

Moodle-integrálás intézményi környezetben

Dr. Tornóci László, Dr. Kokovay Ágnes

Semmelweis Egyetem
E-learning és Digitális Tartalomfejlesztő Igazgatóság

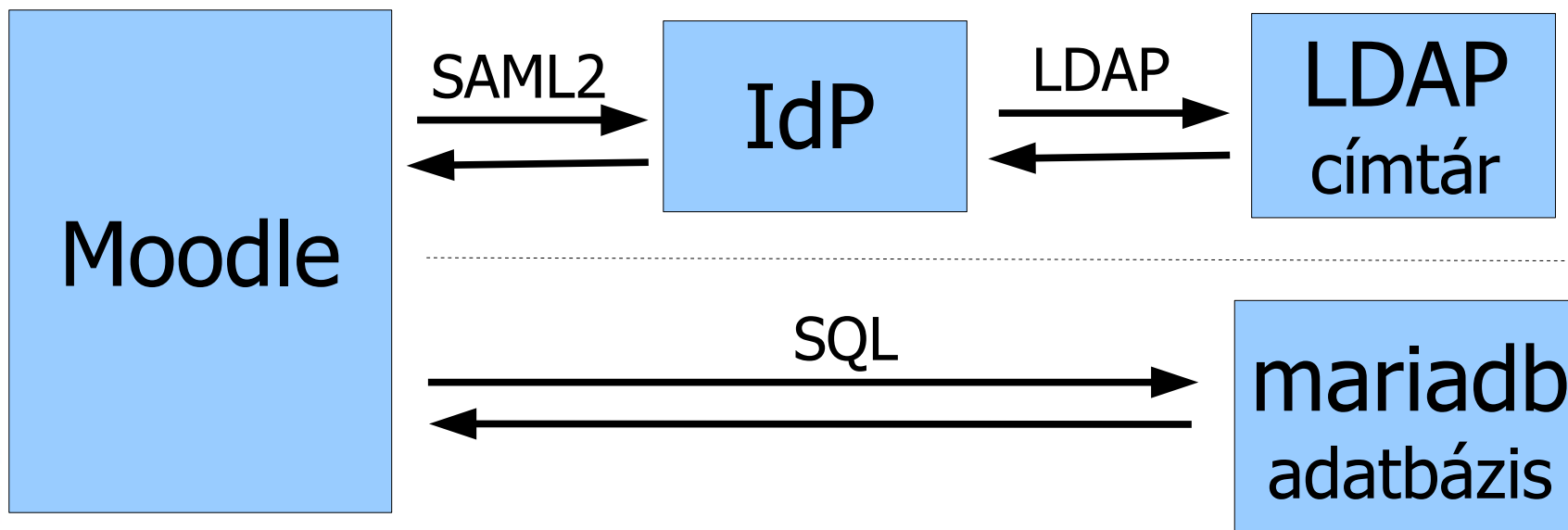


A felhasználóazonosítás és a jogosultságkezelés kiszervezése

MoodleMoot 2012

login név
személynév
email cím

felhasználó-
azonosítás



kurzusokhoz való
jogosultságok

jogosultság-
kezelés

Moodle-Shibboleth előnyök

- Nincs gond a login nevek és jelszók kiosztásával (több ezer hallgatónál ez már nem mindegy)
- A felhasználóknak kényelmes:
 - ha egy jelszóval több szolgáltatást érhetnek el (eduID, eduGAIN)
 - a single-sign-on
- Ha idegen szerverre kell átjelentkeztetnem a felhasználóimat valamelyik kurzusból az autentikáció megőrzésével, akkor egyszerűen fölveszem egy másik SP-nek. Az SSO miatt a felhasználónak nem kell bejelentkeznie, az adatai mégis rendelkezésre állnak az idegen szerveren is.
- Hivatkozhatom a Moodle-ból külső, védett tartalmakra (pl. cikk)

Moodle-Shibboleth hátrányok

- A kurzusok elérési jogosultságára nem terjed ki a megoldás, ezt külön kell megcsinálni (ez nem is biztos, hogy baj)
- A felhasználók csak az első bejelentkezéskor jönnek létre a moodle-ban, addig nem látjuk őket (bár ez LDAP-ból vélhetően megoldható)

A felmerült igények

- Ne kelljen várni arra, hogy a hallgató belép magától
- Képezzük le a Neptunban levő csoportbeosztást a Moodle-ra
- Munkavédelem:
 - a frissen bekerülő munkavállalók automatikusan kapjanak hozzáférést a Moodle-ban levő tanfolyamokhoz
 - legyen könnyen ellenőrizhető, hogy ki teljesítette a kötelezettségét

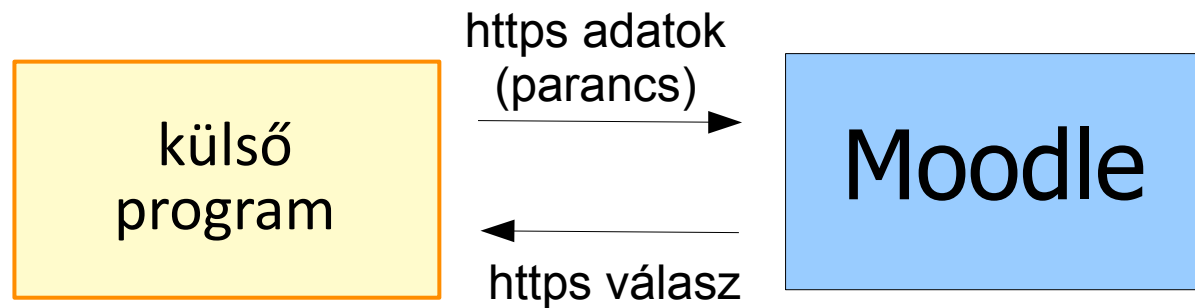
mindezt
automatizálva
szeretném

Mit feltételez mindez?

Azt, hogy egy külső programból, automatizált módon képes legyek ezekre:

- Felhasználók kezelése
 - Létezés ellenőrzése
 - Új felhasználó létrehozása
- A kurzus – felhasználó kapcsolatok kezelése
 - a csoportok és azok tagságának kezelése
 - kurzusjogosultság kezelése
 - kurzushasználat és eredmények lekérdezése

Számomra a megoldás: a Moodle „Web services”



Site administration | Plugins | Web services:

Web services

- Overview
- API Documentation
- External services
- Manage protocols
- Manage tokens

Overview ←
API Documentation
External services
Manage protocols
Manage tokens

Áttekintés

Step	Status	Description
1. Enable web services	Yes	Web services must be enabled in Advanced features.
2. Enable protocols	xmlrpc rest	At least one protocol should be enabled. For security reasons, only protocols that are to be used should be enabled.
3. Create a specific user		A web services user is required to represent the system controlling Moodle.
4. Check user capability		The user should have appropriate capabilities according to the protocols used, for example webservice/rest:use, webservice/soap:use. To achieve this, create a web services role with protocol capabilities allowed and assign it to the web services user as a system role.
5. Select a service		A service is a set of web service functions. You will allow the user to access to a new service. On the Add service page check 'Enable' and 'Authorised users' options. Select 'No required capability'.
6. Add functions		Select required functions for the newly created service.
7. Select a specific user		Add the web services user as an authorised user.
8. Create a token for a user		Create a token for the web services user.
9. Enable developer documentation	No	Detailed web services documentation is available for enabled protocols.
10. Test the service		Simulate external access to the service using the web service test client. Use an enabled protocol with token authentication. WARNING: The functions that you test WILL BE EXECUTED, so be careful what you choose to test!

Speciális felhasználó létrehozása

Step

1. Enable web services

2. Enable protocols

3. Create a specific user

4. Check user capability

5. Select a service




6. Add functions

7. Select a specific user

8. Create a token for a user

9. Enable developer documentation

10. Test the service

First name / Surname	Email address	City/town	Country	Last access	Edit
Groupsync Webservice	ws_groupsync@nomail.org	Budapest	Hungary	17 hours 53 mins	  
Add a new user					

Ezzel a felhasználóval senki nem fog a szokásos módon bejelentkezni.

Fel kell viszont ruházni azokkal a jogokkal, amire szükség lesz a küldött parancsok végrehajtásához. Ehhez legjobb létrehozni egy új szerepkört (role), amibe majd beletesszük a szükséges jogosultságokat. A speciális felhasználónknak rendszer szinten megadjuk ezt a szerepkört (system role).

De honnan tudjuk, hogy milyen jogokra lesz szüksége a speciális felhasználónak?

Ehhez ki kell válasszuk az API dokumentációból a használni kívánt eljárásokat.

Overview

API Documentation ←

External services

Manage protocols

Manage tokens

Az API dokumentáció

API Documentation

To create a client we advise you to read the [Moodle web service client documentation](#)

Print all

- [auth_email_get_signup_settings](#) ▶
- [auth_email_signup_user](#) ▶
- [core_auth_confirm_user](#) ▶
- [core_auth_is_age_digital_consent_verification_enabled](#) ▶
- [core_auth_is_minor](#) ▶
- [core_auth_request_password_reset](#) ▶
- [core_badges_get_user_badges](#) ▶
- [core_block_get_course_blocks](#) ▶
- [core_calendar_create_calendar_events](#) ▶
- [core_calendar_delete_calendar_events](#) ▶
- [core_calendar_get_action_events_by_course](#) ▶
- [core_calendar_get_action_events_by_courses](#) ▶
- [core_calendar_get_action_events_by_timesort](#) ▶
- [core_calendar_get_calendar_day_view](#) ▶
- [core_calendar_get_calendar_event_by_id](#) ▶
- [core_calendar_get_calendar_events](#) ▶
- [core_calendar_get_calendar_monthly_view](#) ▶
- [core_calendar_get_calendar_upcoming_view](#) ▶
- [core_calendar_submit_create_update_form](#) ▶
- [core_calendar_update_event_start_day](#) ▶
- [core_cohort_add_cohort_members](#) ▶
- [core_cohort_create_cohorts](#) ▶
- [core_cohort_delete_cohort_members](#) ▶
- [core_cohort_delete_cohorts](#) ▶
- [core_cohort_get_cohort_members](#) ▶
- [core_cohort_get_cohorts](#) ▶

Ez a meghívható eljárások (functions) listája. A nevek elég pontosan orientálnak a tekintetben, hogy melyik eljárás mire jó. Az adott eljárás nevére kattintva megkapjuk, hogy milyen paramétereket kell/lehet megadni, és milyen adatokat kapunk vissza.

A listából válasszuk ki, hogy pontosan melyik eljárásokra lesz szükségünk!

- Overview
- API Documentation
- External services ←
- Manage protocols
- Manage tokens

Az új webservice létrehozása

- Step
- 1. Enable web services
- 2. Enable protocols
- 3. Create a specific user
- 4. Check user capability
- 5. Select a service
- 6. Add functions
- 7. Select a specific user
- 8. Create a token for a user
- 9. Enable developer documentation
- 10. Test the service

External services

Information

A service is a set of functions. A service can be accessed by all users or just specified users.

Built-in services

External service	Plugin	Functions	Users	Edit
Attendance	mod_attendance	Functions	All users	Edit
Moodle Mobile additional features service	local_mobile	Functions	All users	Edit
Moodle mobile web service	moodle	Functions	All users	Edit

Custom services

External service	Delete	Functions	Users	Edit
groupsync	Delete	Functions	Authorised users	Edit

Add

- Overview
- API Documentation
- External services ←
- Manage protocols
- Manage tokens

Eljárások hozzáadása

Add functions to the service "groupsync"

- | Step |
|-----------------------------------|
| 1. Enable web services |
| 2. Enable protocols |
| 3. Create a specific user |
| 4. Check user capability |
| 5. Select a service |
| 6. Add functions |
| 7. Select a specific user |
| 8. Create a token for a user |
| 9. Enable developer documentation |
| 10. Test the service |

Function	Description	Required capabilities	Edit
core_course_create_categories	Create course categories	moodle/category:manage	Remove
core_course_create_courses	Create new courses	moodle/course:create, moodle/course:visibility	Remove
core_course_delete_categories	Delete course categories	moodle/category:manage	Remove
core_course_get_courses	Return course details	moodle/course:view, moodle/course:update, moodle/course:viewhiddencourses	Remove
core_enrol_get_enrolled_users	Get enrolled users by course id.	moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update, moodle/site:accessallgroups	Remove

Innen tudjuk, hogy a korábban létrehozott szerepkört, amit már hozzárendeltünk a speciális felhasználónkhoz, milyen jogosultságokkal kell feltöltenünk.

- Overview
- API Documentation
- External services ←
- Manage protocols
- Manage tokens

A felhasználó hozzáadása

- Step**
1. Enable web services
 2. Enable protocols
 3. Create a specific user
 4. Check user capability
 5. Select a service
 6. Add functions
 7. Select a specific user
 8. Create a token for a user
 9. Enable developer documentation
 10. Test the service

External services

Information

A service is a set of functions. A service can be accessed by all users or just specified users.

Built-in services

External service	Plugin	Functions	Users	Edit
Attendance	mod_attendance	Functions	All users	Edit
Moodle Mobile additional features service	local_mobile	Functions	All users	Edit
Moodle mobile web service	moodle	Functions	All users	Edit

Custom services

External service	Delete	Functions	Users	Edit
groupsync	Delete	Functions	Authorised users	Edit

Add

- Overview
- API Documentation
- External services
- Manage protocols
- Manage tokens ←

A token létrehozása

Create token

Token

Username / User id !

Service !

IP restriction

Valid until Enable

Save changes Cancel

There are required fields in this form marked ! .

- Step
- 1. Enable web services
- 2. Enable protocols
- 3. Create a specific user
- 4. Check user capability
- 5. Select a service
- 6. Add functions
- 7. Select a specific user
- 8. Create a token for a user
- 9. Enable developer documentation
- 10. Test the service

Token	First name / Surname	Service	IP restriction	Valid until
6d6223e57f0f5406ef67d03869cd8632	Groupsync Webservice	groupsync	10.10.10.10,20.20.20.20	31 December 2018, 12:00 AM

Egyszerű feladatokra írjunk scriptet! (egy példa perl nyelven)

Célszerű a használt funkciókra saját eljárásokat írni, amiket kényelmesen meghívhatunk.
Pl. egy felhasználó létezésének ellenőrzése / az adatai lekérése

```
Package Moodle_WS;

use LWP::UserAgent;
use JSON;

BEGIN { use vars qw($url_ws $token $ua);
        $url_ws = "https://my.moodle.site/moodle/webservice/rest/server.php"; # service URL
        $token   = "6d6223e57f0f5406ef67d03869cd8632";                       # token
        $ua      = LWP::UserAgent->new;                                       # a web user agent létrehozása
    };

sub get_user_data {
    my($login_name) = @_; # a login név az egyetlen argumentum
    my $params = {
        'wstoken'           => $token, # a tokennel azonosítjuk magunkat
        'wsfunction'        => "core_user_get_users", # ezt az eljárást akarjuk meghívni
        'moodlewsrestformat' => "json", # json formátumban várjuk a választ
        'criteria[0][key]'   => 'username', # a username alapján keressen
        'criteria[0][value]' => $login_name, # az átadott felhasználónév
    };
    my $result = $ua->post($url_ws, $params); # a web user agent elküldi a parancsot
    if (not $result->is_success) { # ha nem sikerült
        print $result->status_line, "\n"; # akkor írjuk ki, hogy miért
    }
    my $jsondecoder = JSON->new->allow_nonref; # létrehozunk egy JSON dekódert
    return $jsondecoder->decode($result->content); # és visszaadjuk a dekódolt eredményt
}

1;
```

Nézzük meg, hogy működik-e!

Ha ezt a programot futtatjuk le:

```
#!/usr/bin/perl -w

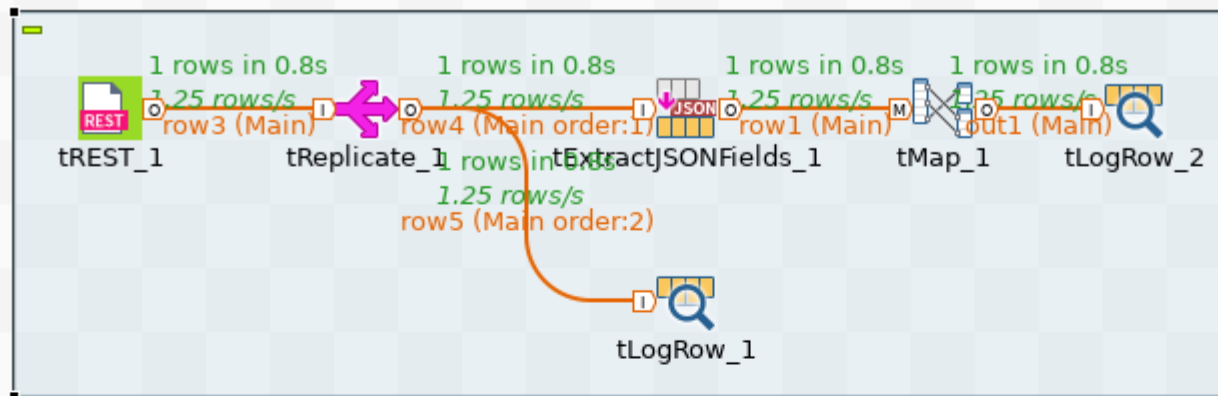
use Moodle_WS;
use Data::Dumper;

my $ret = Moodle_WS::get_user_data('torlasz');
print Dumper($ret);
```

Ezt az outputot kapjuk (rövidítve):

```
$VAR1 = {
    'warnings' => [],
    'users' => [
        {
            'department' => '',
            'descriptionformat' => 1,
            'mailformat' => 1,
            'lastaccess' => 1538554136,
            'email' => 'tornoci.laszlo@med.semmelweis-univ.hu',
            'auth' => 'shibboleth',
            'city' => 'Budapest',
            'id' => 123456,
            'theme' => '',
            'country' => 'HU',
            'firstaccess' => 1319009497,
            'lang' => '',
            'username' => 'torlasz',
            'fullname' => "Torn\x{f3}ci L\x{e1}szl\x{f3}"
        }
    ]
};
```


Egy hatékony fejlesztőeszköz: Talend Open Studio for Data Integration



Designer Code

Job(Moodle_WS_Rest_getdata 0.1) Contexts(Moodle_WS_Rest_getdata) Component Run (Job Moodle_WS_Rest_getdata)

Job Moodle_WS_Rest_getdata

Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run

Execution

Run Kill Clear

```
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|569|torlasz |Tornóci László|tornoci.laszlo@med.semmelweis-univ.hu|shibboleth|
2011-10-19 |2018-10-03 10:08:56|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
```

Grafikus fejlesztő környezetben dolgozunk, az adatok áramlását „rajzoljuk” le, majd a program ebből java nyelvű programot készít, amit le is futtat. A letölthető ingyenes verzió teljes értékűen használható.

Köszönöm a figyelmet!

