Zend_Auth_Adapter_Cas - Jeremy Postlethwaite

<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_Auth_Adapter_Cas Component Proposal

Proposed Component Name	Zend_Auth_Adapter_Cas
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Auth_Adapter_Cas
Proposers	Jeremy Postlethwaite
Revision	1.0 - 11 May 2009: Ready for community review 1.1 - 30 Jul 2009: Moved to ready for recommendation 1.2 - 31 Jul 2009: Added standalone use case for inclusion in non-MVC environments 1.3 - 16 Dec 2009: CAS authentication adapter revised and checked into the incubator (wiki revision: 19)

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_Auth_Adapter_Cas is authentication adapter for CAS (Central Authentication Service) single-sign on protocol.

Central Authentication Service project, more commonly referred to as CAS. CAS is an authentication system originally created by Yale University to provide a trusted way for an application to authenticate a user.

2. References

- JA-SIG Central Authentication Service
- Central Authentication Service (at Wikipedia)

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** implement Zend_Auth_Adapter_Interface.
- This component **will** authenticate CAS ticket against CAS server.
- This component **will** return Zend_Auth_Result::SUCCESS when CAS ticket is valid.

- This component **will** return Zend_Auth_Result::FAILURE when CAS ticket is invalid.
- This component **must** support optional SSL / TLS encrypted transport.
- This component **will not** handle user redirects.
- This component **will not** save any data.

4. Dependencies on Other Framework Components

- Zend_Http_Client
- simplexml_load_string with support for namespaces:

5. Theory of Operation

CAS authentication is typically handled by a remote server.

6. Milestones / Tasks

This CAS authentication adapter is capable of working with CAS versions 1, 2 and 3

- Milestone 1: [DONE] Initial proposal published for review.
- Milestone 2: [DONE] Working prototype and some examples.
- Milestone 3: [DONE] Working prototype checked into the incubator supporting use cases.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

7. Class Index

- Zend_Auth_Adapter_Cas

8. Use Cases

UC-01 in MVC

Load parameters from Zend_Config:

cas.ini

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Authentication
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[cas]

params.hostname = "cas.example.org"
params.port = 443
params.path = "cas/"

; Optional parameters - default values shown
params.protocol = 'https'
params.serviceParam = 'service'
params.ticketParam = 'ticket'
params.validationParam = 'serviceValidate'
params.xmlNameSpace = 'http://www.yale.edu/tp/cas'

;
; Specify the Zend_Http_Client adapter
;
; default: Zend_Http_Client_Adapter_Socket
;
;cas.params.clientAdapter.adapter = "Zend_Http_Client_Adapter_Curl"
;
; Verify SSL peer
;
; FALSE to stop cURL from verifying the peer's certificate. Alternate
; certificates to verify against can be specified with the CURLOPT_CAINFO
; option or a certificate directory can be specified with the CURLOPT_CAPATH
; option. CURLOPT_SSL_VERIFYHOST may also need to be TRUE or FALSE if
; CURLOPT_SSL_VERIFYPEER is disabled (it defaults to 2).
;
; CURLOPT_SSL_VERIFYPEER = 64
;
;cas.params.clientAdapter.options.curloptions.64 = false
;
; Verify SSL host
;
; 1 to check the existence of a common name in the SSL peer certificate.
; 2 to check the existence of a common name and also verify that it matches the
; hostname provided.
;
; CURLOPT_SSL_VERIFYHOST  = 81
;
;cas.params.clientAdapter.options.curloptions.81 = false
;
; Add a custom certificate authority bundle for cURL
;
; @link http://curl.haxx.se/ca/cacert.pem
;
; CURLOPT_CAINFO = 10065
;
;cas.params.clientAdapter.options.curloptions.10065 =
"/usr/local/zend/apache2/conf/ssl.crt/ca-bundle.crt"
```

or load directly:

```
$config = array(
    'hostname'     => 'cas.example.org',
    'port'         => 443,
    'path'         => 'cas/',
);
```

Authenticate in an action

```
/**
 * casAction
 *
 * CAS Login Action
*/
public function casAction()
{

    $auth = Zend_Auth::getInstance();

    $config = array(
        'hostname'  => 'cas.example.org',
        'port'      => 443,
        'path'      => 'cas/',
    );

    $adapter = new Zend_Auth_Adapter_Cas($config);

    // If no identity is set and a ticket exists, attempt to authenticate
    if(!$auth->hasIdentity() && $adapter->hasTicket()) {

        $result = $auth->authenticate($adapter);

        if(!$result->isValid()) {
            $this->view->messages = $result->getMessages();
            return;
        }
    }

    // Logout if requested
    if(isset($_GET['logout'])) {

        $auth->clearIdentity();

        // Specify the landing URL to hit after logout
        $landingUrl = 'http://' . $_SERVER['HTTP_HOST'];
        $this->_redirect($adapter->getLogoutUrl($landingUrl));
    }

    // Send to CAS for authentication
    if(!$auth->hasIdentity()) {
        $this->_redirect($adapter->getLoginUrl());
    }
}
```

UC-02 in non-MVC

Standalone instance for non-MVC environments

```
<?php
/**
 * Base path
```

```
 */
define('BASE_PATH', realpath(dirname('../../')));

/**
 * Application path
 */
define('APPLICATION_PATH', BASE_PATH . '/application');

/**
 * Library path
 */
define('LIBRARY_PATH', BASE_PATH . '/library');

// Set the include path
set_include_path(LIBRARY_PATH . PATH_SEPARATOR . get_include_path());


$config = array(
    'hostname'  => 'cas.example.org',
    'port'      => 443,
    'path'      => 'cas/',
);

/**
 * cas
 *
 * CAS Login
 */
function cas($config)
{

    /**
     * @see Zend_Auth
     */
    require_once 'Zend/Auth.php';

    $auth = Zend_Auth::getInstance();

    /**
     * @see Zend_Auth_Adapter_Cas
     */
    require_once 'Zend/Auth/Adapter/Cas.php';

    $adapter = new Zend_Auth_Adapter_Cas($config);

    $messages = array();

    // If no identity is set and a ticket exists, attempt to authenticate
    if(!$auth->hasIdentity() && $adapter->hasTicket()) {

        $result = $auth->authenticate($adapter);

        if(!$result->isValid()) {

            $messages = $result->getMessages();
            return $messages;
        }
    }
```

```
    // Logout if requested
    if(isset($_GET['logout'])) {

        $auth->clearIdentity();

        // Expire the cookie
        if(isset($_COOKIE[session_name()])) {
            setcookie(session_name(), '', time()-42000, '/');
        }

        // Destroy the session.
        session_destroy();

        // Specify the landing URL to hit after logout
        $landingUrl = 'http://' . $_SERVER['HTTP_HOST'];
        return header('Location: ' . $adapter->getLogoutUrl($landingUrl));
    }

    // Send to CAS for authentication
    if(!$auth->hasIdentity()) {
        return header('Location: ' . $adapter->getLoginUrl());
    }

    return $messages;
}
?>
<html>
    <title>CAS Authentication Example</title>
    <body>

    <h1>CAS Authentication Example</h1>

    <p>
    <a href="/">Home</a>
    </p>

    <p>
    <a href="?logout">Logout</a>
    </p>

    <?php

    $messages = cas($config);

    /**
     * @see Zend_Debug
     */
    require_once 'Zend/Debug.php';

    // Variable dumps

    Zend_Debug::dump($messages, eval('return __FILE__;') . ' $messages', true);

    Zend_Debug::dump($_COOKIE, eval('return __FILE__;') . ' $_COOKIE', true);

    Zend_Debug::dump($_SESSION, eval('return __FILE__;') . ' $_SESSION', true);

    Zend_Debug::dump($_SERVER, eval('return __FILE__;') . ' $_SERVER', true);
```

?>

```
    </body>
</html>
```

9. Class Skeletons

http://framework.zend.com/svn/framework/standard/incubator/library/Zend/Auth/Adapter/Cas.php

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