Zend_Service_Amazon_Ses

<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_Ses Component Proposal

Proposed Component Name	Zend_Service_Amazon_Ses
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Amazon_Ses
Proposers	Nicholas Miller Konr Ness
Zend Liaison	TBD
Revision	1.0 - 29 January 2011: Initial Draft. (wiki revision: 26)

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_Amazon_Ses would bring to Zend Framework the API calls available for Amazon's Simple Email Service.

2. References

- Amazon Simple Email Services Official Page
- Amazon Simple Email Services API Documentation
- Github Temporary Repository

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** implement all specified Amazon SES web-based actions (See references -> Amazon Simple Email Services API Documentation).
- This component **will** use Amazon's Version 3 Authentication
- This component **will not** save any external data.
- This component **will** use SimpleXMLElement to read Amazon's response messages.

- This component **will** allow subclassing
- This component **will** include an accompanying mail transport wrapper class.
- This component **will** extend Zend_Service_Amazon_Abstract
- This component **will** make http requests using Zend_Http_Client
- This component **will** validate any supplied http uri with the Zend_Uri_Http::valid() method

4. Dependencies on Other Framework Components

- Zend_Service_Amazon_Exception
- Zend_Service_Amazon_Abstract
- Zend_Http_Client
- Zend_Uri_Http

5. Theory of Operation

Zend_Service_Amazon_Ses would map the actions outlined Amazon's SES API documentation as class methods (See below use cases). Using the component is performed by instantiating a new Zend_Service_Amazon_Ses object. Being this component extends from Zend_Service_Amazon_Abstract AWS credentials are supplied either as static defaults, or at the time of instantiation through constructor arguments.

6. Milestones / Tasks

- Milestone 1: Write a working prototype "[DONE]"
- Milestone 2: Finalize Revisions based on community input
- Milestone 3: Port code to Zend Framework 2.0
- Milestone 4: Working prototype checked into the incubator when requirements are met.
- Milestone 5: Unit tests exist, work, and are checked into SVN.
- Milestone 6: Initial documentation exists.

7. Class Index

- Zend_Service_Amazon_Ses
- Zend_Service_Amazon_Ses_Exception
- Zend_Service_Amazon_Ses_Email
- Zend_Mail_Transport_Ses

8. Use Cases

UC-01 Statically Supplying AWS Credentials

```
Zend_Service_Amazon_Ses::setKeys('aws-key', 'aws-secret-key');
```

UC-02 Supplying AWS Credentials During Instantiation

```
$ses = new Zend_Service_Amazon_Ses('aws-key', 'aws-secret-key');
```

UC-03 Verify an Email Address

```
$ses = new Zend_Service_Amazon_Ses();
$ses->verifyEmailAddress('email-address@example.com');
```

UC-04 Deleting a Verified Email

```
$ses = new Zend_Service_Amazon_Ses();
$ses->deleteVerifiedEmailAddress('email-address@example.com');
```

UC-05 Sending an Email with the SendRawEmail action

```
$ses = new Zend_Service_Amazon_Ses();
$messageId = $ses->setRawEmail('...full email text with headers...');
```

UC-06 Sending an Email with the SendEmail Action

```
$email = new Zend_Service_Amazon_Ses_Email();
$email->addTo('recipient@example.com')
    ->setFrom('from-address@example.com', 'My Name')
    ->setSubject('Subject')
    ->setReturnPath('return-path-address@example.com')
    ->setBodyText('Text part of email')
    ->setBodyHtml('<b>Html part of email!</b>');

$ses = new Zend_Service_Amazon_Ses();
$messageId = $ses->sendEmail($email);
```

UC-07 Using Zend_Mail_Transport_AmazonSes

```
$ses = new Zend_Service_Amazon_Ses();

$mail = new Zend_Mail();
$mail->setFrom('a@gmail.com', 'From Name');
$mail->addTo($to, 'a@gmail.com');
$mail->setSubject($subject);
$mail->send(new Zend_Mail_Transport_AmazonSes($ses));
```

9. Class Skeletons

```
<?php
```

```
class Zend_Service_Amazon_Ses extends Zend_Service_Amazon_Abstract
{
    /**
     * Sets an alternative endpoint to the default
     *
     * @param  Zend_Uri_Http $endpoint
     * @return Zend_Service_Amazon_Ses
     * @throws InvalidArgumentException If the provided endpoint url is not valid
     */
    public function setEndpoint(Zend_Uri_Http $endpoint)
    {
    }

    /**
     * Gets the provided endpoint
     *
     * @return Zend_Uri_Http
     */
    public function getEndpoint()
    {
    }

    /**
     * Verifies an email address.
     * This action causes a confirmation email message to be sent to the
     * specified address.
     *
     * @param  RFC-822 Compliant Email Address $email
     * @return void
     */
    public function verifyEmailAddress($email)
    {
    }

    /**
     * Sends an email message, with header and content specified by the client.
     *
     * The SendRawEmail action is useful for sending multipart MIME emails.
     * The raw text of the message must comply with Internet email standards;
     * otherwise, the message cannot be sent.
     *
     * Raw email must:
     * + Message must contain a header and a body, separated by a blank line.
     * + All required header fields must be present.
     * + Each part of a multipart MIME message must be formatted properly.
     * + MIME content types must be among those supported by Amazon SES.
     *   Refer to the Amazon SES Developer Guide for more details.
     * + Content must be base64-encoded, if MIME requires it.
     *
     * @param  string $message    The raw text of the message.
     * @param  string $from       (Optional) From email address, if not included in the raq email's
headers.
     * @param  array  $recipients (Optional) Additional receipients to what's provided in the raw
email's headers.
     * @return string AWS Message Id
     * @throws RuntimeException   If the AWS repsonse is missing XML elements
     */
    public function sendRawEmail($message, $from = null, array $recipients = array())
    {
    }

    /**
     * Returns a list containing all of the email addresses that have been verified.
     *
     * @return array
     */
    public function listVerifiedEmailAddresses()
    {
    }
```

```
    /**
     * Composes an email message based on input data and then immediately queues the message for
sending.
     *
     * @param  Zend_Service_Amazon_Ses_Email $email
     * @return string AWS Message Id
     * @throws RuntimeException If the AWS request did not return the properly formatted XML response.
     */
    public function sendEmail(Zend_Service_Amazon_Ses_Email $email)
    {
    }

    /**
     * Returns the user's current activity limits.
     *
     * The following array keys are returned:
     *
     * max24HourSend: The maximum number of emails the user is
     *                allowed to send in a 24-hour interval.
     *
     * maxSendRate: The maximum number of emails the
     *              user is allowed to send per second.
     *
     * sentLast24Hours: The number of emails sent during
     *                  the previous 24 hours.
     *
     * @return array
     * @throws RuntimeException If the AWS request did not return the properly formatted XML response.
     */
    public function getSendQuota()
    {
    }

    /**
     * Deletes the specified email address from the list of verified addresses.
     *
     * @param string $email RFC-822 Compliant Email Address
     * @return void
     */
    public function deleteVerifiedEmailAddress($email)
    {
    }
}

class Zend_Service_Amazon_Ses_Email
{
    /**
     * Gets the HTML part of the message body
     * @return string
     */
    public function getBodyHtml()
    {
    }

    /**
     * Sets the HTML part of the message body
     * @param  string $bodyHtml
     * @param  string $charset (Optional)
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function setBodyHtml($bodyHtml, $charset = 'utf-8')
    {
    }

    /**
     * Gets the Return Path
     * @return string
     */
    public function getReturnPath()
    {
```

```
    }

    /**
     * Sets the Return Path
     * @param  string $returnPath Email Address
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function setReturnPath($returnPath)
    {
    }

    /**
     * Gets the Source (AWS Version of From Email Address)
     * @return string
     */
    public function getFrom()
    {
    }

    /**
     * Sets the From Address
     * @param  string $from RFC-822 Compliant Email Address
     * @param  string $name
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function setFrom($from, $name = null)
    {
    }

    /**
     * Returns all the TO recipients
     * @return array
     */
    public function getTo()
    {
    }

    /**
     * Add a TO address
     * @param  string $email
     * @param  string $name
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function addTo($email, $name = null)
    {
    }

    /**
     * Clears all TO addresses
     * @return void
     */
    public function clearTo()
    {
    }

    /**
     * Sets the reply-to email address(es) for the message.
     * If the recipient replies to the message, each reply-to address will
     * receive the reply.
     *
     * @param  string $email Email Address
     * @param  string $name (Optional)
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function addReplyTo($email, $name = null)
    {
    }

    /**
     * Gets the reply-to email address(es) for the message.
```

```
     *
     * @return array
     */
    public function getReplyTo()
    {
    }

    /**
     * Clears the reply to addresses.
     * @return void
     */
    public function clearReplyTo()
    {
    }

    /**
     * Gets registered CC addresses
     * @return array
     */
    public function getCc()
    {
    }

    /**
     * Clears CC Addresses
     * @return void
     */
    public function clearCc()
    {
    }

    /**
     * Adds the CC address
     * @param  string $email
     * @param  string $name
     * @return Zend_Service_Amazon_Ses
     */
    public function addCc($email, $name = null)
    {
    }

    /**
     * Gets the BCC email addresses
     * @return array
     */
    public function getBcc()
    {
    }

    /**
     * Adds a BCC address
     *
     * @param  string $email
     * @param  string $name
     * @return Zend_Service_Amazon_Ses
     */
    public function addBcc($email, $name = null)
    {
    }

    /**
     * Clears BCC Addresses
     * @return void
     */
    public function clearBcc()
    {
    }

    /**
     * Clears all recipients
```

```
     *
     * @return void
     */
    public function clearRecipients()
    {
    }

    /**
     * Gets the message subject
     *
     * @return string
     */
    public function getSubject()
    {
    }

    /**
     * Sets the message subject
     *
     * @param string $subject
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function setSubject($subject)
    {
    }

    /**
     * Gets the text part of the mail message
     *
     * @return string
     */
    public function getBodyText()
    {
    }

    /**
     * Sets the text part of the mail message
     *
     * @param  string $bodyText
     * @return Zend_Service_Amazon_Ses_Email
     */
    public function setBodyText($bodyText, $charset = 'utf-8')
    {
    }

    /**
     * Returns the parameters needed to make a SendEmail request to SES
     *
     * @return Zend_Service_Amazon_Ses_Response_SendEmail
     */
    public function getParams()
    {
    }
}

class Zend_Service_Amazon_Ses_Exception extends Zend_Service_Amazon_Exception
{
}


class Zend_Mail_Transport_AmazonSes extends Zend_Mail_Transport_Abstract
{
    /**
     * The message id for the last successfully sent message
     * @var string
     */
    protected $_lastMessageId;

    /**
     * Amazon Simple Email Service Object
```

```
     * @var Zend_Service_Amazon_Ses
     */
    protected $_ses;

    /**
     * Constructor
     * Recongized $options keys are:
     * + accessKey: Amazon AWS Access key
     * + secretKey: Amazon AWS Secret key
     * + endpoint: Amazon SES endpoint url
     *
     * @param  Zend_Config|array $options
     * @throws InvalidArgumentException If $options is neither an array or Zend_Config object
     */
    public function __construct($options = array())
    {
    }

    /**
     * Builds and stores a SES service object with the class
     *
     * @param  array $options
     * @return void
     */
    protected function _setSes(array $options)
    {
    }

    /**
     * Passes an email to the Amazon Simple Email Service Object
     *
     * @return void
     */
    protected function _sendMail()
    {
    }

    /**
     * Returns the last message id generated by Amazon's Simple Email Service
     * @return string
     */
    public function getLastMessageId()
    {
    }
}
```

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