Contents

Introduction

This page will show you **how to download data from web server via http request**.

Request Methods

In website, there are 2 HTTP request methods, **GET** and **POST**. Both of them are used in WidSets too but there is one more addition request method call **EXEC**. Let?s see what are the different.

GET

Request parameters are encoded and appended in URL as Query String, for example.

```
http://yourname.com/index.php?data=Hello&value=World
```

And here is the example request sent to server.

```
GET /indexget.php?data=Hello&value=World HTTP/1.1
```

```
Host: www.yourhost.com
User-Agent: WidSets 3.0.1
```

All request parameters in method have to be string type only. Binary data are not allowed.

Usage Example

To use HTTP request, you have to add http service in widget.xml.

widget.xml

```
<services>
  <service type="http" id="httpService"/>
</services>
```

Next, define fecthing method in helium source code.

helium Source Code

```
void fetchgetdata()
{
  const String URL = "http://www.nuuneoi.com/neoi/widsets/httprequest/indexget.php";

  Prompt prompt = new Prompt(null, "Loading...", null, null);
  prompt.push();

  //http parameters to be used in query
  Value params = [
    "data" => "Hello",
    "value" => "World"
  ];

  //widsets http service parameters
  Value arg = [
    "url" => URL,
    "params" => params
  ];

  // Start Fetching
  call(null, "httpService", "get", arg, ok, nok);

  void ok(Object state, Value ret)
  {
    prompt.pop();
    prompt = null;
    setBubble(null, "Completed: " + ret.toString());
  }

  void nok(Object state, String error)
  {
    prompt.pop();
    prompt = null;
    setBubble(null, "HTTP POST Failed: " + error);
  }
}
```

Let's see at **call** function. Here is the document about it.

```
public void call(Object state,
                 String service,
                 String action,
                 Value argument,
                 SuccessCallback onSuccess,
                 FailureCallback onFailure)
```

Parameters:

```
state - State object to support call, it is passed success and failure callbacks.
service - Name of service to call
action - Name of action to call
argument - Argument to action
onSuccess - Success callback
onFailure - Failure callback
```

So we pass the arguments like this.

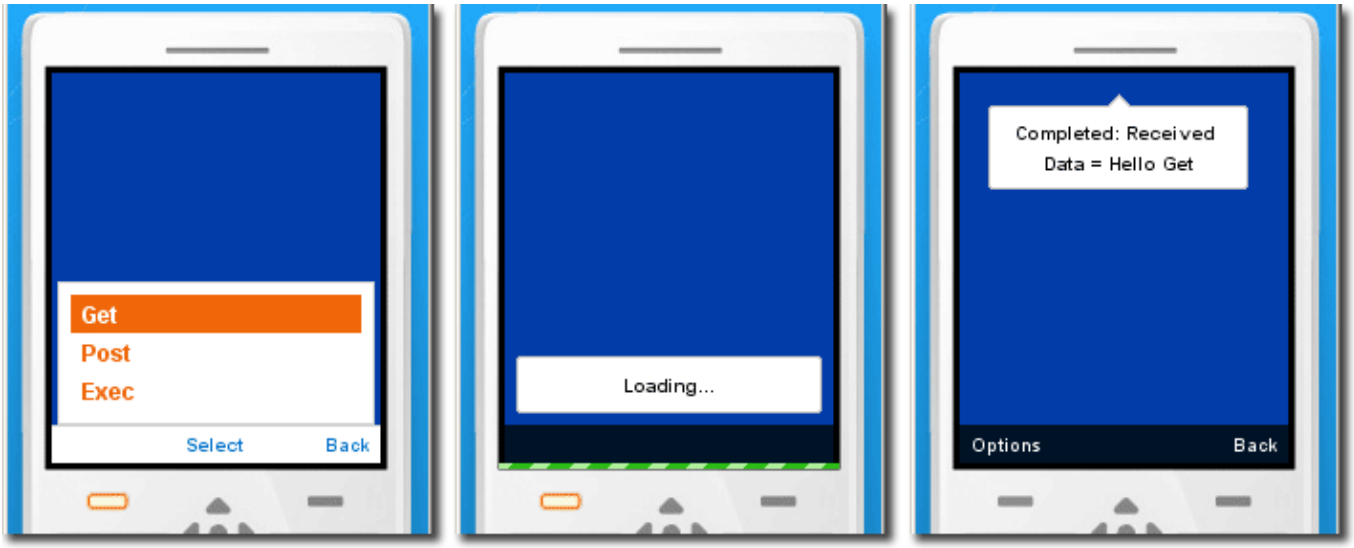
```
call(null, "httpService", "get", arg, ok, nok);
```

- service = httpService (link to http service defined in widget.xml)
- action = get (HTTP request method)
- argument = arg (list of request arguments)
- onSuccess = ok (function that will be called when success)
- onFailure = nok (function that will be called when fail)

indexget.php

```
<?
  echo "Received Data = " . $_GET['data'];
?>
```

Result:



POST

Request parameters are encoded and sent as HTTP request body. HTTP Request sent to server will vary by request parameter. If all request parameters are String type, it will be in **application/x-www-form-urlencoded** form.

```
POST /indexpost.php HTTP/1.1
Host: www.yourhost.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
User-Agent: WidSets 3.0.1

data=Hello&value=World
```

While if there is any binary data in request parameters, it will be in **multipart/form-data** form.

```
POST /indexpost.php HTTP/1.1
Host: www.yourhost.com
Content-Type: multipart/form-data; boundary=--------------------------7d7b11140c94
User-Agent: WidSets 3.0.1

-----------------------------7d7b11140c94
Content-Disposition: form-data; name="data"

Hello
-----------------------------7d7b11140c94
Content-Disposition: form-data; name="value"

World
-----------------------------7d7b11140c94
Content-Disposition: form-data; name="file"; filename="picture.jpg"
Content-Type: image/jpeg

adi98fejl&*^E&^lk
-----------------------------7d7b11140c94
```

Usage Example

Add http service in widget.xml if you didn?t add it yet.

widget.xml

```
<services>
  <service type="http" id="httpService"/>
</services>
```

Next, define fecthing method in helium source code.

helium Source Code

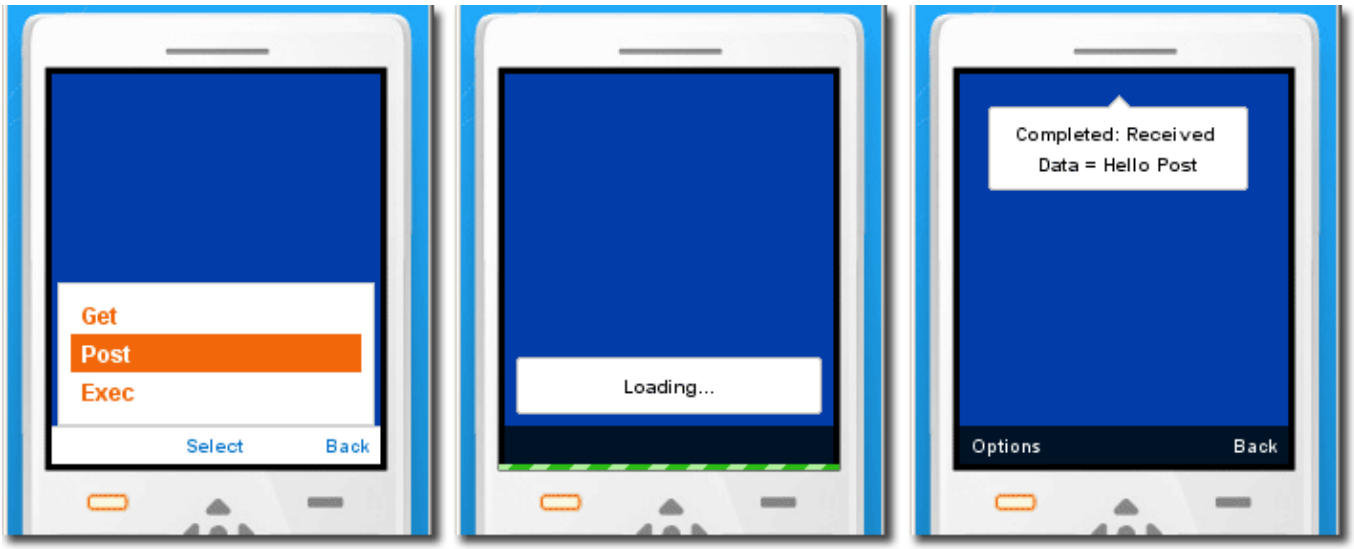
```
void fetchpostdata()
{
  String URL = "http://www.nuuneoi.com/neoi/widsets/httprequest/indexpost.php";

  Prompt prompt = new Prompt(null, "Loading...", null, null);
  prompt.push();
```

```
  ByteArray bArr = new ByteArray(17).set(0, 'H', 'e', 'l', 'l', 'o', 0, 'b', 'i', 'n', 'a', 'r',

  //http parameters to be used in query
  Value params = [
    "data" => "Hello Post",
    "file" => ["image/jpeg", bArr]
  ];

  //widsets http service parameters
  Value arg = [
    "url" => URL,
    "params" => params
  ];

  // Start Fetching
  call(null, "httpService", "post", arg, ok, nok);

  void ok(Object state, Value ret)
  {
    prompt.pop();
    prompt = null;
    setBubble(null, "Completed: " + ret.toString());
  }

  void nok(Object state, String error)
  {
    prompt.pop();
    prompt = null;
    setBubble(null, "HTTP POST Failed: " + error);
  }
}
```

indexpost.php

```
<?
  echo "Received Data = " . $_POST['data'];
  // If you wanna do anything with "file" please use $_FILE and do it like file uploading
?>
```

Result:



EXEC

EXEC is HTTP request **using pre-defined configuration in widget.xml**. It can be used as both GET and POST request method depend on configurations. Strong point of this request type is service URL contains in widget.xml which stays on WidSets Server. So client (user) will never know this URL. This way suits for business widget that developer doesn?t wanna make anyone know the service URL.

Usage Example

Add http service version 2 in widget.xml.

widget.xml

```
<parameters>
  <parameter type="string" name="widgetname" description="Name of widget" editable="no" visible=
  <parameter name="execurl" visible="false" editable="false" protected="true" sendmobile="false"
    <value>http://www.nuuneoi.com/neoi/widsets/httprequest/indexget.php</value>
  </parameter>
  <parameter name="execmethod" value="get"/>
</parameters>

<services>
  <service type="http" id="httpService"/>
  <service type="http" version="2" id="httpExecService">
    <reference from="execurl" to="url" />
    <reference from="execmethod" to="method" />
  </service>
</services>
```

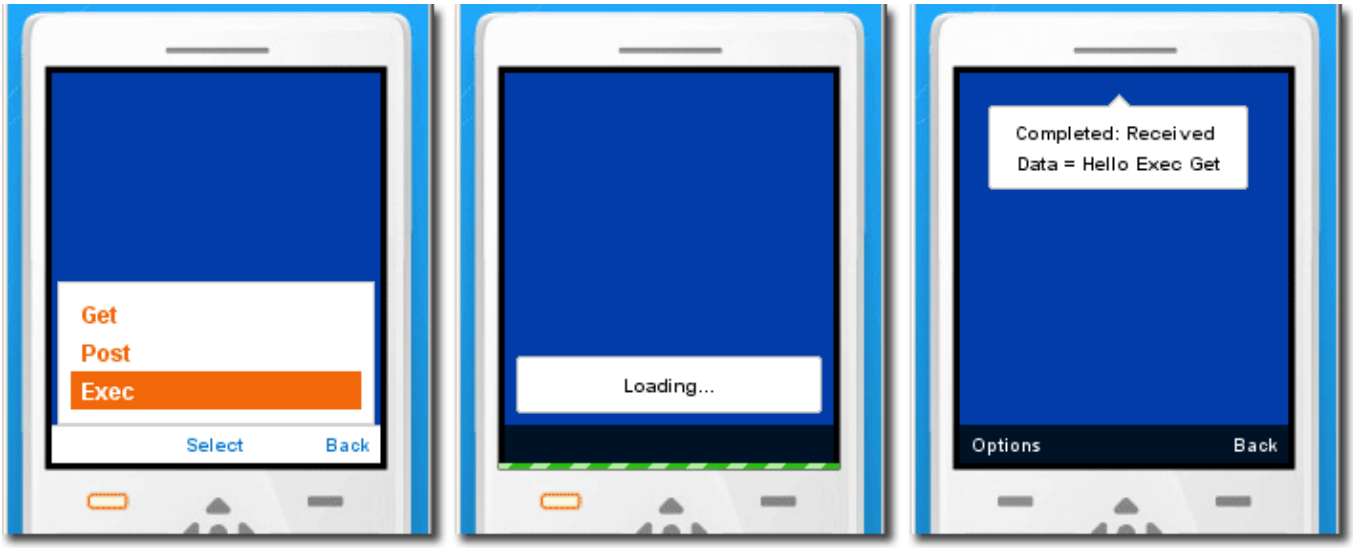
Next, define fecthing method in helium source code.

helium Source Code

```
void fetchexecdata()
{
  Prompt prompt = new Prompt(null, "Loading...", null, null);
  prompt.push();

  //http parameters to be used in query
  Value params = [
    "data" => "Hello Exec Get",
    "value" => "World"
  ];

  // Start Fetching
  call(null, "httpExecService", "exec", params, ok, nok);

  void ok(Object state, Value ret)
  {
    prompt.pop();
    prompt = null;
    setBubble(null, "Completed: " + ret["content"].toString());
  }

  void nok(Object state, String error)
  {
    prompt.pop();
```

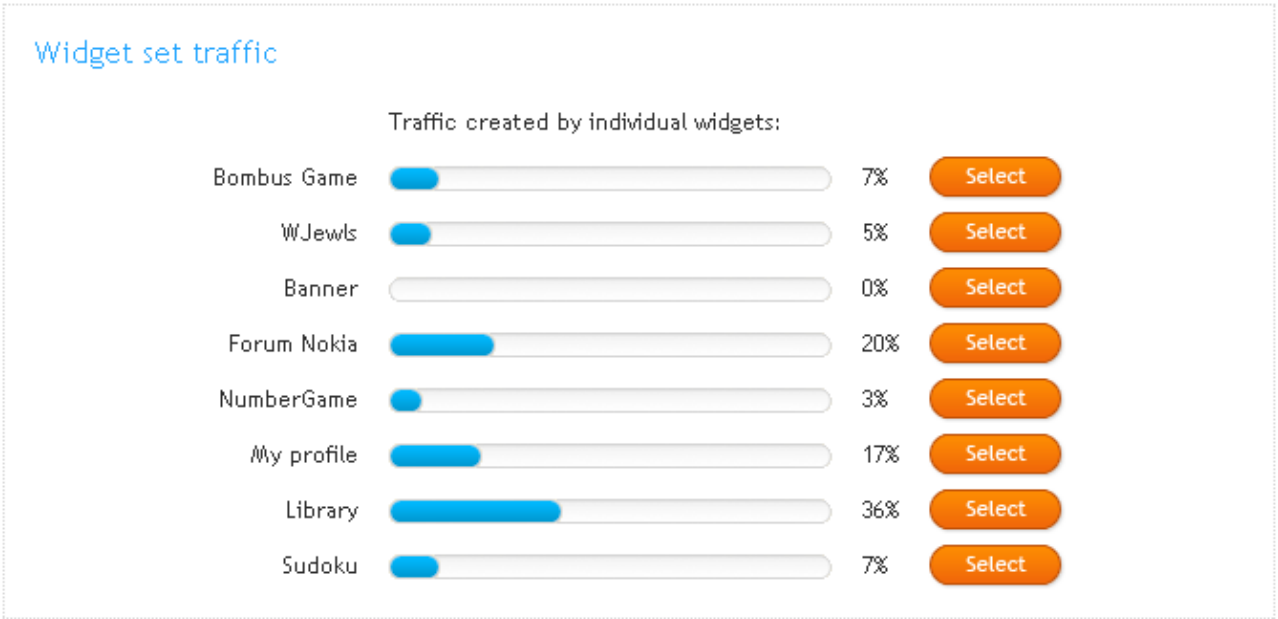
```
    prompt = null;
    setBubble(null, "HTTP POST Failed: " + error);
  }
}
```

Result:

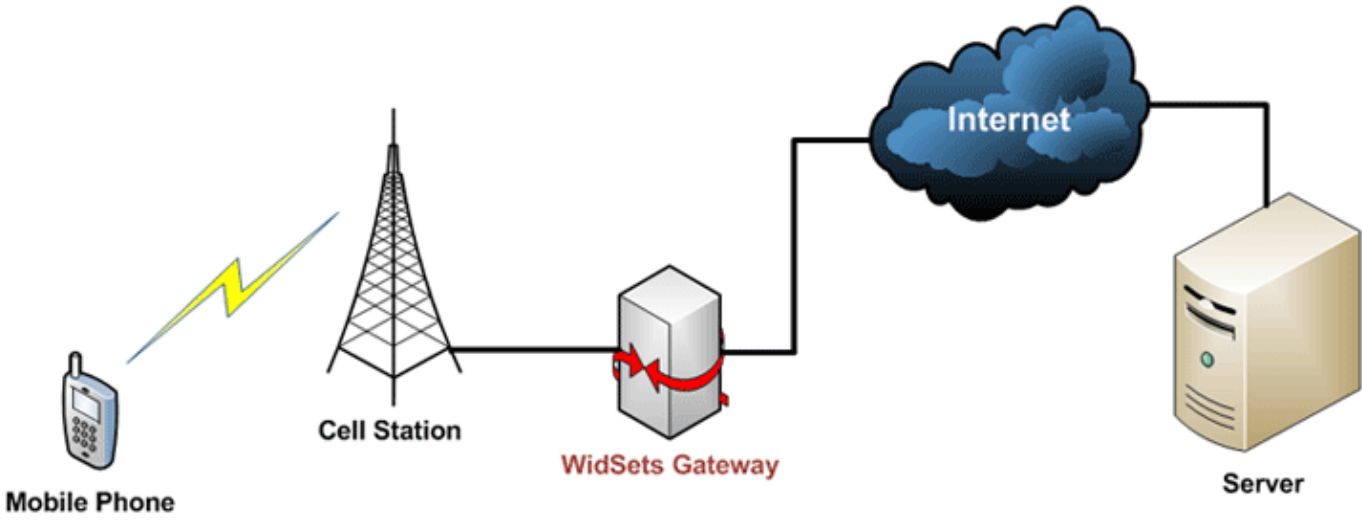


WidSets Gateway

Somebody may already notice that WidSets has ability to collect Traffic Data as you can see in Traffic Manager page of WidSets site.



How could WidSets do that? The answer is WidSets doesn?t connect to web server directly but **it connects via WidSets Gateway**.



From this reason, please remind that **WidSets can?t access private/local IP** in any case. However, you are still able to connect to local service on mobile using Mobile Web Server (How_to_use_MWS_to_provide_local_device_functionality). Nevertheless I definitely **DON'T** recommend it.

Code Snippet

You can download source code for this tutorial from File:WidSets HTTP Request Example.zip

See Also

- WidSets for Rookie EP 1 : First Step to WidSets SDK
- WidSets for Rookie EP 2 : First Compilation with WidSets SDK
- WidSets for Rookie EP 3 : Understand Hello World
- WidSets for Rookie EP 4 : Fasten WidSets Development
- WidSets for Rookie EP 5 : EditPlus Integration
- WidSets for Rookie EP 6 : Softkey Menu
- WidSets for Rookie EP 7 : Standard UI
- WidSets for Rookie EP 8 : Canvas
- WidSets for Rookie EP 9 : Timer
- WidSets for Rookie EP 10 : Key Handling
- **WidSets for Intermediate EP 1 : HTTP Request**
- WidSets for Intermediate EP 2 : HTTP with XML Filter
- WidSets for Advance EP 1 : Life Pictures Project
- WidSets SDK Tips : Emulator Language Changing
- WidSets SDK Tips : Emulator Skin Changing
- WidSets SDK Tips : Add Custom Emulator Skin