Zend_Service_Facebook - Bjoern Zapadlo

<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_Facebook Component Proposal

Proposed Component Name	Zend_Service_Facebook
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Facebook
Proposers	My Name
Zend Liaison	TBD
Revision	1.1 - 28 April 2011: Second 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_Service_Facebook provides an easy to API to Facebook's API and model-classes, without using facebook's own class.

2. References

- Facebook
- Facebook SDK

3. Component Requirements, Constraints, and Acceptance Criteria

*Component will be configurable (AppId, secret, ...)
*Component will be able to check if user is logged in
*Component will be able to redirect to log in window
*Component will be able to check if user is lo
*Component will be able to perform http calls to facebook's api
*Component will be able to provide model objects for all types of facebook objects

4. Dependencies on Other Framework Components

- Zend_Exception
- Zend_Http_Client
- Zend_JSON

5. Theory of Operation

The Component can be used to query facebook's graph api to get information about different type of facebook objects, like users, albums, ... It should act as a facade to the api. Stuff like $facebookUser = $facebook->getUser('userId') will provide an easy way to access the api. In addition return data will be presented in model objects with generic getters.

6. Milestones / Tasks

- Milestone 1: Write object factory class
- Milestone 2: Write object classes
- Milestone 3: Write facebook access classes
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Documentation exists.

7. Class Index

*Zend_Service_Facebook
*Zend_Service_Facebook_Exception
*Zend_Service_Facebook_Object_Factory
*Zend_Service_Facebook_Object_Interface
*Zend_Service_Facebook_Object_ObjectAbstract
*Zend_Service_Facebook_Object_Album
*Zend_Service_Facebook_Object_Application
*Zend_Service_Facebook_Object_Checkin
*Zend_Service_Facebook_Object_Comment
*Zend_Service_Facebook_Object_Event
*Zend_Service_Facebook_Object_FriendList
*Zend_Service_Facebook_Object_Group
*Zend_Service_Facebook_Object_Insights
*Zend_Service_Facebook_Object_Link
*Zend_Service_Facebook_Object_Message
*Zend_Service_Facebook_Object_Note
*Zend_Service_Facebook_Object_Page
*Zend_Service_Facebook_Object_Photo
*Zend_Service_Facebook_Object_Post
*Zend_Service_Facebook_Object_Review
*Zend_Service_Facebook_Object_Simple
*Zend_Service_Facebook_Object_StatusMessage
*Zend_Service_Facebook_Object_Subscription
*Zend_Service_Facebook_Object_Thread
*Zend_Service_Facebook_Object_User
*Zend_Service_Facebook_Object_Video

8. Use Cases

UC-01	**Facebook API can be queried**
UC-02	**Facebook login window is presented if user has to log in**
UC-03	**Data will be parsed an returned properly**

9. Class Skeletons

```php
<?php
class Zend_Service_Facebook_Object_Factory
{
    /* CONSTANTS */
    /**
     * types
     */
    const TYPE_ALBUM          = 'Album';
    const TYPE_APPLICATION    = 'Application';
    const TYPE_CHECKIN        = 'Checkin';
    const TYPE_COMMENT        = 'Comment';
    const TYPE_EVENT          = 'Event';
    const TYPE_FRIENDLIST     = 'FriendList';
    const TYPE_GROUP          = 'Group';
    const TYPE_INSIGHTS       = 'Insights';
    const TYPE_LINK           = 'Link';
    const TYPE_MESSAGE        = 'Message';
    const TYPE_NOTE           = 'Note';
    const TYPE_PAGE           = 'Page';
    const TYPE_PHOTO          = 'Photo';
    const TYPE_POST           = 'Post';
    const TYPE_REVIEW         = 'Review';
    const TYPE_STATUS_MESSAGE = 'StatusMessage';
    const TYPE_SUBSCRIPTION   = 'Subscription';
    const TYPE_THREAD         = 'Thread';
    const TYPE_USER           = 'User';
    const TYPE_VIDEO          = 'Video';


    /* FACTORY METHOD */
    /**
     * factory for factory objects
     *
     * @param string $type
     * @param array $data
     * @return Facebook_Object_ObjectAbstract
     * @throws Facebook_Exception
     */
    public static function factory($type, array $data)
    {
        // get constants of the class itself to check if type is correct
        $reflection = new Zend_Reflection_Class(__CLASS__);
        if (!in_array($type, $reflection->getConstants())) {
            throw new Facebook_Exception('Type of Object is incorrect');
        }

        $className = 'Facebook_Object_' . $type;

        if (!class_exists($className)) {
            throw new Facebook_Exception('Class "' . $className . '" could not be instantiated.');
        }

        return new $className($data);
    }
```

```
}

class Zend_Service_Facebook
{
    public function __construct(ZendConfig $config)
    {

    }
}

class Zend_Service_Facebook_Object_ObjectAbstract implements Facebook_Object_Interface
{
}
```

```
class Zend_Service_Facebook_Object_Album extends Facebook_Object_ObjectAbstract
{}
```

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