



Kapow SMS libraries: Integrating SMS support with your PHP, and Windows applications

Version: 20.1

Publication date: 6th January 2015



Kapow SMS Gateway

All content herein including images, text and the Kapow logo is subject to copyright © and belongs to Kapow, the UK's first SMS Gateway. This document may not be distributed or reproduced without express consent and/or permission from Kapow. Any unauthorised distribution of this document disseminated in print, electronic or any other format is prohibited.

Copyright © 2015, Cygnet Internet Services Ltd.

Head Office

Kapow

Cygnet Internet Services Ltd

90 Long Acre

London

WC2E 9RZ

t - 020 71 835 835

f - 020 71 836 836

e - info@kapow.co.uk

Sales

US Sales (Manhattan, NY): +1 212 913 0553

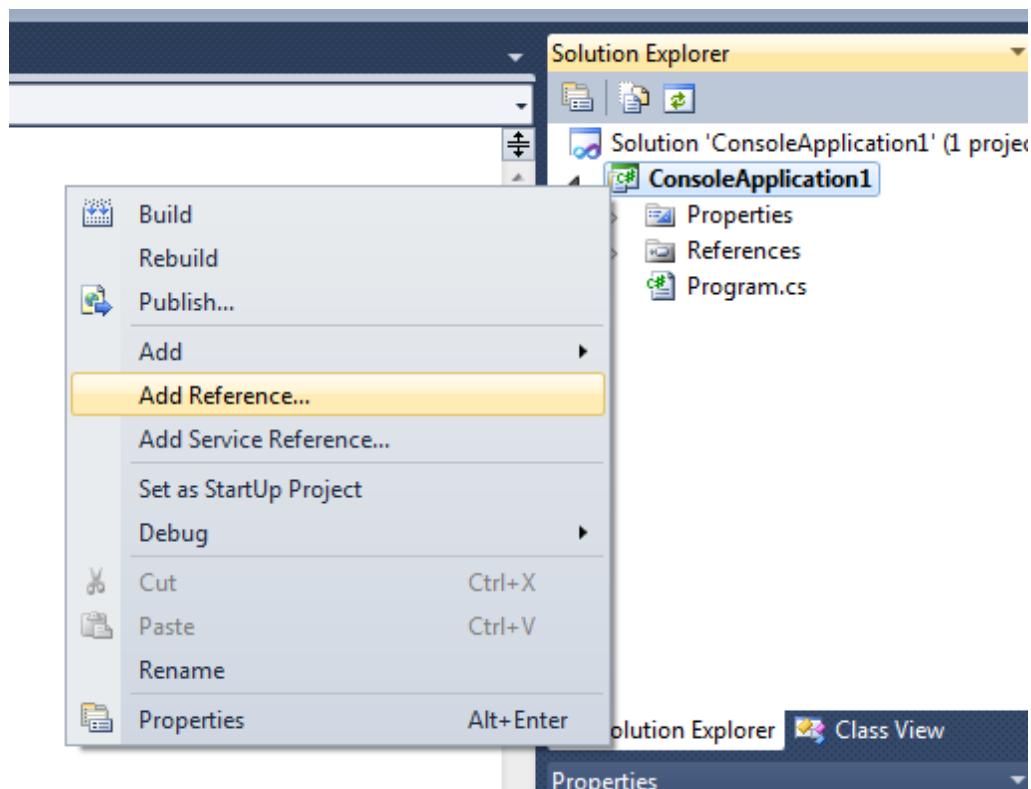
CONTENTS

<u>Dynamic Link Library usage.....</u>	<u>4</u>
<u>Linking to the DLL.....</u>	
<u>Sending a message.....</u>	
<u>Checking credit.....</u>	
<u>Checking status.....</u>	
<u>PHP standard usage.....</u>	<u>7</u>
<u>Referencing the include file.....</u>	
<u>Sending a message.....</u>	
<u>Checking credit.....</u>	
<u>Checking status.....</u>	
<u>Checking day status.....</u>	
<u>PHP with curl usage.....</u>	<u>9</u>
<u>Referencing the include file.....</u>	
<u>Sending a message.....</u>	
<u>Checking credit.....</u>	
<u>Checking status.....</u>	

Dynamic Link Library usage

Linking to the DLL

Before you can call any of the functions with the DLL file, your project must be linked to it first. In Visual Studio, this can be done by right-clicking on the project in Solution Explorer and selecting Add Reference:



Once the dialog pops up, select Browse and then navigate to the location of the kapowlib.dll file.

Make sure that every code file that will be referencing the Kapow DLL also has a using statement like the one below:

```
"using kapowlib;"
```

Sending a message

In order to send a message, you will first need to instantiate the Message object:

```
Message msg = new Message();
```

Once you have a Message object, you will then need to call the SendMessage method. This method will return the a string containing the response from the Kapow script, sendmessage.php for example, will return either the number of remaining credits or an error, only.

The Message class has a method called "SendMessage" which can take upto eleven arguments, in order:

1. username – your username [compulsory]
2. password – your password [compulsory]
3. mobile – the number to send to [compulsory]
4. sms – the body of the message to be sent [compulsory]
5. dlr – delivery report; either an email address or url
6. flash – set to true if this is to be a flash message
7. from_id – from_id
8. url – a url to be redirected to after sending
9. route – route
10. sendtime – for scheduling sending
11. returnid – return id

Checking credit

To check an account's remaining credit, instantiate the Credit class and then call the object CheckCredit method.

CheckCredit takes two arguments:

- username
- password

This will return simply the amount of remaining credits, or an error if encountered.

Checking status

To check the status of an individual message, instantiate the Status class and object's CheckStatus method which takes the following arguments:

- username

- `returnid`

This will return the status of the message, for example 'D' means the message was delivered.

PHP standard usage

Referencing the include file

At the top of each PHP file that will use the Kapow SMS Libraries, either use

- `"include('kapowlib-stdphp.inc');"`, or
- `"require('kapowlib-stdphp.inc');"`

To include the necessary code into your source code.

Sending a message

Call the `send_message` method which takes ten arguments:

1. `username` (string)
2. `password` (string)
3. `mobile` (string)
4. `sms` (string)
5. `from_id` (string)
6. `url` (string)
7. `route` (string)
8. `returnid` (string)
9. `sendtime` (string)
10. `dlr` (string)

This will return the HTTP response as a string, for instance, "OK 123" for message sent (number is remaining credit), "USERPASS" for incorrect credentials.

Checking credit

To check an account's remaining credit, `check_credit` method:

`check_credit` takes two arguments:

1. `username`
2. `password`

This will return simply the amount of remaining credits, or an error if encountered.

Checking status

To check the status of an individual message, instantiate the Status class and object's `check_status` method which takes the following arguments:

1. `username`
2. `returnid`

This will return the status of the message, for example 'D' means the message was delivered.

Checking day status

To check the status of messages for a particular day, use the `check_daystatus` function which takes five arguments:

1. `username`
2. `password`
3. `day`
4. `month`
5. `year`

If day, month and year are not provided, this function will check today's messages only.

PHP with curl usage

Referencing the include file

At the top of each PHP file that will use the Kapow SMS Libraries, either use

- `"include('kapowlib-curl.inc');",` or
- `"require('kapowlib-curl.inc');"`

To include the necessary code in.

Sending a message

Call the `send_message` method which takes ten arguments:

1. `username (string)`
2. `password (string)`
3. `mobile (string)`
4. `sms (string)`
5. `from_id (string)`
6. `url (string)`
7. `route (string)`
8. `returnid (string)`
9. `sendtime (string)`
10. `dlr (string)`

This will return the HTTP response as a string, for instance, "OK" for message sent, "USERPASS" for incorrect credentials.

Checking credit

The `check_credit` method will check an account's remaining credit, and takes two arguments:

- `username (string)`
- `password (string)`

This will return the number of remaining credits or an error code.

Checking status

To check the status of an individual message, instantiate the `Status` class and object's `CheckStatus` method which takes the following arguments:

- username
- returnid

This will return the status of the message, for example 'D' means the message was delivered.