

EJSApp External Interface

User's Guide

Luis de la Torre*, Rubén Heradio†

December 14, 2012

Abstract

This document summarizes the *EJSApp external interface*, which supports the inter-operation between EJS simulations managed into moodle by EJSApp and third-party moodle plugins (e.g., IPAL: <http://www.compadre.org/ipal/>).

Some test code for the interface has been included in the `test.php` file, inside the `external_interface` directory of EJSApp.

Contents

1	Importing the EJSApp external interface	2
2	Retrieving information from EJSApp	2
2.1	Getting EJSApp instances	2
2.2	Getting EJSApp state files	2
3	Drawing an EJS simulation	2

*ldelatorre@dia.uned.es

†rheradio@issi.uned.es

1 Importing the EJSApp external interface

The EJSApp external interface is implemented by the file `ejsapp_external_interface.php`. To import it from your moodle plugin just add the code in Figure 1.

```
1 require_once ($CFG->dirroot, '/mod/ejsapp/external_interface/ejsapp_external_interface.php');
```

Figure 1: Importing the EJSApp external interface

2 Retrieving information from EJSApp

This section summarizes how to extract information regarding EJSApp instances. Such functionality helps third-party plugins to provide users with configuration menus to specify what EJS simulations should be visualized, and to customize their state and appearance.

2.1 Getting EJSApp instances

To get the available EJSApp instances in a moodle site, use the function `get_ejsapp_instances`. It has one optional parameter to constraint the search for a given course and returns an array of objects with EJSApp instances information. For instance:

- Figure 2 gets all EJSApp instances in course with id 30. Figure 3 shows the correspondent output for an hypothetical course.
- Figure 4 gets all EJSApp instances for all courses where the user has editing permissions.

Remark: the resulting array produced by `get_ejsapp_instances` just includes EJSApp instances of courses where the user has role `editingteacher` or `manager`. That is, `get_ejsapp_instances` guarantees that a teacher cannot retrieve information about EJSApp instances that belong to courses where he doesn't have editing permissions.

```
1 $courseid = 30;  
2 $ejsapp_instances = get_ejsapp_instances($courseid);  
3 var_dump($ejsapp_instances);
```

Figure 2: Retrieving the ej sapp instances in the course with id 30

2.2 Getting EJSApp state files

Through the EJSApp form, the user may optionally specify an initial state file for an EJS simulation. In addition, the user can store additional state files by selecting the option `State input\output⇒Save state` on the contextual menu that emerges by clicking the mouse right button on an EJS applet. Such additional state files are stored in the user's private file area.

To get all the state files related to an EJSApp, use the function `get_ejsapp_states`. It receives the identifier of the EJSApp instance and returns an array of objects with information about:

1. The initial state file of the EJSApp instance (if it exists).
2. All the states files associated to the EJSApp instance that are stored in the user's private files area.

For instance, Figures 5 and 6 show an example of retrieving the current user's state files associated to the EJSApp instance with id 85.

3 Drawing an EJS simulation

Function `draw_ejsapp_instance` generates the HTML and javascript code to visualize an EJSApp simulation. That is, using the functions described in Section 2 third-party plugins retrieve information from EJSApp, process it and show it to their users as configuration menus. Then, `draw_ejsapp_instance` is used to print the EJS simulations according to the user preferences.

`draw_ejsapp_instance` has the following parameters:

```

1 array
2   0 =>
3     object(stdClass)[350]
4       public 'id' => string '83' (length=2)
5       public 'course' => string '30' (length=2)
6       public 'name' => string '3 tanks' (length=7)
7       public 'intro' => string '' (length=0)
8       public 'introformat' => string '1' (length=1)
9       public 'appwording' => string '' (length=0)
10      public 'appwordingformat' => string '1' (length=1)
11      public 'timecreated' => string '1354202230' (length=10)
12      public 'timemodified' => string '0' (length=1)
13      public 'applet_name' => string 'virtThreeTanksUNED' (length=18)
14      public 'class_file' => string 'Threetank.virtThreeTanksUNED_pkg.virtThreeTanksUNEDApplet.class' (length=63)
15      public 'codebase' => string '/mod/ejsapp/jarfiles/30/83/' (length=27)
16      public 'mainframe' => string 'MainFrame' (length=9)
17      public 'is_collaborative' => string '0' (length=1)
18      public 'applet_size_conf' => string '0' (length=1)
19      public 'preserve_aspect_ratio' => string '0' (length=1)
20      public 'custom_width' => null
21      public 'custom_height' => null
22      public 'is_rem_lab' => string '0' (length=1)
23      public 'height' => string '523' (length=3)
24      public 'width' => string '667' (length=3)
25
26   1 =>
27     object(stdClass)[347]
28       public 'id' => string '84' (length=2)
29       public 'course' => string '30' (length=2)
30       public 'name' => string 'rem servo' (length=9)
31       public 'intro' => string '' (length=0)
32       public 'introformat' => string '1' (length=1)
33       public 'appwording' => string '' (length=0)
34       public 'appwordingformat' => string '1' (length=1)
35       public 'timecreated' => string '1354202283' (length=10)
36       public 'timemodified' => string '0' (length=1)
37       public 'applet_name' => string 'remServoUNED' (length=12)
38       public 'class_file' => string 'Servo__UNED.remServoUNED_pkg.remServoUNEDApplet.class' (length=54)
39       public 'codebase' => string '/mod/ejsapp/jarfiles/30/84/' (length=27)
40       public 'mainframe' => string 'MainFrame' (length=9)
41       public 'is_collaborative' => string '0' (length=1)
42       public 'applet_size_conf' => string '0' (length=1)
43       public 'preserve_aspect_ratio' => string '0' (length=1)
44       public 'custom_width' => null
45       public 'custom_height' => null
46       public 'is_rem_lab' => string '1' (length=1)
47       public 'height' => string '558' (length=3)
48       public 'width' => string '750' (length=3)
49
50   2 =>
51     object(stdClass)[326]
52       public 'id' => string '85' (length=2)
53       public 'course' => string '30' (length=2)
54       public 'name' => string 'Gyroscope' (length=10)
55       public 'intro' => string '' (length=0)
56       public 'introformat' => string '1' (length=1)
57       public 'appwording' => string '' (length=0)
58       public 'appwordingformat' => string '1' (length=1)
59       public 'timecreated' => string '1354202419' (length=10)
60       public 'timemodified' => string '1355241443' (length=10)
61       public 'applet_name' => string 'ejs_Gyroscope' (length=13)
62       public 'class_file' => string 'gyroscope.Gyroscope_pkg.GyroscopeApplet.class' (length=45)
63       public 'codebase' => string '/mod/ejsapp/jarfiles/30/85/' (length=27)
64       public 'mainframe' => string 'mainFrame' (length=9)
65       public 'is_collaborative' => string '1' (length=1)
66       public 'applet_size_conf' => string '2' (length=1)
67       public 'preserve_aspect_ratio' => string '0' (length=1)
68       public 'custom_width' => string '1000' (length=4)
69       public 'custom_height' => string '600' (length=3)
70       public 'is_rem_lab' => string '0' (length=1)
71       public 'height' => string '540' (length=3)
72       public 'width' => string '554' (length=3)

```

Figure 3: Result of executing Figure 2

```

1 $ejsapp_instances = get_ejsapp_instances();

```

Figure 4: Retrieving all ejssapp instances for the whole moodle site

```

1 $ejsapp_id = 85;
2 $state_files = get_ejsapp_states(85);
3 var_dump($state_files);

```

Figure 5: Retrieving the current user's state files associated to the EJSApp instance with id 85

```

1 array
2   0 =>
3     object(stdClass)[301]
4       public 'state_name' => string 'Gyroscope_ruben.xml' (length=19)
5       public 'state_id' => string '77/mod_ejsapp/private/0/Gyroscope_ruben.xml' (length=43)
6   1 =>
7     object(stdClass)[300]
8       public 'state_name' => string 'Gyroscope_Variables.xml' (length=23)
9       public 'state_id' => string '327/mod_ejsapp/xmlfiles/85/Gyroscope_Variables.xml' (length=50)

```

Figure 6: Result of executing Figure 5

1. `$ejsapp_id`: identifier of the EJSApp instance which is going to be visualized.
2. `$state_file`: optional string parameter that identifies an state file (see the `state_id` field in Figure 6, lines 5 and 9). If it is specified, the EJSApp instance is printed in the state described by the `$state_file` parameter, elsewhere, it values `null` by default and the EJSApp instance is printed in the initial state file configured in the EJSApp form.
3. `$width` and `$height`: optional int parameters that set the the size of the printed EJS simulation. If they are not specified, the simulation will be printed as configured in the EJSApp form.

Remark: `draw_ejsapp_instance` doesn't require that the EJSApp instances are visible. In other words, it can print EJSApp instances hidden by the course administrator.

```

1 // Draw ejssapp with id=85 following its form configuration
2 $code_1 = draw_ejsapp_instance(85);
3 echo $code_1;
4
5 echo '<br/>'; // write an end of line
6
7 // Draw ejssapp with id=85 with state $state_files[0]->state_id, width=500 and height=300
8 $code_2 = draw_ejsapp_instance(85, $state_files[0]->state_id, 500, 300);
9 echo $code_2;

```

Figure 7: Drawing EJS simulations

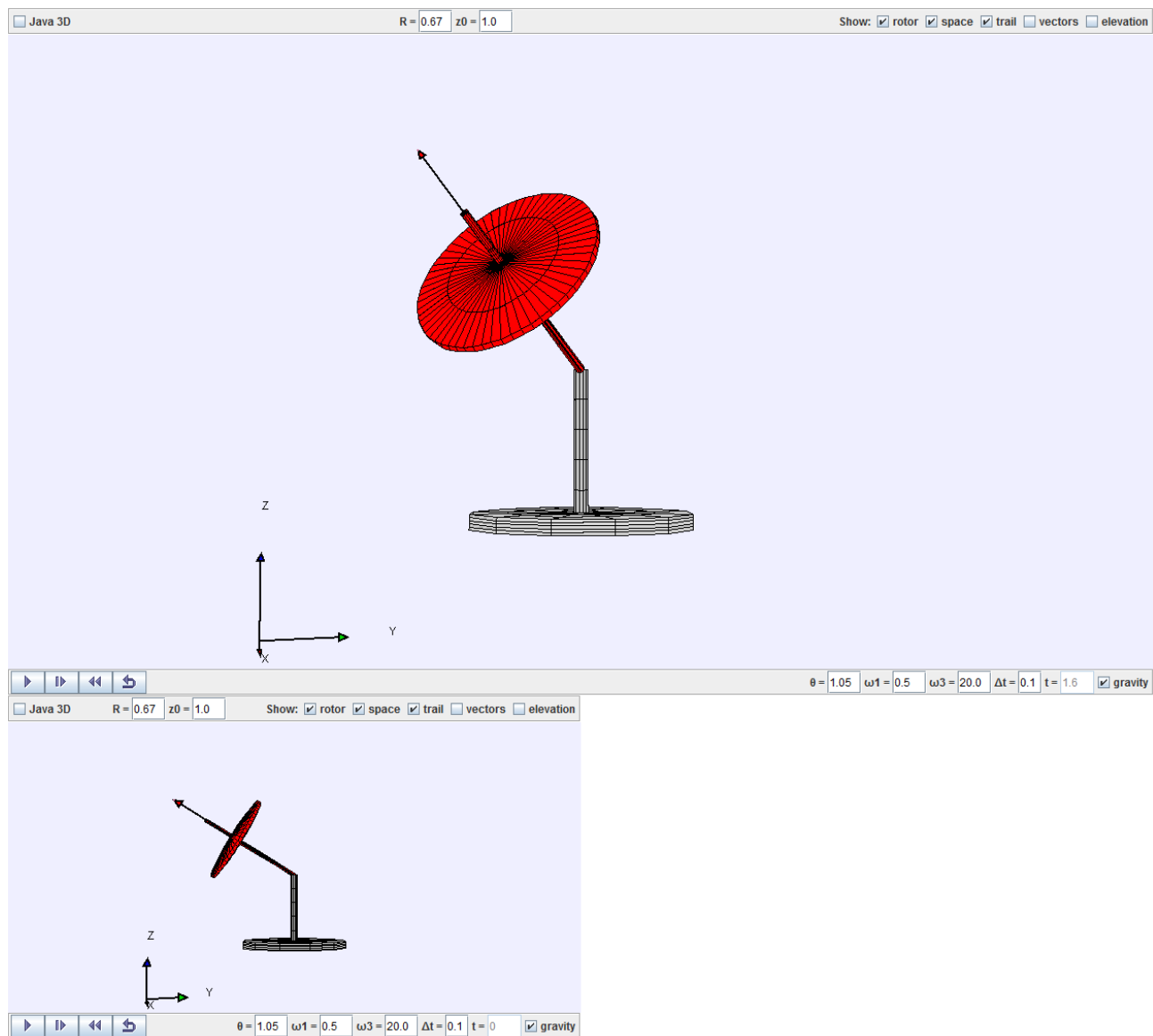


Figure 8: Result of executing Figure 7