

- Home
- Documentation
- Downloads
- Demo
- Tracker
- Development
- Translation
- Moodle.net
- Search

You are here

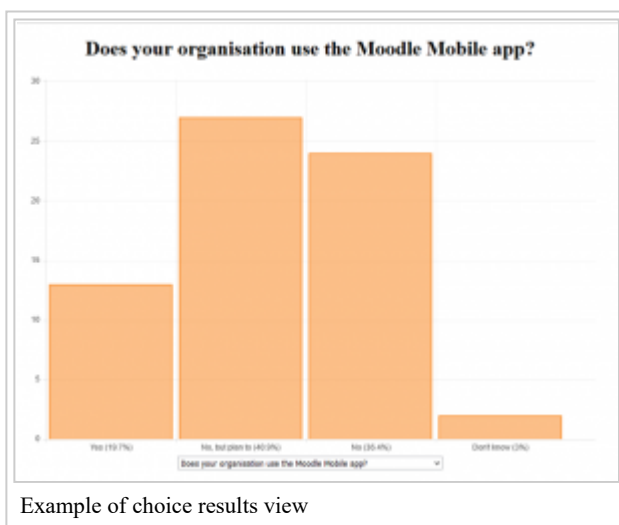
Choice Results

Main page ► Managing a Moodle site ► Moodle app ► **Choice Results**
Moodle app

- Moodle app features
- New for mobile
- Moodle app guide for admins
- Moodle app notifications
- Creating mobile-friendly courses
- Moodle App Block support
- Moodle Desktop
- Moodle app FAQ

Many teachers and lecturers use clickers and student response systems to add a level of real-time interaction during a lecture or presentation. Moodle has a inbuilt poll activity called choice which can be set up to allow students select one of a number of options. With so many students having mobile devices, this facilitates the teacher to have the students take the choice activity either using the mobile web interface of Moodle, or using the Moodle Mobile App.

This guide takes you through setting up a Moodle web service to view live choice results in a real time dynamic graph - so replacing the clicker requirement and using the students' own mobile devices instead.



Contents

- 1 Initial set up
- 2 Moodle site configurations
- 3 PHP script page for displaying results

4 Sample Code

Initial set up

You will need to configure the following components for this, as follows:

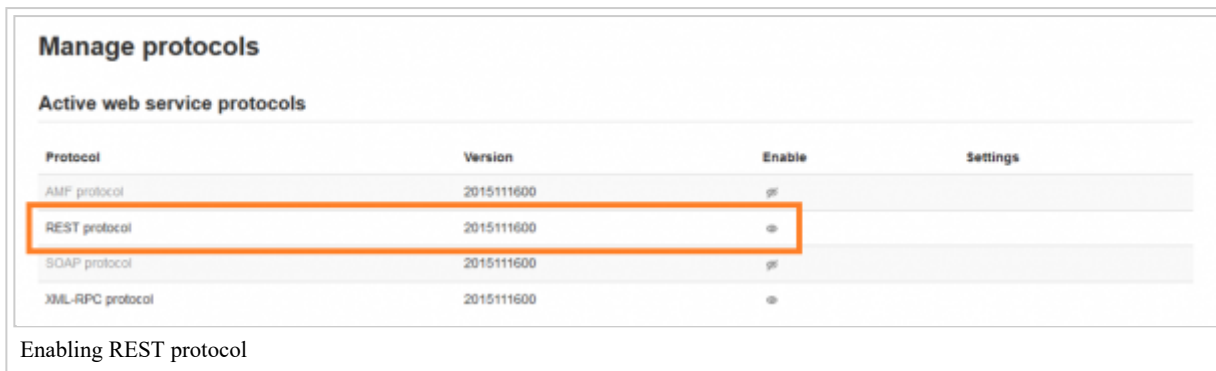
1. Configurations for your Moodle site
2. A separate PHP script page which calls your Moodle site via Moodle web services
3. A chartjs (or similar online graphing application) installation for displaying the choice results in visual graphs online

Moodle site configurations

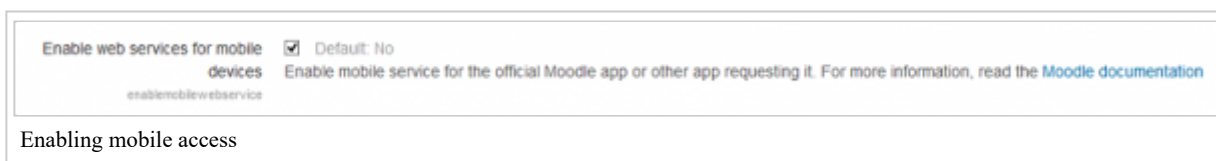
1) Ensure your site is ready for web services: This is at *Site administration -> Advanced features* where you need to enable and save the "Enable web services" setting



2) Ensure your site is ready for web services protocols This is at *Site administration -> Plugins -> Web services -> Manage protocols* where you need to enable and save either of the protocols you wish to use (for this guide, we will be using the REST protocol)



3) Ensure your site settings are ready for mobile access, thus allowing choice submissions from users via mobile devices This is at *Site administration -> Plugins -> Web services -> Mobile* where you need to enable and save "Enable web services for mobile devices"



4) Add a new user - call them "Choice View User" or something similar. See *Managing accounts* for further details on this operation.

5) Add a new role - call it "Choice WS Role" or something similar. This is at *Site administration -> Users -> Permissions -> Define roles*. On the first Adding a new role page, leave "Use role or archetype" set at "No role" and click on Continue For Context types where this role may be assigned - assign to System

Short name [?]

Custom full name [?]

Custom description [?]

Role archetype [?]

Context types where this role may be assigned

- System
- User
- Category
- Course
- Activity module
- Block

Add custom role

In capabilities - filter for the word service - click on Allow for Web service: REST protocol, or whichever protocol you're using

Filter

Capability	Permission [?]	Risks
System		
Create a web service token for mobile access <small>moodle/webservice:createmobiletoken</small>	<input type="checkbox"/> Allow	
Create a web service token <small>moodle/webservice:createtoken</small>	<input type="checkbox"/> Allow	
Web service: AMF protocol		
Use AMF protocol <small>webservice/amf.use</small>	<input type="checkbox"/> Allow	
Web service: REST protocol		
Use REST protocol <small>webservice/rest.use</small>	<input checked="" type="checkbox"/> Allow	
Web service: SOAP protocol		
Use SOAP protocol <small>webservice/soap.use</small>	<input type="checkbox"/> Allow	
Web service: XML-RPC protocol		
Use XML-RPC protocol <small>webservice/xmlrpc.use</small>	<input type="checkbox"/> Allow	

Add capability

Finally click on "Create this role" See Creating custom roles for further details on this operation.

6) Assign your new user to the new role. This is at *Site administration* -> *Users* -> *Permissions* -> *Assign system roles*. Click on your new role Search for your new user (you may need to filter by the user name) Select your user in the list and click on Add to assign the user to the role

Assign role 'Choice WS Role' in System ?

WARNING! Any roles you assign from this page will apply to the assigned users throughout the entire system, including the front page and all the courses. ✕

Existing users		Potential users
None	◀ Add Add Remove ▶	Potential users matching 'choice' (1) Choice View (choiceview@)
Search		Search
<input type="text"/>	<input type="button" value="Clear"/>	<input type="text" value="choice"/>
		<input type="button" value="Clear"/>
Search options ▶		

Assign role

When you reload the Assign system roles page, your new user should be listed in the Users with role column for the role.

7) Create the web service to be called for the choice results This is at Site administration -> Plugins -> Web services -> External services Add the web service, call it Choice View or something similar, click on Enabled, click on Authorised users only and finally click on Add Service.

External service

Name*

Short name

Enabled

Authorised users only ?

[Show more...](#)

Add web service

Next, add two functions to the web service, by searching for the Choice functions (you may need to filter by the word choice): - mod_choice_get_choice_results - mod_choice_get_choices_by_courses

Choice View

▼ Add functions

Name* Search choice

mod_choice_delete_choice_responses>Delete the given submitted responses in a choice
 mod_choice_get_choice_options>Retrieve options for a specific choice
 mod_choice_get_choice_results>Retrieve users results for a given choice.
 mod_choice_get_choices_by_courses>Returns a list of choice instances in a provided set of courses, if no courses
 mod_choice_submit_choice_response>Submit responses to a specific choice item.
 mod_choice_view_choice>Trigger the course module viewed event and update the module completion status.

Add functions Cancel

There are required fields in this form marked *

Add functions

8) Add your new user to the web service Authorised users list This is at *Site administration -> Plugins -> Web services -> External services* To add authorised users, click on the Authorised users link for your Choice View web service, this is similar to the interface where you assigned your user to the Choice View role. Search for your new user (you may need to filter by the user name) Select your user in the list and click on Add to assign the user to the Authorised users list

9) Create a web service token to authenticate the expected HTTP(S) requests This is at *Site administration -> Plugins -> Web services -> Manage tokens* Click on Add Search for the user you created for the Choice View In the Service dropdown, select the Choice View service you also just created Add an IP restriction if needed Add a time restriction if needed Click on Save changes and copy the resulting token

Create token

▼ Token

User* Search choice

Choice View

Service* Choice View

IP restriction

Valid until 5 April 2016 Enable

Save changes Cancel

Add token

10) Enrol your new user, "Choice View User", as a teacher in all courses from where you want to access the choice activities. This is needed so that the web service will be able to access the choice activity details using your user's token.

PHP script page for displaying results

This page will access your Moodle site choice activities and display their results with visual graphs. Copy our sample PHP script code to wherever you want it hosted and called from. It will need http or https access to your Moodle site where the choice activities are hosted. It will also need to have some level of security if it's going to be a publicly accessible URL as the PHP script itself contains details of a REST web service login with access to your Moodle site. We would suggest a htaccess login or similar. Edit and save the sample PHP script code to add your Moodle site URL and Choice View token values to where the code states "<yourURL>" and "<yourtoken>" respectively.

```
<?php
$url="<yourURL>";
$function="mod_choice_get_choice_results";
$function2="mod_choice_get_choices_by_courses";
$token="<yourtoken>";
```

Code configuration

The script code will also need access to an installation of chartjs, or other graphing application, in order to display the graphs, please see <http://www.chartjs.org/> for more details. You will then also need to edit the script code to add the server directory location of your chartjs installation to where the code states "<yourchartjslocation>".

To reveal the results, just type in the URL of your PHP script code into your browser.

The web service will now display all course accessible choice activities in the page dropdown, displaying the one which was first created initially. If you know the choice id from the database already, you can instead add the parameter to the URL as follows: "?choiceid=X".

One final point, the sample PHP script code is set with a simple reload of the page every 5 seconds so that the displayed graph updates with the latest results regularly. This reload could be more elegantly processed if needed, using AJAX for instance.

Sample Code

```
<?php
$url="<yourURL>";
$function="mod_choice_get_choice_results";
$function2="mod_choice_get_choices_by_courses";
$token="<yourtoken>";

// Grabs choice details including title
$ch2 = curl_init();
curl_setopt($ch2, CURLOPT_URL, "$url/webservice/rest/server.php?wsfunction=$function2&wstoken=$token&moodlewsrestfo");
curl_setopt($ch2, CURLOPT_RETURNTRANSFER, 1);
$result2 = json_decode(curl_exec($ch2));

if (isset($_REQUEST['choiceid'])) {
    $choiceid=intval($_REQUEST['choiceid']);
} else if (count($result2->choices) > 0) { // accessible choices found
    $choiceid=intval($result2->choices[0]->id);
} else {
    $choiceid = 0; // default value
}
if ($choiceid < 1) {
    echo("invalid id");
}
$colour="249,128,18"; // Moodle orange, change as required.

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "$url/webservice/rest/server.php?wsfunction=$function&wstoken=$token&choiceid=$choiceid");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$result = json_decode(curl_exec($ch));

$graph_title = "";
$dropdown = "<select name='choiceid' onchange='this.form.submit()'\>\n";
$dropdown .= "<option value='0'\>Select</option>\n";

foreach ($result2->choices as $choice) {
    $dropdown .= "<option value='{ $choice->id}'";
```

```

    if ($choice->id == $choiceid) {
        $graph_title = $choice->name;
        $dropdown .= " selected";
    }
    $dropdown .= ">{$choice->name}</option>\n";
}
$dropdown .= "</select>\n";

// data for bar graph
$chart = new stdClass();
$data = array();
$labels = array();
foreach ($result->options as $a) {
    $data[] = $a->numberofuser;
    $labels[] = $a->text. " (" .round($a->percentageamount,1)."%");
}
$chart->labels = $labels;
$dataset = new stdClass();
$dataset->data = $data;
$dataset->fillColor = "rgba({$colour},0.5)";
$dataset->strokeColor = "rgba({$colour},0.8)";
$dataset->highlightFill = "rgba({$colour},0.75)";
$dataset->highlightStroke = "rgba({$colour},1)";
$chart->datasets = array($dataset);

curl_close($ch);

?>
<!doctype html>
<html>
    <head>
        <meta http-equiv="refresh" content="5" >
        <title>Bar Chart</title>
        <script src="<yourchartjslocation>"></script>
    </head>
    <body>
        <center>
            <h1><?php echo($graph_title); ?></h1>
            <div id="canvas-holder">
                <canvas id="chart-area" width="900" height="600"/>
            </div>

            <form><?php echo($dropdown); ?><noscript><input type="submit" value="Submit"></noscript></form>

            <script>
                var pieData = <?php echo json_encode($chart); ?>;

                window.onload = function(){
                    var ctx = document.getElementById('chart-area').getContext('2d');
                    window.myPie = new Chart(ctx).Bar(pieData, {animationSteps : 1, responsive: false});
                };
            </script>
        </center>
    </body>
</html>

```

Retrieved from "https://docs.moodle.org/38/en/index.php?title=Choice_Results&oldid=122890"

Category: Mobile

- This page was last modified on 8 April 2016, at 17:04.
- Content is available under GNU General Public License unless otherwise noted.