

## VelocityReport Plugin - Defect #483

**plugins.Velocity.createErrorResponse() returns HTTP status 200 regardless of what is passed.**

04/03/2012 03:05 PM - Ryan Parrish

<b>Status:</b>	Closed	<b>Start date:</b>	04/03/2012
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	velocity	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	v2.0		
<b>Browser (if web client):</b>			

### Description

It is my understanding that createErrorResponse should be setting the HTTP status with the code that is passed, however it is returning a HTTP 200 response for any code passed and instead putting the code in the body.

I would assume that the following code should return a HTTP 404, but instead returns a 200 with the 404 in the body.  
functionvr\_getContext(request){varresponseObject;responseObject=plugins.Velocity.createErrorResponse(404);return responseObject;}

### Response

```
HTTP/1.1 200 OK
Content-Length: 19
Server: Apache-Coyote/1.1
Cache-Control: max-age=0, must-revalidate, proxy-revalidate
Expires: 0
Content-Type: application/json;charset=UTF-8
Date: Tue, 03 Apr 2012 12:57:09 GMT
Connection: close
```

```
{"errorCode":"404"}
```

This creates an issue with RESTful sites that need to use HTTP status codes to convey the availability of a resource in the case of a 404.

### History

#### #1 - 04/03/2012 03:07 PM - Ryan Parrish

Wow...redmine totally ate my ticket, here is the summary...

that function I posted I would expect to return a HTTP 404, but instead returns a 200 with the status code in the body. HTTP response pasted below.

```
HTTP/1.1 200 OK
Content-Length: 19
Server: Apache-Coyote/1.1
Cache-Control: max-age=0, must-revalidate, proxy-revalidate
Expires: 0
Content-Type: application/json;charset=UTF-8
Date: Tue, 03 Apr 2012 12:57:09 GMT
Connection: close
```

```
{"errorCode":"404"}
```

#### #2 - 04/03/2012 05:39 PM - Patrick Talbot

- Status changed from New to In Progress

I'm investigating: that's weird because I totally set the status with such code:

```
if (responseObject != null && responseObject.getResponseCode() == ResponseType.ERROR) {
    response.setStatus(responseObject.getErrorCode());
}
```

I just tried it and this get executed, although the status stays 200!

**#3 - 04/03/2012 06:07 PM - Patrick Talbot**

- *Status changed from In Progress to Closed*

Found the problem: the status was set after some content was written to the output buffer, so the servlet container considered that the content was found.

Putting the same code before writing to the output stream set the status while allowing to return a response.

Will be fixed in next version