Tool Sync: Web services API

Remote control service of the component

Web services allow external software components to interact with the component.

tool_sync_set_config(service, confkey, confvalue)

Synopsis: This function lets you remotely modify any configuration parameter of the synchronisation tool.

- **service**: Synchronisation service identifier (courses, users, enrols, cohorts)
- **confkey**: Configuration key
- **confvalue**: Configuration value

Service configuration tables

Service: courses

Key	Values
fileuploadlocation	file name
filedeletelocation	file name
filedeleteidentifier	id, shortname, idnumber
fileexistlocation	file name
existfileidentifier	id, shortname, idnumber
fileresetlocation	file name
fileresetidentifier	id, shortname, idnumber
forceupdate	0 or 1

Service: users

Key	Values
filelocation	file name
primaryidentity	id, idnumber, username, email

Service: enrols

Key	Values
filelocation	file name
courseidentifier	id, idnumber, shortname

Key	Values
useridentifier	id, idnumber, username, email

Service: cohorts

Key	Values
filelocation	file name
useridentifier	id, idnumber, username, email
cohortidentifier	id, idnumber
autocreate	0 or 1
syncdelete	0 or 1

tool_sync_commit_file(draftitemid)

Synopsis: This function moves a file previously uploaded to the draft area to its final location in the synchronisation tool's file system.

- **draftitemid**: The draft area identifier retrieved during the last call to the upload web service.

Example implementation in PHP:

```
$path = '<location_of_the_file>';
$params = array('token' => '<valid_token>',
                'itemid' => 0,
                'filearea' => 'draft');
$uploadurl = '<moodle_url>/webservice/upload.php';
$ch = curl_init($uploadurl);
$curlfile = new CURLFile($path, 'x-application/zip', basename($path));
$params['resourcefile'] = $curlfile;
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
echo "Firing CUrl $uploadurl ... \n";
if (!$result = curl_exec($ch)) {
    echo "CURL Error : ".curl_error($ch)."\n";
    return;
}
$result = json_decode($result);
$filerec = array_pop($result);
// Now commit the file.
$params = array('wstoken' => '<valid_token>',
                'wsfunction' => 'tool_sync_commit_file',
                'moodlewsrestformat' => 'json',
                'draftitemid' => $filerec->itemid);
$commiturl = '<moodle_url>/webservice/rest/server.php';
$this->send($commiturl, $params);
```

tool_synch_process(service, action)

Synopsis: Triggers the appropriate synchronisation processing based on the current configuration of the tool.

- **service**: Synchronisation service identifier ('courses', 'enrols', 'users' or 'cohorts')
- **action**: Operation identifier (valid for courses: 'create', 'delete', 'reset', otherwise 'sync' for all other services).

tool_sync_deploy_course(categorysourceid, categoryid, templatesourceid, templateid, shortname, fullname, idnumber, replacements)

Synopsis: Deploys a course based on an existing course archive or a backup available in the file system.

- **categorysourceid**: Source of the category identifier (can be 'id' or 'idnumber')
- **categoryid**: Target category identifier
- **templateidsource**: Source of the template course identifier (can be 'id', 'shortname', 'idnumber')
- **templateid**: Template course identifier. If this identifier cannot be treated as an identifier, but rather as a path in the file system, then the archive matching this path will be considered as the source of the course data.
- **shortname**: Short name of the course to create
- **fullname**: Full name of the course to create
- **idnumber**: External identifier of the course to create
- **replacements**: A JSON-serialized structure containing replacement indicators to apply in various fields and objects of the course. (*)

(*) *Syntax under experimentation / documentation to be developed*

Example of an external trigger sequence

For an operation driven from the outside, here is the complete sequence of operations to perform:

- Download of the file through the “standard” Moodle upload
- Commit of the retrieved draftitemid (tool_sync_commit) to move it from the draft area to the file container of the tool sync
- Modification of the configuration (if the file name has changed, otherwise the config is already ready to run) (tool_sync_set_config, for the desired service) this will reset the file name of one of the “locations” for the right service and the right action. The WS can modify all configuration keys of tool_sync
- Launch of the matching tool_sync_process for the desired service, with the saved configuration.

Unification service for primary business object manoeuvres

This service is intended to complete or take over from the standard Moodle web services, when the latter lack API unification, in particular regarding the primary identifiers of Moodle's main business

objects.

This service plan will add missing functions to the standard API, or offer an alternative signature for functions that do not make it possible to use external or alternative identifiers on the business objects to be handled.

In general, it is agreed to be able to provide all possible identification entries on the following bases:

Entity	Possible identifiers

Course | id, shortname, idnumber | User | id, username(@mnethostid), idnumber, email | Cohort | id, idnumber | Course group | id, CourseID@name | Role | id, shortname |

Currently implemented

- tool_sync_enrol_user_enrol()
- tool_sync_enrol_user_unenrol()

Back to the technical guide - Back to the component index - Back to the catalogue

From:
https://docsen.activeprolearn.com/ - **Documentation Moodle ActiveProLearn**

Permanent link:
https://docsen.activeprolearn.com/doku.php?id=tool:sync:webservices

Last update: **2026/11/13 22:01**