# Zend_Service_Amazon_DevPay - Wil Sinclair

<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_Amazon_DevPay Component Proposal

| | |
|---|---|
| **Proposed Component Name** | Zend_Service_Amazon_DevPay |
| **Developer Notes** | http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Amazon_DevPay |
| **Proposers** | Wil Sinclair |
| **Revision** | 0.1 - 18 November 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

DevPay is a new service from Amazon that provides both a UI-driven subscription and billing interface paired with a product activation and licensing web service. According to Amazon:

> *Amazon DevPay allows you to use Amazon's infrastructure to manage customer subscriptions and billing for your Amazon EC2 Machine Images (AMIs) or applications that use Amazon S3. To use Amazon DevPay, you simply:*
>
> - *Develop your software application using Amazon S3 or an Amazon EC2 Machine Image (AMI).*
> - *Use the simple Amazon DevPay web interface to register your application or AMI with Amazon DevPay, provide a product description and configure your desired pricing.*
> - *Embed the Amazon DevPay purchase pipeline link into your web site to allow your customers to purchase your product.*
> - *Integrate with Amazon DevPay's authentication mechanism to ensure that only authorized users access your application.*
> - *Monitor your business on the Amazon DevPay Activity page.*

This API has been designed to accommodate new APIs and support for additional AWS services in the future.

## 2. References

- Amazon DevPay
- License Service API
- Amazon EC2
- Amazon S3

## 3. Component Requirements, Constraints, and Acceptance Criteria

This component **must** support the full Amazon DevPay Licensing Service API.
This component **must** report all AWS errors to the developer.

## 4. Dependencies on Other Framework Components

- Zend_Exception
- Zend_Http
- Zend_Service_Amazon_Authentication

## 5. Theory of Operation

Zend_Service_Amazon_DevPay is instantiated directly using the credentials supplied by the developer. All operations can then be called from this object. All errors are wrapped in a Zend_Service_Amazon_Exception instance. All errors are passed through, except in cases where the client is advised to retry the request. If the retries fail, these errors are also wrapped in an instance of Zend_Service_Amazon_Exception and thrown to the user.

## 6. Milestones / Tasks

- Milestone 1: [DONE] Create Proposal
- Milestone 2: Working prototype checked into the laboratory
- Milestone 3: Collect feedback and get ZF team's recommendation
- Milestone 4: Finalize proposal
- Milestone 5: Commit code to trunk

## 7. Class Index

- Zend_Service_Amazon_DevPay

## 8. Use Cases

Usage is very simple. All parameters and return values are either strings or arrays.

**UC-01**

```
$devPayService = new Zend_Service_Amazon_DevPay(<my access key>, <my secret key>);
$devPayService->activateHostedProduct();
```

## 9. Class Skeletons

```php
<?php

class Zend_Service_Amazon_DevPay extends Zend_Service_Amazon_Abstract
 {

    /**
     * Create Amazon DevPay client.
     *
     * @param  string $access_key       Override the default Access Key
     * @param  string $secret_key       Override the default Secret Key
     * @param  string $region           Sets the AWS Region
     * @return void
     */
    public function __construct($accessKey=null, $secretKey=null, $authenticate = true) {}

 /**
     * Set DevPay endpoint to use
     *
     * @param string|Zend_Uri_Http $endpoint
     * @return Zend_Service_Amazon_DevPay
     */
    public function setEndpoint($endpoint) {}

    /**
     * Get DevPay endpoint
     *
     * @return Zend_Uri_Http
     */
    public function getEndpoint() {}


    /**
     * @param $activationKey
     * @param $productToken
     * @param $tokenExpiration
     * @return array Array containing 'AWSAccessKeyId', 'SecretAccessKey', and 'UserToken'
     */
 public function activateDesktopProduct($activationKey,
                                        $productToken,
                                        $tokenExpiration) {}
    /**
     * @param $activationKey
     * @param $awsAccessKeyId
     * @param $tokenExpiration
     * @param $productToken
     * @return array Array containing 'PersistentIdentifier' and 'UserToken'
     */
    public function activateHostedProduct($activationKey,
                                          $awsAccessKeyId,
                                          $tokenExpiration,
                                          $productToken) {}
    /**
     * @param $awsAccessKeyId
     * @param $userToken
     * @return array Array containing one or more product codes.
     */
    public function getActiveSubscriptionsByPid($awsAccessKeyId,
                                                $persistentIdentifier) {}

    /**
     * @param $additionalTokens
     * @param $awsAccessKeyId
     * @param $userToken
     * @return string The refreshed user token
     */
    public function refreshUserToken($additionalTokens,
                                     $awsAccessKeyId,
                                     $userToken) {}
```

```
    /**
     * @param $additionalTokens
     * @param $awsAccessKeyId
     * @param $userToken
     * @return string The refreshed user token
     */
    public function verifyProductSubscriptionByPid($awsAccessKeyId,
                                                   $persistentIdentifier,
                                                   $productCode) {}

    /**
     * @param $awsAccessKeyId
     * @param $productToken
     * @param $userToken
     * @return boolean True if the user is subscribed to the product, false otherwise
     */
    public function verifyProductSubscriptionByTokens($awsAccessKeyId,
```

```
                                                                $productToken,
                                                                $userToken) {}
  }
```

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