",
"url": "https://www.example.com/~rwoodhouse/",
"skype": "rebecca.woodhouse",
"institution": "Acme Inc.",
"department": "Marketing",
"phone1": "+64 4 555 1234",
"phone2": "+64 21 555 1234",
"address": "1 Civic Square, London, England",
"country": "GB",
"timezone": "Europe/London",
"lang": "en",
"theme": "ventura",
"auth": "manual",
"calendartype": "gregorian",
"emailstop": true,
"suspended": true,
"tenant": {"idnumber": "tenant1"},
"custom_fields": [
  {"shortname": "worklocation", "data": "Remote", "data_type": "TEXT"},
  {"shortname": "employeeid", "data": "456789", "data_type": "TEXT"}
],
"force_password_change": true,
"generate_password": true
}

```

Types

core_user_custom_field

Type containing read-only custom field data. Each object of this type represents the value of the field. The format of this string will depend on the custom field type it is representing but will contain the field's stored value.

Field Name	Description
shortname - String!	Shortname of the custom field the data relates to.
data - String!	The custom field data value for this user. Returns the value stored in the database for this user and field, without any processing (such as inserting field default when value is empty, or converting of timestamps to human-readable strings).
data_format - Int	Format of the stored value of the data field. Provided at the time the data is stored, see the <code>core_user_custom_field_input</code> input type

Field Name	Description
	<p>for more information. Currently only used by the text area custom field type when rendering in the UI.</p> <p>Options are:</p> <ul style="list-style-type: none"> • 1: HTML • 2: Plain text • 4: Markdown • 5: JSON
data_type - <code>String!</code>	Indicating data type of user profile custom field.

EXAMPLE

```
{
  "shortname": "employeeid",
  "data": "EM12345",
  "data_format": 1,
  "data_type": "TEXT"
}
```

Types

core_user_custom_field_input

Input type representing a single custom field for a particular user.

Input Field	Description
shortname - <code>String!</code>	<p>Shortname is used to uniquely identify the custom field the data is for. It must exactly match the shortname of a custom field that exists on the site or an error will be thrown. Note if the same shortname is provided multiple times in a request (not recommended) they will be applied in order so the last one will be used.</p>
data - <code>String</code>	<p>Represents the value for the custom field for this user. This field is required unless the delete flag is set. The format of this data will depend on the field type of the field specified by shortname:</p> <ul style="list-style-type: none"> • checkbox: string of 0 (unchecked) or 1 (checked). • menu: A string exactly matching one of the menu options (case sensitive). • text: A string of plain-text.

Input Field

Description

- `textarea`: A string of formatted text, in the format specified via the `data_format` field (see below).
- `datetime`: A string in the format `YYYY-MM-DD` or `YYYY-MM-DD-HH-MM-SS` (depending on if the field uses the 'Include time?' option). If the field defines a year range, the input will be validated to ensure it falls within that range. If only `YYYY-MM-DD` is provided for a field where 'Include time?' is enabled, midnight will be used for the time. Stored as a unix timestamp using the timezone of the client's service account.
- `date`: A string in the format `YYYY-MM-DD`. Stored as a timestamp so uses noon UTC for time to ensure date displayed is consistent across most timezones.

Optional property, which must be provided `textarea` field types but should not be passed for other types. Used to record the format of the 'data' field so it can be correctly rendered. Options are:

`data_format` - [Int](#)

- 1: HTML
- 2: Plain-text
- 4: Markdown
- 5: JSON

`delete` - [Boolean](#)

Flag indicating that the data for this custom field should be deleted. This flag should only be set if not passing new or updated data for the field.

EXAMPLE

```
{"shortname": "employeeid", "data": "EM12345", "data_format": 1, "delete"
```

Types

`core_user_delete_user_result`

Mutation result type returned after deletion of a user.

Returns the id of deleted user if it was deleted successfully.

Field Name	Description
<code>user_id - core_id!</code>	Internal database id of the user that was just deleted from the system.

EXAMPLE

```
{"user_id": 4}
```

Types

core_user_enrolment

Represents user enrolment data.

Field Name	Description
<code>id - core_id!</code>	The unique identifier for the user enrolment.
<code>enrolment - core_enrol!</code>	The enrolment instance.
<code>status - core_user_enrolment_status_enum!</code>	The status of the user enrolment.
<code>timestart - core_date!</code>	The start date for the enrolment (in the specified format, defaulting to TIMESTAMP).
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format - core_date_format</code></p> </div>	
<code>timeend - core_date</code>	The end date for the enrolment (in the specified format, defaulting to TIMESTAMP).
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format - core_date_format</code></p> </div>	
<code>timecreated - core_date!</code>	The creation time for the enrolment (in the specified format, defaulting to TIMESTAMP).
<div style="border: 1px solid #ccc; padding: 5px;"> <p>Arguments</p> <hr/> <p><code>format - core_date_format</code></p> </div>	
<code>timemodified - core_date!</code>	The modification time for the enrolment (in the specified format, defaulting to TIMESTAMP).

Arguments

Field Name	Description
format - core_date_format	
modified_by - core_user!	The user who last modified the enrolment.

EXAMPLE

```
{
  "id": 9,
  "enrolment": core_enrol,
  "status": "ENROL_USER_ACTIVE",
  "timestart": "1698226645",
  "timeend": "1698226645",
  "timecreated": "1698226645",
  "timemodified": "1698226645",
  "modified_by": {"id": 7, "idnumber": "abc"}
}
```

Types

core_user_enrolment_status_enum

Enumerates the possible user enrolment status values.

Enum Value	Description
ENROL_USER_ACTIVE	The user enrolment is currently active.
ENROL_USER_SUSPENDED	The user enrolment is currently suspended.

EXAMPLE

```
"ENROL_USER_ACTIVE"
```

Types

core_user_update_user_input

Input provided when updating a user.

Input Field	Description
username - param_username	Username for the user. This field is used in combination with the password for logging in with

Input Field	Description
	some authentication types. Username must be unique on the site.
firstname - String	First name of the user.
lastname - String	Last name of the user.
email - param_email	Email address of the user. Must be unique unless 'allowaccountssameemail' admin setting is enabled.
password - String	New password for the user. This will only be used with certain auth methods such as 'manual'.
idnumber - String	The id number of the user.
firstnamephonetic - String	Optional phonetic spelling of the user's first name.
lastnamephonetic - String	Optional phonetic spelling of the user's last name.
middlename - String	Middle name of the user.
alternatename - String	Alternate name of the user.
city - String	City where the user is located. If not provided the system will use the admin setting 'defaultcity' if set.
description - String	Rich-text description field for the user's profile.
descriptionformat - core_format	Rich-text format of the description field.
url - String	URL for the user's profile.
skype - String	Skype username for the user's profile.
institution - String	Institution (organisation) the user belongs to, for the user's profile. This is not connected to job assignment organisation.
department - String	Department the user belongs to.
phone1 - String	Primary phone number of the user.
phone2 - String	Secondary phone number of the user.
address - String	The user's address.
country - String	ISO-3166 two-letter country code of the user, e.g. 'NZ'. Country code list: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2 .
timezone - String	Timezone of the user, as a string, for example 'Pacific/Auckland'. We recommend using location-based timezones rather than UTC+1 etc. A list of available timezones can be found at https://www.php.net/manual/en/timezones.php .

Input Field	Description
	Note that if the admin setting 'forcetimezone' is configured, any value provided here will be stored but not displayed/used - the forcetimezone value will be used instead.
lang - String	Language of the user, for example 'en_us'. Valid options depend on the site's installed language packs.
theme - String	Theme setting for the user. If the admin setting 'allowuserthemes' is enabled, then a valid theme name will be accepted from this request, e.g. 'ventura'. If that admin setting is disabled, then the theme value will be ignored.
auth - String	Authentication plugin for this user, for example 'manual'. The available options will depend on which authentication plugins are installed and enabled on the site.
calendartype - String	Calendar type for this user, for example 'gregorian'. The available options will depend on which calendar plugins are installed and enabled on the site.
emailstop - Boolean	Whether this user should be completely prevented from receiving any email notifications. Default is false.
suspended - Boolean	Whether this user should be marked as suspended. Suspended users are unable to log in. Default is false.
tenant - totara_tenant_tenant_reference	A reference to a tenant that the user should be assigned to. The API client must be permitted to assign the specified tenant. To change tenant the API client must have 'totara/tenant:manageparticipants' capability in system context.
custom_fields - [core_user_custom_field_input!]	Optional array of custom profile field data (each array element describes a single custom field value for this user). Hidden and restricted custom fields require 'moodle/user:viewalldetails' capability.
force_password_change - Boolean	Prompt the user to change their password on their next login.
generate_password - Boolean	Generate a random password for the user and send it to them via email.

EXAMPLE

```
{
  "username": "rebecca.woodhouse",
```

```
"firstname": "Rebecca",
"lastname": "Woodhouse",
"email": "rebecca.woodhouse@example.com",
"password": "supersecret",
"idnumber": "RW1234",
"firstnamephonetic": "reh BEH kuh",
"lastnamephonetic": "WUUD hows",
"middlename": "Jane",
"alternatename": "Becky",
"city": "London",
"description": "I have an interest in agile project management and softw",
"descriptionformat": "PLAIN",
"url": "https://www.example.com/~rwoodhouse/",
"skype": "rebecca.woodhouse",
"institution": "Acme Inc.",
"department": "Marketing",
"phone1": "+64 4 555 1234",
"phone2": "+64 21 555 1234",
"address": "1 Civic Square, London, England",
"country": "GB",
"timezone": "Europe/London",
"lang": "en",
"theme": "ventura",
"auth": "manual",
"calendartype": "gregorian",
"emailstop": true,
"suspended": true,
"tenant": {"idnumber": "tenant1"},
"custom_fields": [
  {"shortname": "worklocation", "data": "Remote", "data_type": "TEXT"},
  {"shortname": "employeeid", "data": "456789", "data_type": "TEXT"}
],
"force_password_change": true,
"generate_password": true
}
```

Types

core_user_user_reference

Input for identifying a user.

The user must be specified by providing one of the following:

- The user's internal database id
- The user's idnumber
- The user's username
- The user's email address

A user reference must uniquely identify a single user to be valid. Deleted users are excluded, as are users not accessible due to multitenancy restrictions.

Input Field	Description
<code>id - core_id</code>	Internal database id of the user.
<code>username - String</code>	Username of the user.
<code>idnumber - String</code>	The id number of the user.
<code>email - String</code>	Email address of the user.

EXAMPLE

```
{
  "id": 5,
  "username": "target.user",
  "idnumber": "EM12345",
  "email": "target.user@example.com"
}
```

Types

`core_user_user_result`

User result type returned after the creation of a new user, updates to an existing user, or from selecting an existing user. Returns the user information.

Field Name	Description
<code>user - core_user!</code>	User object for the user that was created or updated as part of the operation.

EXAMPLE

```
{
  "user": {
    "id": 4,
    "username": "jsmith",
    "email": "jsmith@example.com",
    "firstname": "John",
    "lastname": "Smith",
    "custom_fields": [
      {"shortname": "employeeid", "data": "EM12345", "data_type": "TEXT"}
    ]
  }
}
```

```
}  
}
```

Types

core_user_user_status

Enum for user status filter.

Option ALL indicates all statuses.

Enum Value	Description
ALL	
ACTIVE	
SUSPENDED	

EXAMPLE

```
"ALL"
```

Types

core_user_users_filters

Input type used when querying for a list of users. Specifies filters to be applied to the results.

Input Field	Description
status - core_user_user_status	The status of a user. By default, only ACTIVE users are returned. Deleted users are never returned.
since_timecreated - core_date	Unix creation timestamp of tag including an integer timestamp, numeric string or an ISO-8601 string.
since_timemodified - core_date	Unix timemodification timestamp of tag including an integer timestamp, numeric string or an ISO-8601 string.

EXAMPLE

```
{  
  "status": "SUSPENDED",
```

```
"since_timecreated": "1696837540",
"since_timemodified": "1696837540"
}
```

Types

core_user_users_query

Input type used when querying for a list of users. Specifies pagination and sorting information that can impact the structure of the results.

Input Field	Description
pagination - core_pagination_input	Pagination information such as which page to return and the number of results requested.
sort - [core_sort_input!]	The sort order of the query. Allowed entity fields for the sort column are 'id', 'firstname', 'lastname', 'username' and 'timemodified'.
filters - core_user_users_filters	The filters for querying users.

EXAMPLE

```
{
  "pagination": {
    "cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319",
    "limit": 20
  },
  "sort": [{"column": "timemodified", "direction": "ASC"}],
  "filters": core_user_users_filters
}
```

Types

core_user_users_result

Result returned from the core_user_users query. Contains a page of results along with pagination information.

Field Name	Description
items - [core_user!]!	Array of one page of users returned by the query.
total - Int!	Total number of users from this query (across all pages).
next_cursor - String!	Cursor to request the next set of results for this query.

EXAMPLE

```
{
  "items": [
    {
      "username": "test",
      "firstname": "Test",
      "lastname": "User",
      "email": "testuser@example.com"
    },
    {
      "username": "another",
      "firstname": "Another",
      "lastname": "User",
      "email": "anotheruser@example.com"
    }
  ],
  "total": 47,
  "next_cursor": "eyJsaW1pdCI6MSwiY29sdW1ucyI6eyJpZCI6M319"
}
```

Web API

Queries

totara_webapi_status

Simple query returning "ok" to test that you are able to successfully execute GraphQL queries.

Response

Returns a [totara_webapi_status](#)

QUERY

```
query totara_webapi_status {
  totara_webapi_status {
    status
    timestamp
  }
}
```

RESPONSE

```
{
  "data": {
    "totara_webapi_status": {
      "status": "ok",
      "timestamp": "2022-04-17"
    }
  }
}
```

Types

totara_webapi_status

Type containing information about the current status of the API. status is always 'ok'. This is intended to test that you are able to successfully execute GraphQL queries.

Field Name	Description
status - String!	The status of the API (always returns 'ok' if working).
timestamp - core_date	Timestamp of the request.

Arguments

format - [core_date_format](#)

Format of the timestamp in the response.

EXAMPLE

```
{"status": "ok", "timestamp": "2022-04-17"}
```

Documentation by [Anvil SpectaQL](#)

[Purge all caches](#)

Powered by [Totara](#)