

Zend_Toolbar - Ben Scholzen

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_Toolbar Component Proposal

Proposed Component Name	Zend_Toolbar
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Toolbar
Proposers	Lead: Ben Scholzen Contributor: Christoph Dorn
Zend Liaison	TBD
Revision	1.0 - 11 June 2009: Initial Draft. (wiki revision: 7)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

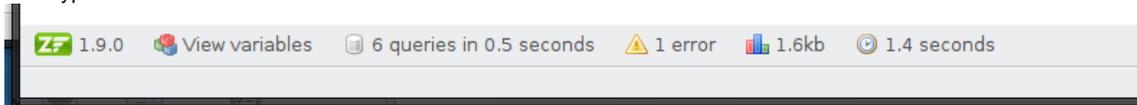
1. Overview

Zend_Toolbar is a component which provides the developer with enhanced debugging functionality. Inspired by the Scienta ZF Debug plugin, it provides you with useful information about the current request.

Instead of printing out the information as plain HTML, Zend_Toolbar exposes the debug information via headers, and displays them via a Firefox extension at the bottom of the browser. This gives you two major advantages; First, you can debug non-HTML responses (like JSON, generated images and so on). Second, you are able to debug requests in production environment without disturbing other users.

For allowing production debugging, the Zend_Toolbar Firefox extension allows to define hostnames and paths to debug, and let you provide a password, so unauthorized users will not get any debug information via the headers.

Prototype of the Toolbar in Firefox 3.0:



2. References

- Scienta ZF Debug

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** only support Firefox 3 and up (does anyone want to write an IE equivalent extension?).
- This component **will only** display the toolbar when the user is authorized and debug information are available
- This component **must** be backward compatible (e.g. a toolbar extension v2.0.0 should be able to display information from a 1.9.0)
- This component **will** allow authorization via hostname/path prefixes (password only)
- This component **will** allow to inspect variables and executed queries
- This component **will** allow to inspect information about the router and evaluated routes
- This component **will** display additional information like runtime, memory usage, filesize, etc.
- This component **will** use Zend_Wildfire for communication between server and extension

4. Dependencies on Other Framework Components

- Zend_Exception
- Zend_Controller
- Zend_Db
- Zend_Wildfire

5. Theory of Operation

The component is instantiated in the bootstrap by calling `Zend_Toolbar::init()`; with regarding options, like e.g. a required password. When a request comes in, it collects all required information and appends them to the response object as headers at the end.

6. Milestones / Tasks

- Milestone 1: [DONE] Prototype checked into SVN:
http://framework.zend.com/svn/framework/standard/branches/user/dasprid/Zend_Toolbar/
- Milestone 2: Working prototype checked into the incubator
- Milestone 3: Unit tests exist, work, and are checked into SVN.
- Milestone 4: Initial documentation exists.

7. Class Index

- Zend_Toolbar
- Zend_Toolbar_Plugin

8. Use Cases

UC-01

Initialising Zend_Toolbar

```
Zend_Toolbar::init();
```

UC-02

Initialising Zend_Toolbar with password protection

```
Zend_Toolbar::init(array('password' => 'foobar');
```

9. Class Skeletons

```
class Zend_Toolbar {  
    /**  
     * Initiate the toolbar  
     *  
     * @param array $options  
     * @return void  
     */  
    public static function init(array $options = null)  
    {  
    }  
}  
  
class Zend_Toolbar_Plugin {  
    public function dispatchLoopStartup(Zend_Controller_Request_Abstract $request)  
    {  
    }  
  
    public function dispatchLoopShutdown()  
    {  
    }  
}
```

```
]]></ac:plain-text-body></ac:macro>  
]]></ac:plain-text-body></ac:macro>
```