Zend_Service_WindowsAzure - Maarten Balliauw

<p><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_Service_WindowsAzure Component Proposal

Proposed Component Name	Zend_Service_WindowsAzure
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_WindowsAzure
Proposers	Maarten Balliauw
Zend Liaison	Unknown
Revision	1.0 - 14 March 2009: Initial Draft. 1.1 - 02 July 2009: Updated with more features. 1.2 - 06 August 2009: Updated with more features. (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

Zend_Service_WindowsAzure is a component that allows applications to make use of the Windows Azure API's. Windows Azure is a Microsoft platform which allows users to store unstructured data (think: files) and structured data (think: database) in a cloud service. More on http://www.microsoft.com/Azure.

The current proposal targets all 3 Windows Azure storage services. These services are:

- Blob Storage
- Table Storage
- Queue Service

An example implementation of this can be found on CodePlex: PHP SDK for Windows Azure and in the ZF SVN laboratory.

2. References

- Windows Azure
- Blob Storage API reference
- Table Storage API reference
- Queue Storage API reference

- Phyton version of the API

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** allow the use of Windows Azure blob storage within PHP applications.
- This component **will** allow the use of Windows Azure table storage within PHP applications.
- This component **will** provide a session handler which uses Windows Azure table storage.
- This component **will** allow the use of Windows Azure queue storage within PHP applications.
- This component **will** use the RESTful API that Windows Azure provides.
- This component **will** throw Exceptions upon failure.

4. Dependencies on Other Framework Components

This component has no dependencies on other Zend Framework components. It should be available for use in Zend Framework applications as well as standalone.

5. Theory of Operation

The component will connect to the REST interface that Windows Azure offers (see references) and will provide convenient access to the services offered. The necessary classes for interacting with the RESTful services will be built, allowing for a transparent manner to integrate Windows Azure in a Zend Framework application.

6. Milestones / Tasks

- Milestone 1: Windows Azure Blob Storage API implementation
- Milestone 2: Windows Azure Table Storage API implementation
- Milestone 3: Windows Azure Queue Storage API implementation
- Milestone 4: Windows Azure Table Storage session handler implementation

Each milestone takes integration tests and documentation into account.

7. Class Index

- Zend_Service_WindowsAzure
- Zend_Service_WindowsAzure_RetryPolicy
- Zend_Service_WindowsAzure_RetryPolicy_Exception
- Zend_Service_WindowsAzure_RetryPolicy_NoRetry
- Zend_Service_WindowsAzure_RetryPolicyRetryN
- Zend_Service_WindowsAzure_Storage
- Zend_Service_WindowsAzure_Storage_Batch
- Zend_Service_WindowsAzure_Storage_BatchStorage
- Zend_Service_WindowsAzure_Storage_Blob
- Zend_Service_WindowsAzure_Storage_BlobContainer
- Zend_Service_WindowsAzure_Storage_BlobInstance
- Zend_Service_WindowsAzure_Storage_DynamicTableEntity
- Zend_Service_WindowsAzure_Storage_Table
- Zend_Service_WindowsAzure_Storage_TableEntity
- Zend_Service_WindowsAzure_Storage_TableEntityQuery
- Zend_Service_WindowsAzure_Storage_TableInstance
- Zend_Service_WindowsAzure_Storage_Queue
- Zend_Service_WindowsAzure_Storage_QueueInstance
- Zend_Service_WindowsAzure_Storage_QueueMessage

- Zend_Service_WindowsAzure_Credentials
- Zend_Service_WindowsAzure_Exception
- Zend_Service_WindowsAzure_SessionHandler

8. Use Cases

UC-01

A script creates a new Zend_Service_WindowsAzure_Storage_Blob instance and uses it to upload, download and list blobs located on the Windows Azure Services Platform.

UC-02

A script creates a new Zend_Service_WindowsAzure_Storage_Table instance and uses it to store, retrieve, update and delete entities located on the Windows Azure Services Platform.

UC-03

A script creates a new Zend_Service_WindowsAzure_Storage_Table instance and uses it to store, retrieve, update and delete entities located on the Windows Azure Services Platform, encapsultaed in a Zend_Service_WindowsAzure_Storage_Batch instance which enables transaction support.

UC-04

A script creates a new Zend_Service_WindowsAzure_Storage_Queue instance and uses it to enqueue and dequeue messages that should be processed by the application. Queue messages are located on the Windows Azure Services Platform.

UC-05

A script hosted on Windows Azure Services Platform wants to make use of sessions in more than one Web Role instance. The Zend_Service_WindowsAzure_SessionHandler can be used to register a PHP session handler based on Zend_Service_WindowsAzure_Storage_Table, which provides regular session functions like session_start().

9. Class Skeletons

```
class Zend_Service_WindowsAzure {
}

class Zend_Service_WindowsAzure_RetryPolicy {
}

class Zend_Service_WindowsAzure_RetryPolicy_Exception {
}

class Zend_Service_WindowsAzure_RetryPolicy_NoRetry {
}

class Zend_Service_WindowsAzure_RetryPolicyRetryN {
}

class Zend_Service_WindowsAzure_Storage {
}

class Zend_Service_WindowsAzure_Storage_Batch {
}
```

```
class Zend_Service_WindowsAzure_Storage_BatchStorage {
}

class Zend_Service_WindowsAzure_Storage_Blob {
    public function createContainer();
    public function getContainerAcl();
    public function setContainerAcl();
    public function getContainer();
    public function getContainerMetadata();
    public function setContainerMetadata();
    public function deleteContainer();
    public function listContainers();
    public function putBlob();
    public function putLargeBlob();
    public function copyBlob();
    public function getBlob();
    public function getBlobInstance();
    public function getBlobMetadata();
    public function setBlobMetadata();
    public function deleteBlob();
    public function listBlobs();
}

class Zend_Service_WindowsAzure_Storage_BlobContainer {
}

class Zend_Service_WindowsAzure_Storage_BlobInstance {
}

class Zend_Service_WindowsAzure_Storage_DynamicTableEntity {
}

class Zend_Service_WindowsAzure_Storage_Table {
    public function listTables();
    public function createTable();
    public function deleteTable();
    public function insertEntity();
    public function deleteEntity();
    public function retrieveEntityById();
    public function retrieveEntities();
    public function updateEntity();
    public function mergeEntity();
}

class Zend_Service_WindowsAzure_Storage_TableEntity {
}

class Zend_Service_WindowsAzure_Storage_TableEntityQuery {
}

class Zend_Service_WindowsAzure_Storage_TableInstance {
}

class Zend_Service_WindowsAzure_Storage_Queue {
    public function createQueue();
    public function getQueue();
    public function getQueueMetadata();
    public function setQueueMetadata();
    public function deleteQueue();
```

```
    public function listQueues();
    public function putMessage();
    public function getMessages();
    public function peekMessages();
    public function clearMessages();
    public function deleteMessage();
}

class Zend_Service_WindowsAzure_Storage_QueueInstance {
}

class Zend_Service_WindowsAzure_Storage_QueueMessage {
}

class Zend_Service_WindowsAzure_Credentials {
}

class Zend_Service_WindowsAzure_Exception {
}

class Zend_Service_WindowsAzure_SessionHandler {
```

```
    public function register();
}
```

]]></ac:plain-text-body></ac:macro>
]]></ac:plain-text-body></ac:macro>PHP SDK for Windows Azure</p>