Zend_Http_UserAgent - Interakting

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[]]></ac:plain-text-body></ac:macro>
<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_Http_UserAgent (was Zend_Browser) Component Proposal

Proposed Component Name	Zend_Http_UserAgent (was Zend_Browser)
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Http_UserAgent (was Zend_Browser)
Proposers	Raphaël Carles
Zend Liaison	
Revision	1.0 - 03 August 2010: Initial Draft. (wiki revision: 13)

Table of Contents

1. Overview

This proposition covers several classes dedicated to the client browser/device detection and the available associated features and capabilities.
Its aim is to provide an interface to devices identification libraries like WURFL or DeviceAtlas and ease browsers differences handling, including mobile browsers.
This normalization of client environment detection can ease the management of multi-support development.

2. References

- WURFL
- Tera WURFL
- DeviceAtlas

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** allow quick detection of browsers, using per-session storage of last identification data
- This component **will** ease the use of external device identification libraries
- This component **will** be lightweight by using singleton pattern
- This component **will** provide an easy way to include new browsers types to detect.
- This component **will not** provide content adaptation/content replacement mechanisms or helpers

4. Dependencies on Other Framework Components

- Zend_Session (optional)

5. Theory of Operation

The UserAgent component should be seen as an information provider to Zend Framework applications at any level (helpers, controllers...)

Detection relies on declared or forced user agent information from server vars. It allows to have a standard behavior in the case that user agent string is present, a default one otherwise, and a forced mode where user agent is given at call time.

The identification class has a declared list of browsers types, ordered by priority.
Priorities can be changed to reflect application orientation (eg. a mobile-oriented website should have a faster identification of mobile devices than desktop ones).
New browser types can be developed and added to the priority list (or to extend an existing one), allowing wider recognition (probes, text browsers, ...) for the application that uses it.

The identification function is not called directly, although this is also possible.
All calls should be done to the getInstance method to execute the full identification process only one time per-request, or if session is activated, one time per-session and user agent.

After a quick detection of browser type, Zend_UserAgent *can* populate features by two ways :

- by responding directly to features checks (eg. return false to every request for a text browser, return true to every request for a desktop browser)
- by delegating to a Zend_UserAgent_Features_Adapter that will retrieve features.

The result is then stored as mentioned to bypass identification at next call (unless another user agent is forced).

6. Milestones / Tasks

Component already done for specific developments without ZF.

- Milestone 1: refactoring of existing code and adpatations to ZF design standards
- Milestone 2: Working prototype checked into the incubator supporting use cases #1, #2, ...
- Milestone 3: Working prototype checked into the incubator supporting use cases #3 and #4.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

7. Class Index

- Zend_UserAgent

- Zend_UserAgent_AbstractUserAgent
- Zend_UserAgent_Mobile
- Zend_UserAgent_Tablet
- Zend_UserAgent_Desktop (by default)
- Zend_UserAgent_Bot
- Zend_UserAgent_Text

- Zend_UserAgent_Features_Adapter
- Zend_UserAgent_Features_Adapter_WurflPhpApi (the first adapter to be provided)
- Zend_UserAgent_Features_Adapter_Wurfl
- Zend_UserAgent_Features_Adapter_TeraWurfl
- Zend_UserAgent_Features_Adapter_DeviceAtlas

- Zend_UserAgent_Storage
- Zend_UserAgent_Storage_NonPersistent
- Zend_UserAgent_Storage_Session

The first adapter provided will be Zend_UserAgent_Features_Adapter_WurflPhpApi http://wurfl.sourceforge.net/nphp/

8. Use Cases

9. Class Skeletons

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