Zend_Cache_Backend_Mongodb - Simon Harris

<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_Cache_Backend_Mongodb Component Proposal

Proposed Component Name	Zend_Cache_Backend_Mongodb
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Cache_Backend_Mongodb
Proposers	Simon Harris
Zend Liaison	TBD
Revision	1.0 - 18 August 2010: Initial Draft. (wiki revision: 12)

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

The MongoDB backend for Zend_Cache would allow developers to use MongoDB as a backend for content/data caching for the Zend_Cache component. MongoDB is a scalable, high-performance, open source, document-oriented "NoSQL" database.

2. References

- Zend_Cache
- MongoDB
- mongodb-user at Google groups

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** implement a fully funtional and configurable Zend_Cache backend using MongoDB

4. Dependencies on Other Framework Components

- Zend_Cache_Backend
- Zend_Cache_Backend_Interface
- Zend_Exception

5. Theory of Operation

The component is instantiated and used in a manner consistent with existing Zend_Cache backends

6. Milestones / Tasks

- Write proposal
- Gather feedback
- Review by the Zend team
- Develop full implementation and unit tests
- Documentation

7. Class Index

- Zend_Cache_Backend_Mongodb

8. Use Cases

UC-01: Simple Caching Example

```php
<?php

$frontendOptions = array(
    'lifetime' => 3600,
    'automatic_serialization' => true
);

$backendOptions = array(
    'connection_string'  => 'mongodb://localhost',
    'database'           => 'my_caching_db',
    'collection'         => 'my_caching_collection',
    'persistent_connect' => false
);

$cache = Zend_Cache::factory('Core',
                             'MongoDB',
                             $frontendOptions,
                             $backendOptions
);


$cacheKey = 'my_cache_key';

if (!$data = $cache->load($cacheKey)) {

    $data = perform_some_expensive_operation();
    $cache->save($data, $cacheKey);

}

do_something_with($data);
```

9. Class Skeletons

Class skeletons will follow the current Zend_Cache_Backend_Interface_Extended and to offer a consistent interface and identical functionality to other backends. Class skeletons will be published here as development progresses.

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

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