JASMIN - AJAX Protocol

*Thomas Pronk*

*tpronk@uva.nl*

*V1, 2012-04-24*

# Introduction

This document describes how AJAX communication is resolved within JASMIN. AJAX requests are managed by *AjaxManager.js* at the client side, and *ajax\_handler.php* at the server side. These modules transmit JSON data structures between eachother. Below these data structures are described in more detail.

# General Operation

## How to use *AjaxManager.js*

All AJAX requests are initialized from the client; *AjaxManager.js*. How to use AjaxManager:

* The AjaxManager is constructed with an URL to ajax\_handler, a report function, time between retries, and maximum number of retries.
* Once constructed, you can send requests via the *send* function. This function takes three arguments:
  + The request, which should be an associative array containing: (1) “type”, which is a string describing the type of request, and (2) “content”, the content of the requests that varies according to type.
  + A function to call on success. The success function will receive two arguments: (1) requestId, and (2) data returned by *ajax\_handler.php* in response to this request.
  + A function to call on fail. The fail function will receive two arguments: (1) requestId, and (2) an error message.

If *send()* is called, *AjaxManager.js* adds the request to the queue, and makes AJAX requests to ajax\_handler.php. Once a request time out, it will keep trying until the maximum attempts is exceeded.

## What *AjaxManager.js* sends to *ajax\_handler.php*

The JSON structure sent to ajax\_handler.php is an indexed array of requests. Each request is an associative array, structured like this:

* “requestId”, (int) the ID of this request
* “attemptCounter”, (int) number of times *AjaxManager.js* has already tried to send it
* “content”, the content of the request, which in turn has the indexes:
  + “type”, string describing type of request
  + “content”, content of request (varying with type)

## What *ajax\_handler.php* sends to *AjaxManager.js*

The JSON structure returned to *AjaxManager.js* is an indexed array of responses to the requests. Each response is an associative array, structured like this:

* “requestId”, (int) the ID of the request responded to
* “content”, the content of the response (which varies with the type of the request)

# Request types

Below is a list of the types of requests that can be sent to ajax\_handler.php:

|  |  |
| --- | --- |
| **Type** | **Description** |
| store\_events | Store a set of events via the Logger. |
| translate | Translate terms |
| session\_state | Get/set a session\_state |
| now | Current server time as UNIX timestamp |
| get\_participation\_var | Get participation variables (2do) |
| set\_participation\_var | Set participation variable (2do) |

## store\_events

Store a set of events via the Logger. Content should contain two indexes: (1) “tasklog\_id” – (int) the tasklog\_id of the current task, (2) “events” – an indexed array of events to log. Each event is an associative array containing these indexes:

|  |  |
| --- | --- |
| **Type** | **Description** |
| time | (int) Time at which the event is generated as UNIX timestamp |
| source | (string) Module generating the event |
| type | (string) Type of event |
| name | (string) Name of event |
| params | (string) Additional information |

The *ajax\_handler* returns true if logging was successful.

To get the tasklog\_id, initialize the logger with the correct variables. By creating a view via the Template library, a javascript *config* variable is provided, which contains all relevant ids (optionally encrypted) and urls. The tasklog\_id is available as *config[ “tasklog\_id” ]*.

## translate

Get a set of translations via the Translator. Content should contain two indexes: (1) “type” – (string) type of translation requested to Translator, (2) “content” – What to translate (varies with type of translation requested). The table below describes the type of translations that can be requested.

|  |  |
| --- | --- |
| **Type** | **Description** |
| experiment | Get all translations that belong to an experiment. “content” should be (int) experiment\_id. |
|  |  |

The *ajax\_handler* returns the requested translations as an associative array, of which the indexes are the terms, and the values are the translations. Each translation is an associative array with two indexes: “value” – (string) the translation of this term in the current language, “status” – (int) the status of the translation (1 = requested, 2 = testing, 3 = done).

## session\_state

Get or set a session\_state. Content should contain two indexes: (1) “type” – (string) “get” or “set”. (2) “slot” – key of storage slot to set/get. (3) “content” – (string) If type==set, content is the new value for this storage slot.

## now

Returns current server time as UNIX timestamp.

# Need a demo?

See “controllers/experiment/test\_ajax.php” for a demo of the AjaxManager being used.