

**Author:** Rabah Meradi  
**Contact:** [rabah.meradi@logilab.fr](mailto:rabah.meradi@logilab.fr)  
**Organization:** Logilab S.A. (Toulouse)  
**Date:** 2015-05-12  
**Status:** In progress  
**Revision:** 2  
**Version:** 1

## **Abstract**

This document show how CubicWeb handle HTTP methods (GET, POST, PUT, DELETE, HEAD and Options) when using the pyramid framework.

It also shows where CubicWeb framework doesn't respect the REST architecture.

# **Table of Contents**

<b>1 Overview</b>	<b>1</b>
<b>2 Test procedure</b>	<b>1</b>
2.1 Dependencies	2
2.2 Test	2
<b>3 Results</b>	<b>3</b>
3.1 HTTP methods	3
3.2 Javascript calls	3
3.2.1 cubicweb.ajax.js	3
3.2.2 cubicweb.ajax.box.js	3
3.2.3 cubicweb.editions.js	3
3.2.4 cubicweb.calendar.js	4

## **1 Overview**

In this document we shows how CubicWeb handle HTTP basic methods (GET, POST, PUT, DELETE, HEAD, Options) with Pyramid. The test procedure is explained in the next section.

To verify if CubicWeb respect REST architecture, we verify that the CubicWeb JavaScript methods that call the API respect the below points:

1. Use GET method to get a resource
2. Use POST method to add a new resource
3. Use PUT method to update a resource
4. Use DELETE to delete a resource

## **2 Test procedure**

To see how CubicWeb handle HTTP methods we do the following:

1. Create a pyramid instance
2. Add the *blog* cube to the created instance
3. Start the instance

4. Create a blog entry to be used for test
5. Make requests to created entry url with the different HTTP methods and use different headers for each request.
6. See how CubicWeb handles these requests (response status, headers)

## 2.1 Dependencies

1. cubicweb
2. pyramid-cubicweb
3. cubicweb-pyramid
4. cubicweb-blog
5. Rester (used to make the requests)

## 2.2 Test

A script is used (test.sh) to help automate the test procedure. The script will setup a virtual environment and install all the needed dependencies.

The script will ask to setup the created instance. So you must do the following:

1. add the blog cube to the created instance
2. start the instance (use pyrami command)
3. create a blog entry and get its id

The script will ask to enter the id of of the created blog entry so it can run the requests.

After the tests are ran the logs of each request are printed. In the logs we can see the status and content-type headers.

To test that the HTTP method is not altered before arriving to the views we do the following:

1. create a cube which has the following schema:

```
class Test(EntityType):
    name = String(maxsize=50, required=True)
```

2. add the following view:

```
from cubicweb.predicates import is_instance
from cubicweb.web.views import primary

class BlogPrimaryView(primary.PrimaryView):
    __select__ = is_instance('Test')

    def entity_call(self, entity):
        self.w(u'<div class="test">')
        self.w(u'%s' % self._cw.http_method())
        self.w(u'</div>')
```

3. create a Test entity
4. Make requests using the different HTTP methods and see if the the printed method is the same as the one used to make the request.

## 3 Results

### 3.1 HTTP methods

CubicWeb handle all the HTTP methods with the same way. It return the same response for the HTTP methods: GET, POST, DELETE and PUT. The view receive the correct HTTP method for these methods. When doing HEAD or Options the code of the view is not called.

It always return *text/html* even if we specify that we only accept *application/json* or *application/xml* formats (or another format).

### 3.2 Javascript calls

#### 3.2.1 *cubicweb.ajax.js*

Function	HTTP Method	Python method	Comment
_	GET	_	This function doesn't change anything in the server. GET is ok.
unloadPageData	GET	unload_page_data (ajaxcontroller)	This function remove data from the user's session. Use Delete?
removeBookmark	Default (POST)	delete_bookmark (bookmark.py)	It delete a bookmark. Use DELETE method?
userCallBackThenUpdateUI	Default (POST)	render	The render method may change something in the server side. It depends on the rql query.
unregisterUserCallBack	Default (POST)	render	Same as the method above.
asyncRemoteExec	POST	any method	We can use this function to call any python method. This function must used only to call a method that change something in the server side.

#### 3.2.2 *cubicweb.ajax.box.js*

Function	HTTP method	Python method	Comment
ajaxValidateSelectorInput	Default (POST)	any method	Same as asyncRemoteExec method.
ajaxBoxRemoveLinkedEntity	Default (POST)	any method	Same as asyncRemoteExec

#### 3.2.3 *cubicweb.editions.js*

Function	HTTP method	Python method	Comment
addPendingInsert	GET	add_pending_insert	This method change the server state. We should do a POST.
addPendingDelete	Default (POST)	add_pending_delete	This method delete data in the server. Use DELETE method?

cancelPendingDelete	Default (POST)	add_pending_delete	Same as the method above
---------------------	----------------	--------------------	--------------------------

### **3.2.4 *cubicweb.calendar.js***

Function	HTTP method	Python method	Comment
dateSelected	GET	format_date	This method doesn't change the server state. GET is ok.