



SubitoSMS JSON

JSON APIs

Date: sab 30 agosto 2008

Author: Julien Buratto

Version: 1.2

Contents

Introduction.....	4
Available actions.....	4
Using JSON.....	4
The Auth array.....	4
Checking credit.....	4
Example: request.....	4
Example: Response.....	5
XML notation.....	5
Request.....	5
Response.....	5
Check status.....	5
Example : request.....	6
Example: Response.....	6
XML notation.....	6
Request.....	6
Response.....	7
Status values.....	7
Send SMS messages.....	8
Example: request.....	8
Example: response.....	8
XML notation.....	8
Request.....	9
Response.....	9
Error messages.....	9

XML Notation.....	9
Error codes.....	9
PHP Examples.....	10
Requirements.....	10
The request wrapper.....	10
PHP > 4.3.0.....	10
PHP > 5.2.0.....	10
Check credit.....	11
Send bulk SMS.....	11
Send bulk SMS and check the status.....	11
References.....	12

Introduction

SubitoSMS provides integration APIs using JSON objects

In order to interact with the SMS gateway, all JSON requests must target the URL:

http://www.subitosms.it/ (HTTPS is also available).

SubitoSMS allows to encapsulate bulk messages in 1 single request.

For a bulk requests it will be assigned one id which can be retrieved and used to evaluate the status of each single message in the bulk.

Available actions

The gateway exposes 3 basic actions:

- **credit** – to check for remaining sms credits
- **send** – to send single and bulk sms
- **status** – to check sms logs and delivery statuses

Using JSON

In order to use these JSON APIs you need to make an HTTP request to the target URI specifying the action value and a JSON array.

See the “PHP Examples” section to see a practical example

The Auth array

In order to authenticate each request, an authentication array should be added to each request.

```
{"auth":{"username":"mickey","password":"foobar"},"data":[]}
```

Checking credit

POST /json.php?action=credit

The JSON object must include:

- “auth” array

Example: request

```
{"auth":{"username":"mickey","password":"foobar"},"data":[]}
```

Example: Response

```
{"data":  
{"Italy":"4021","Foreign":"91","ItalyLow":"90","AddressBooks":"12","Polls":"6","  
Credit":null}}
```

- **Italy** – number of high quality sms messages available to Italian carriers
- **Foreign** – number of high quality sms messages available to foreign countries carriers
- **ItalyLow** – number of high quality sms messages available to Italian carriers without alphanumeric sender
- **AddressBooks** – Available AddressBooks on the website
- **Polls** – Available SMS polls on the website
- **Credit** – For future implementations

XML notation

In order to explain the JSON object format, the XML notation is provided.

Request

```
<auth>  
  <username>mickey</username>  
  <password>foobar</password>  
</auth>
```

Response

```
<data>  
  <Italy>4021</Italy>  
  <Foreign>91</Foreign>  
  <ItalyLow>90</ItalyLow>  
  <AddressBooks>12</AddressBooks>  
  <Polls>6</Polls>  
  <Credit>null</Credit>  
</data>
```

Check status

This method allows to request the status of 1 or more group (bulk) of messages.

If you send 1 sms per request, you will have one id per SMS.

If you send more sms per request, you will have one id grouping many SMS

Note: All SMS have their own unique code

Each message reports:

- **sender** – sender that was used
- **dest** – destination number

- **text** – text of the SMS
- **code** – internal code for each single SMS
- **status** – delivery status (see next session)

POST /json.php?action=status

In order to check messages status, the gateway requires:

- “auth” array
- “data” array
 - “id” - The bulk's id you want to get the statuses for.

Example : request

```
{ "auth": [{"username": "mickey", "password": "foobar"}], "data": [{"id": 12293682}, {"id": 12279076}] }
```

Example: Response

The example shows a group (12293682) with 2 SMS and a group (12279076) with 1 SMS

```
[{"id": 12293682, "messages": [{"sender": "Example1", "dest": "+39335123456", "text": "First message", "status": "1", "code": 1803322}], {"id": 12279076, "messages": [{"sender": "Example2", "dest": "+39335123456", "text": "Second message", "status": "1", "code": 1741411}, {"sender": "Example3", "dest": "+39347123456", "text": "Third message", "status": "1", "code": 1741412}]}]
```

Note: If you requested an invalid id in the request, the corresponding array will be empty.

XML notation

In order to explain the JSON object format, the XML notation is provided.

Request

```
<auth>
  <username>mickey</username>
  <password>foobar</password>
</auth>
<data>
  <id>12293682</id>
  <id>12279076</id>
</data>
```

Response

```
<data>
<0>
  <id>12293682</id>
  <messages>
    <0>
      <sender>Example1</sender>
      <dest>+39335123456</dest>
      <text>First message</text>
      <status>1</status>
    </0>
    <1>
      <sender>Example2</sender>
      <dest>+39335123456</dest>
      <text>Second message</text>
      <status>1</status>
    </1>
  </messages>
</0>
<1>
  <id>12293682</id>
  <messages>
    <0>
      <sender>Example3</sender>
      <dest>+39347123456</dest>
      <text>Third message</text>
      <status>1</status>
    </0>
  </messages>
</1>
</data>
```

Status values

- -100: "Gateway error"
- -50: "Destination has been blocked"
- -2: "Not processed"
- -1: "Not processed"
- 0: "Accepted by gateway"
- 1: "Delivered to phone"
- 2: "Not delivered to phone"
- 4: "Buffered"
- 8: "Sent"

- 16: "Refused"

Send SMS messages

When sending more than 1 message, only a group id is returned and this must be used by the "status" method to get statuses.

POST /json.php?action=send

In order to send messages, the gateway requires:

- "auth" array
- "data" array with message details
 - **sender** – Alphanumeric sender (GSM phone number or 11 chars string)
 - **dest** – GSM phone number to send the sms to
 - **text** – Text to be sent (splitting is available to 4 SMS)

Example: request

```
{
  "auth": [{
    "username": "mickey",
    "password": "foobar"
  }],
  "data": [
    {
      "sender": "SubitoSMS",
      "dest": "+39335123456",
      "text": "Hello message",
    },
    {
      "sender": "SubitoSMS",
      "dest": "+393356359515",
      "text": "Hello message",
    }
  ]
}
```

Example: response

```
{"data": {"id": "12293682"}}
```

Note: in order to trace the messages status the id should be saved

XML notation

In order to explain the JSON object format, the XML notation is provided.

Request

```
<auth>
  <username>mickey</username>
  <password>foobar</password>
</auth>
<data>
  <0>
    <sender>SubitoSMS</sender>
    <dest>+39335123456</dest>
    <text>Hello message</text>
  </0>
  <1>
    <sender>SubitoSMS</sender>
    <dest>+39335123456</dest>
    <text>Hello message</text>
  </1>
</data>
```

Response

```
<data>
<id>12293682</id>
</data>
```

Error messages

In case an error arise, the gateway will respond with:

```
{"error":{"message":"Wrong username or password","code":3}}
```

XML Notation

```
<error>
  <message>Wrong username or password</message>
  <code>4</code>
</error>
```

Error codes

- **1** – Syntax error
Auth array is missing or the JSON object submitted is in wrong format
- **2** – Username is undefined
You haven't set the username in the auth object
- **3** - Password is undefined
You haven't set the password in the auth object
- **4** - Wrong username or password
Invalid credentials used

- **5** - You do not have enough credit
You are trying to send more messages than the available credit
- **6** - Action status requires a id to be set
You have requested to check a message status but the id of the message is missing
- **7** – Action xxxx is not defined
You are using an action which was not implemented or the action was not correctly set

PHP Examples

Requirements

In order to use JSON objects in PHP, you need the **JSON** support and **cURL** support.

If you are running PHP > 5, then you should have `json_encode` and `json_decode` built in PHP.

The request wrapper

A basic method to call a remote JSON service could be implemented as:

PHP > 4.3.0

If you are running PHP > 4.3.0, then you need a JSON object converter such as:

<http://mike.teczno.com/JSON/JSON.phps>

Copy the content into the `JSON.php` file and save in the same folder

```
function subito($action,$data){
    require_once("JSON.php"); // Load JSON library
    $request="http://www.subitosms.it/json.php?action=$action";
    $json=new Services_JSON();
    $jco=$json->encode($data);
    $session = curl_init($request);
    curl_setopt ($session, CURLOPT_POST, true);
    curl_setopt ($session, CURLOPT_POSTFIELDS, $jco);
    curl_setopt($session, CURLOPT_HEADER, false);
    curl_setopt($session, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($session);
    curl_close($session);
    return $json->decode($response);
}
```

PHP > 5.2.0

```
function subito($action,$data){
    $action=urlencode($action);
    $request="http://www.subitosms.it/json.php?action=$action";
    $jco=json_encode($data);
    $session = curl_init($request);
    curl_setopt ($session, CURLOPT_POST, true);
    curl_setopt ($session, CURLOPT_POSTFIELDS, $jco);
    curl_setopt($session, CURLOPT_HEADER, false);
```

```

    curl_setopt($session, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($session);
    curl_close($session);
    return json_decode($response);
}

```

Check credit

```

// Create the auth array
$auth=array("username"=>"mickey","password"=>"foobar");
// Set the action
$action="credit";
// The data array is empty
$data=array();
// Incapsulate the auth and the data array
$content=array("auth"=>$auth,"data"=>$data);
// Call the JSON remote method
$obj=subito($action,$content);
// If there is an error, print the message and exit
if (isset($obj->error)) exit("ERROR:".$obj->error->message);
// Print the object received
print_r($obj);
// Echo credit for Italy
echo "You have ".$obj->data->Italy." SMS to Italy";

```

Send bulk SMS

```

// Create the auth array
$auth=array("username"=>"mickey","password"=>"foobar");
// Set the action
$action="send";
// The data array contains 2 messages
$data=array(
0=>array("sender"=>"+39335123456","dest"=>"3351234567","text"=>"First SMS"),
1=>array("sender"=>"SubitoSMS","dest"=>"3351234567","text"=>"Second SMS")
);
// Incapsulate the auth and the data array
$content=array("auth"=>$auth,"data"=>$data);
// Call the JSON remote method
$obj=subito($action,$content);
// If there is an error, print the message and exit
if (isset($obj->error)) exit("ERROR:".$obj->error->message);
// Print the id received
echo("This session took id number".$obj->data->id);

```

Send bulk SMS and check the status

```

// Create the auth array
$auth=array("username"=>"mickey","password"=>"foobar");
// Set the action

```

```

$action="send";
// The data array contains 2 messages
$data=array(
    0=>array("sender"=>"+39335123456","dest"=>"3351234567","text"=>"First SMS"),
    1=>array("sender"=>"SubitoSMS","dest"=>"3351234567","text"=>"Second SMS")
);
// SEND THE MESSAGES
    // Incapsulate the auth and the data array
    $content=array("auth"=>$auth,"data"=>$data);
    // Call the JSON remote method
    $ob=subito($action,$content);
    // If there is an error, print the message and exit
    if (isset($ob->error)) exit("ERROR:".$ob->error->message);
    $bulkid=$ob->data->id;
// CHECK THE STATUS
    // Prepare the array for the next request using the previous id
    $data=array("data"=>array("id"=>$bulkid));
    $content=array("auth"=>$auth,"data"=>$data);
    // Call the JSON remote method
    $ob=subito($action,$content);
    // Check all SMS statuses received
    foreach($ob as $group){
        foreach($group->messages as $message){
            if ($message->status==1) {
                echo("Message to ".$message->dest." was received<br />");
            }
        }
    }
}

```

References

- JSON – <http://www.json.org>
- cURL for PHP - <http://it.php.net/curl>
- JSON for PHP > 4.3.0 - <http://mike.teczno.com/JSON/JSON.phps>
- JSON for PHP > 5.2.0 - <http://it.php.net/manual/it/book.json.php>