Zend_Asterisk

<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_Asterisk Component Proposal

Proposed Component Name	Zend_Asterisk
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Asterisk
Proposers	Emanuele Deserti
Revision	1.2 - 21 February 2008: Second draft 1.1 - 10 October 2007: First draft (wiki revision: 6)

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_Asterisk is a component that contain two main components: Zend_Asterisk_Manager and Zend_Asterisk_Agi
Zend_Asterisk_Manager will connect with the Asterisk Manager interfaces and can send and load data with the local Asterisk server.
Zend_Asterisk_Agi implements a Asterisk Gateway Interface which can be used to control and interact with a call in progress.

2. References

- Download current release
- Asterisk (Digium)
- voip-info.org

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** connect with the Asterisk Manager and can write and read to Asterisk socket.
- This component **will** answer to Asterisk call and hangup an active call.
- This component **will** detect DTMF digits (single or collect digits) and play specific audio files.
- This component **will** read and write Asterisk global variables (like extensions) and call variables (like callerid).

4. Dependencies on Other Framework Components

- Zend_Exception

5. Theory of Operation

The component can be used in two way:

- As Asterisk Manager, who can send command directly to the server and read it's output
- As Asterisk Gateway Interfaces, who can answer to a call, process user input, interact with others services (like a database) and return results or informations to user.

6. Milestones / Tasks

- Milestone 1: design notes will be published here
- Milestone 2: Asterisk Gateway Interface: write to socket and read from socket.
- Milestone 3: Asterisk Gateway Interface: parse callerid, collect digits, play audio files.
- Milestone 4: Asterisk Manager: connect to Asterisk Server, send commands and read outputs.

7. Class Index

- Zend_Asterisk_Exception
- Zend_Asterisk (factory class)
- Zend_Asterisk_Manager
- Zend_Asterisk_Agi

8. Use Cases

MANAGER:send command and read output

```
Zend_Loader::loadClass("Zend_Asterisk_Manager");

$configs = array('server' => 'localhost',
   'post'     => '5038',
   'username' => 'manageruser',
   'secret'   => 'managerpass'
   );

$asm = new Zend_Asterisk_Manager($configs);
if ($asm->connect()) {
    $result = $asm->listCommands();
    Zend_Debug::dump($result);

    $result = $asm->command("sip show peers");
    Zend_Debug::dump($result);

    $result = $asm->command("core show version");
    Zend_Debug::dump($result);

    $result->disconnect();
}
```

MANAGER:originate dial

```
// If the user click on "Dial" link (route is like /module/controller/action/dial/301)
if ($this->_hasParam('dial')) {
    $source = '201'; // The source peer/number, I take this from the Auth session
    $destination = $this->_getParam('dial') // The destination peer/number, I take
this from a GET param

    // Connect to Asterisk Manager
    if ($asm->connect()) {
     // Originate: Channel, Extensions, Context, Priority and Caller ID
        $asm->Originate('SIP/' . $source, $destination, 'internal', '1', null, null,
null, '"Caller ID" <'.$source.'>');
    }
    sleep(1);
}
```

9. Class Skeletons

```
class Zend_Magic_Exception extends Zend_Exception {}

class Zend_Asterisk_Manager {}

class Zend_Asterisk_Agi
{

 function answer() {}
 function hangup() {}
 function getDigits() {}
}
```

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