Front

Summary (one page)

Index

Introduction

Objectives (incluyendo descripción del problema, estudio de alternativas y

metodología empleada).

Technological description (puede incluir especificación, diseño e

implementación).

Conclusions (incluyendo los logros principales alcanzados y posibles

trabajos futuros).

Bibliography

Appendix

**Summary**

This document contains the description and development process of the Moodle Web Services Python Library. This library has been made to provide some external functions to facilitate the development of applications that interact with Moodle and to exploit this Moodle functionality.

Throughout this document we will see the Moodles requirements, its main characteristics, its web services plug-in, the REST protocol sentences to interact with it and the python code to do it and the diverse functionality provided by this python library.  
  
In first place, we will see our starting Moodle's development point, what are his characteristics and what we can do with them.  
After that we will analyze what kind of programs usually use Moodle like an external applications and haw they interact with it to work.  
Later, we will explain this python library solution to develop external applications which interact with Moodle.  
Finally we see the development process of this library, his API and some examples of its use.  
  
Moodle has become a popular teaching support system all around the world. This and the increase of mobile capacities to interact with many systems in the web have made appear new applications for Moodle.  
Because of this, Moodle developers decided to include some additional functionality to allow external applications work with Moodle since version 2.0.  
These functions are implemented like a Moodle plug-in, and its functionality initially was limited.

Later, the list of available functions was extended on version 2.5, so you will be able to do many things that Moodle can do but with this external service.

However, the use of this Moodle functionality requires some advanced know ledges about Moodle functionality and REST or other applications data exchange protocols.

This project tries to give some tools to configure Moodle easily to get this functionality with a python library to abstract at users of the Moodle communication process.

**Introduction**

Like we have said, one of the objectives of this project is to use the Moodle system with external applications.

To do this, the first step is to study how to use Moodle. Logically, to do that first we have to install this Moodle system, but Moodle is not this project objective, so this part will be summarized and referenced to some external official documentation, were this installation process is explained in detail.

Then, we have to configure Moodle so it can receive and response request from other applications. In this step we will see the different protocols that we can use to do this, but after that we will assume that the protocol used will be the REST protocol. On that time, we will see the Moodle’s permissions, administrated with system roles for users. We will see how affects this permissions to the answers given by Moodle too.

Once we have adequately configured Moodle, we will start to see how to do the communication process.  
First we will see the options to get information from Moodle, and we will see some application examples that use these options.  
One of these options is the web services plug-in, and it will be the service used to interact with this python library.

We will see 2 parts in this communication process. The first one is the authentication process, where you have a user with his password associated to a web service with some specific functions. The second one is the request itself, and the response will depend of the user capabilities.

After that, we will study the request and responses involved in this communication, what data needs the web service and what data returns. In this step we will do that the responses will be in JSON format, in order to facilitate the data extraction in python code.

Once this is done, we will analyze the web services API, and will see how to create some functions in python to do the library. These functions will have the same functionality than the functions provided by the Moodle’s web service.

Lastly we will see the library’s API, the functions implemented, their functionality, the parameters necessaries to use them and the data returned by these functions.

**What is Moodle**

Moodle is an open source code system used for teaching and learning purposes. It allows creating on line dynamic web pages, creating courses and managing it with the web interface. This made it popular between teachers around the world as a tool to provide resources to students and support teaching. It needs a data base system and a web server to work properly. Moodle's users are classified by roles. One user is able to have different roles at the same time, and different roles depending on the Moodle's part that he is. For example, one user can have the 'teacher' role in one course and the 'non editing teacher' role in another course. These roles give to users different capabilities on the system. Depending on these capabilities, users will be able to do some changes and consults to the system or not. By default, a user without role doesn’t have any capability on the system, so he can't do anything. The roles are what give the users the chance to do things in Moodle. There are some roles predefined in Moodle:

* Manger
* Course creator
* Teacher
* Non-editing teacher
* Student
* Guest
* Authenticated user
* Authenticated user on front page

These roles can be assigned to work on these contexts:

* System
* User
* Category
* Course
* Activity module
* Block

A category contains courses, and a course contains different type of activities separated in blocks.

<https://moodle.org/about/>

Moodle requirements:

* PHP
* Database system: MySQL
* Web server: apache

Moodle Web Services:

Moodle web services are additional Moodle modules that let you to use Moodle with other protocols than http such as AMF, SOAP, REST and XML-RPC. These services has been tough for interacting Moodle with other applications.

**Creating an external application using Moodle via Web Services**

* Moodle requirements:  
  REST protocol will be used in this application.

Using web services: <http://docs.moodle.org/24/en/Using_web_services>

To Allow users to create and see their own security keys(token) you must assign a system role with this user capabilities:

* + users [moodle/webservice:createtoken](http://docs.moodle.org/24/en/Capabilities/moodle/webservice:createtoken) (if not, the administrator must create the token manually for each user for the service)
  + [webservice/rest:use](http://docs.moodle.org/24/en/Capabilities/webservice/rest:use)
  + service required capability (depends of the service, check Settings-> Site administration-> Plugins-> Web services-> Manage services->Edit)
  + required capabilities for the service functions.
  + Also you must enable the web service function documentation, in Settings-> Site administration-> Plugins-> Web services-> Manage protocols
* Get web services token:  
  <http://docs.moodle.org/dev/Creating_a_web_service_client>

Service short name info: <https://moodle.org/mod/forum/discuss.php?d=197187>

* Get JSON answers to REST requests: <https://moodle.org/mod/forum/discuss.php?d=204469>

In the request you must include the parameter "moodlewsrestformat=json"  
example: /webservice/rest/server.php?wstoken=1cfc5fsd5a6fa75dfa&wsfunction=core\_webservice\_get\_site\_info&moodlewsrestformat=json