

Application for Web-IO:

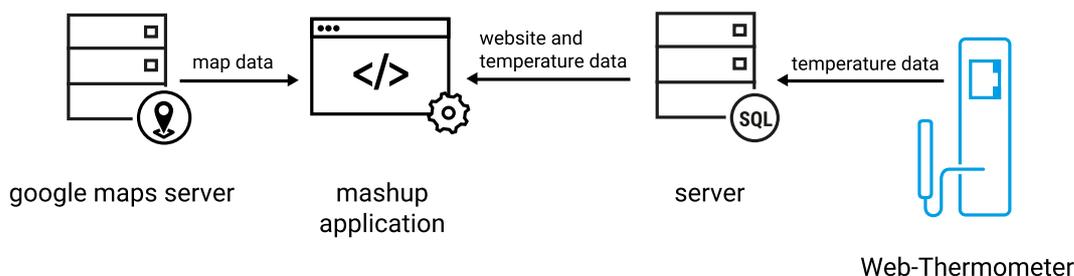
Display measurements and states of Web-IOs in Google Maps

Here a Google Maps map needs to be displayed!

You activate the Google Maps map using the following button, whereby you use the provider Google and (at your own risk in terms of data protection ;) share data with it.

To avoid such extra clicks in the future, try
eu-de-bonn@ec.europa.eu (Regional representative of the European Commission in Bonn)
or support-de@google.com.

The wide availability of the Internet opens up new possibilities in visualization of measurements and automation data. Whereas before all objects in a visualization had to come from the same source, today it's easy to link content from third parties into your own application.



In Web2.0 architecture this technique is referred to as mashup. Data, images, sound and other multimedia content can be used in your own Web site using open API programming interfaces.

A classic example for this technology is Google Maps. The API of Google Maps allows street maps and satellite images from any location to be incorporated into your own page. In addition, measurement values can be merged into these maps.

The following example will illustrate in detail how the temperature measured by a Web-Thermograph is incorporated into a Google Map.

Do you not have a Web Thermometer but would like to try out this example?

No problem: We'll give you a Web Thermometer NTC to try out at no charge for 30 days. Simply fill out the sample order form, and we'll ship you the Web-Thermograph on open invoice. If you return the unit within 30 days, we will credit the invoice in full.

[To sample orders](#)

Preparations

You already have provided your Web-Thermograph

- with power,
- [connected it to your network](#),
- assigned it an IP address - which with [WuTility](#) is no problem.

1. HTML underlying structure of the Web page

- For the sake of simplicity the Web site in this example will contain only the map section and temperature overlay as the content. A Div-Layer is created in the body area of the page with the ID map. This ID can be used to fill the content of the Div-Layer with the Google Maps map data after the Web page is opened.

```

<html>
<head>
<title>Temperature in Map Test</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body onload="load();">
<div id="map" style="width:640px; height:400px;"></div>
</body>
</html>

```

2. Using the Google Maps services

- A basic prerequisite for using the Google Maps services is registration with Google. The Web domain in which the map material is to be displayed must be registered (in our case for example www.WuT.de). Register through the following link: <https://developers.google.com/maps/>.

In return you receive an access key which makes the Google API accessible using a corresponding script.

```

.....
</head>
<script src="http://maps.google.com/?file=api
&amp;ie=iso-8859-1
&amp;oe=iso-8859-1&amp;v=2.x
&amp;key=Registrierungsschlüssel"
type="text/javascript">
</script>
<body onload="load();">
.....

```

3. The PHP script for recalling the temperature from Web-Thermographs

The script shown here must as its own PHP file be invocable on the same PHP server from which the Web site is loaded. It is universally constructed and can be used equally for communication with Web-Thermographs or Web-IOs. The script is opened as needed from the Browser using an AJAX request. The URL is used to transfer all the required parameters.

- **IP:**
IP address of the Web-IO
- **PORT:**
TCP port of the Web-IO (normally 80)
- **COMMAND:**
Possible commands for Web-Thermographs are: single, whereby the sensor number may follow the actual command

Possible commands for the Web-IO Digital are: output, input or counter, whereby the input or output number may follow the actual command. The Web-IO returns the status of the inputs or outputs.

As an additional command the outputs can be set using outputaccess

- **PW:**
Administrator or operator password of the Web-IO (Web-IO Digital only)
- **MASK:**
Specifies in hexadecimal format which outputs are to be set (only Web-IO Digital for outputaccess)
- **STATE:**
Indicates in hex format what state the outputs should be set to (only Web-IO Digital for outputaccess)

The call for querying the temperature would then look as follows:

```
webiorequest.php?IP=URL des Web-Thermographen&PORT=80&COMMAND=single&
```

In reply the script returns the IP address, the device name and the current temperature, semicolon separated.

```
10.40.23.100;WEBIO-03A481;6,8°C
```

The PHP script has the following structure:

```

<?php
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");
header("Cache-Control: no-store, no-cache, must-revalidate");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
parse_str($_SERVER['QUERY_STRING']);
$fp=sockopen($IP, $PORT, $errno, $error, 5);
if (!$fp)
{
    printf("ERROR");
}
else
{
    if ($COMMAND == "outputaccess")
    {
        IF ($MASK == "")
        {
            $MASK="0FFF";
        }
        fputs($fp, "GET /". $COMMAND. "?PW=".$PW."&Mask=".$MASK."&State=".$STATE."&");
    }
    else
    {
        fputs($fp, "GET /". $COMMAND. "?PW=".$PW."&");
    }
    do
    {
        $char=fgetc($fp);
        if($char!=chr(0))
        {
            echo $char;
        }
    }
    while($char!=chr(0));
    fclose($fp);
}
?>

```

This script is saved to the server under webiorequest.php.

4. Combining and displaying temperature and map data

- Calling the needed temperature and map data is controlled by a JavaScript using AJAX.

Loading the Web site calls the Load function using a corresponding instruction in the body tag. For its part, the Load command starts the DataRequest function and transfers the command string required for calling the temperature.

The DataRequest function first checks to see which browser is being used so that the correct method for the XMLHttpRequest can be used (here there are differences between Internet Explorer or Firefox compatible browsers).

Using the PHP script described in step 3. the current temperature is then queried. When the reply is received, the embedded DataReceived function is automatically invoked.

The temperature value is cut out of the receive string and stored together with the current time of day in a new string.

Only now is the map generated. For this the corresponding coordinate pair for the desired location is set in the *point* variable and the desired map inserted in the Div-Layer provided.

The appearance of the map can be specified using various parameters (map or satellite image, navigation element, ...).

Finally a marker is placed at the desired point and the string with temperature and time of day overlaid on the map.

```

<script type="text/javascript">
var map;
var point;
var marker;
var commandstring = `webiorequest.php?IP=klima.wut.de&PORT=80&COMMAND=single&`;

function load()
{

```

```

DataRequest(commandstring);
}

function DataRequest(SendString)
{
var xmlHttp;
try
{ // Internet Explorer
if( window.ActiveXObject )
{
xmlHttp = new ActiveXObject( "Microsoft.XMLHTTP");
}
// Mozilla, Opera und Safari
else if(window.XMLHttpRequest )
{
xmlHttp = new XMLHttpRequest();
}
}
// loading of xmlhttp object failed
catch( excNotLoadable )
{
xmlHttp = false;
alert("no known browser");
}
if (xmlHttp)
{
xmlHttp.onreadystatechange = DataReceived;
xmlHttp.open("GET", SendString, true);
xmlHttp.setRequestHeader("Cache-Control", "no-store, no-cache, must-revalidate");
xmlHttp.setRequestHeader("Expires", "Sat, 05 Nov 2005 00:00:00 GMT");
xmlHttp.setRequestHeader("Pragma", "no-cache");
xmlHttp.send(null);
}
function DataReceived()
{
var Timenow = new Date();
if (xmlHttp.readyState == 4)
{
if (xmlHttp.status == 200)
{
var receivestring = xmlHttp.responseText;
var receivestringpart = receivestring.split(';');
receivestring = 'Außentemperatur um ' + Timenow.getHours() + ':'
+ Timenow.getMinutes() + ' Uhr ' + receivestringpart[2].substring(0,receivestringpart[2].length-2) + '°C';
if( GBrowserIsCompatible() )
{
point = new GLatLng( 51.305099, 7.254627 );
map = new GMap2( document.getElementById( 'map' ) );
map.addControl( new GSmallMapControl() );
map.setCenter( point, 18 );
map.setMapType(G_SATELLITE_MAP)
marker = new GMarker(point);
map.addOverlay(marker);
var content = "<h3>Prima Klima bei W&T</h3>";
content += "<br><a href='http://klima.wut.de'>" + receivestring + "</a>";
marker.openInfoWindowHtml(content);
}
}
}
}
}
}
}
</script>

```

This example intentionally supports only a few basic functions of mashup and is optimized for the models 57714 and 57715 of the Web Thermometer. For the other Web-IO models some adaptations may have to be made. Also, Google Maps API offers many additional functions which were not used here.

Please note that in the example offered here for downloading the registration key matching the domain must be entered.

[↓ Download program example](#)

Do you not have a Web Thermometer but would like to try out this example?

No problem: We'll give you a Web Thermometer NTC to try out at no charge for 30 days. Simply fill out the sample order form, and we'll ship you the Web-Thermograph on open invoice. If you return the unit within 30 days, we will credit the invoice in full.

[To sample orders](#) 



[We are available to you in person:](#)

Wiesemann & Theis GmbH
Porschestra. 12
42279 Wuppertal
Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
Fax: +49 202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, subject to mistakes and changes: Since we can make mistakes, none of our statements should be applied without verification. Please let us know of any errors or misunderstandings you find so that we can become aware of and eliminate them.

[Data Privacy](#)