SMS Class Info

File name
app/vendors/sms.php

Description:
CakePHP Vendor class sending one or many sms by means of:

1. gammu smsd class OR
2. OfficeRoute

Properties

:type (string {mysql | email})
:error (string)
:RelativeValidity (int {default value: 255})

Methods

Public

Comment (construct)
$type and $auth needs to be configured differently depending on your gateway type (Gammu or OfficeRoute).

_construct((string) $type, (array)$auth)

$type = string ('mysql') or
string ('email')

$auth = array(host, user, password, database) or
array(domain)

Comment (sendSMS)
Gammu: $id refers to a database entry corresponding to the SMS being sent. This is can be used to query Gammu for status or to delete entry.
OfficeRoute: $id is a boolean value. True represents an SMS successfully sent. False means that the email was not successfully sent.

(array) $id = sendSMS((string), $message, (int | array) $receiver, (int) $sender)

Comment (getStatus)
This method is only applicable for Gammu.

(array) $result = getStatus((int)$id)

Comment (delOutbox)
This method is only applicable for Gammu.

(int) $no_affected_rows = delOutbox((int | array) $id = null)
Comment (getPhones)
This method is only applicable for Gammu.

(array) $result = getPhones()

Comment (close)
This method is only applicable for Gammu.

(void) = close();

Private

(int) $id = send($msg, $dest, $sender)

(int) $id = inject($sender)

(bool) $status = dbAuth($auth);

(bool) $status = orAuth($auth);

Usage

1) In controller, import sms vendor:

```
App::import('Vendor', 'sms');
```

2) Define authentication credentials:

Case 1: mysql (Gammu)

```
$auth = array('database' => 'gammu',
              'user'     => 'root',
              'host'     => 'localhost',
              'password' => 'changeme'
              );
```

Case 2: email (OfficeRoute)

```
$auth = array('domain' => '<email domain name of office route>');
```

3) Define SMS parameters:

```
$message = "Hello Fred!";
$sender  = "555123456";
```

Case 1: Single receive

```
$receiver = "555654654";
```

Case 2: Multiple receivers

```
$receiver = array("555654654", "555654655");
```

4) Create object

```
$sms = new sms('mysql', $auth);  //gammu

$sms = new sms('email', $auth);  //OfficeRoute
```

5) Send SMS

```
$id = $sms->sendSMS($message,$receiver,$sender);
```

6) Get status of SMS (only in the case of Gammu)

```
$status = $sms->getStatus(14);
$status = $sms->getStatus(array(14,15,16));
```

7) Retrieve information from connected phones (only in the case of Gammu)

```
$data = $sms->getPhones();
```

8) Delete statistics over sent items (only in the case of Gammu)

```
$rows = $sms->delOutbox();              //Delete all
$rows = $sms->delOutbox(1);             //Delete entry with id = 1
$rows = $sms->delOutbox(array(1,2));    //Delete entries with id 1 and 2
```

9) Close connection (only in the case of Gammu)

```
$sms->close();
```

Note

To allow users to access the sms class from the security edition, the ACOS/AROS tables needs to be updated to allow such access.

For testing purpose, this restriction can be overwritten, by allowing this class to all unauthorized uses.
To do so, add "sms" to the list of globally allowed actions in app_controller.php

```
//Allow global access to pages and refresh methods
$this->Auth->allowedActions = array('display','refresh','sms');
```