

# MOD Graphical User Interface (GUI)

Author: Yannis Plaschko

(TINF20C, SWE I Praxisprojekt 2021/2022)

**Project:** Websockets in a lwIP HTTP Server

**Customer:** Rentschler & Holder  
Rotebühlplatz 41  
70178 Stuttgart

**Supplier:** Team 4: Laura Reeken, inf20051@lehre.dhbw-stuttgart.de  
Benjamin Esenwein, inf20074@lehre.dhbw-stuttgart.de  
Yannis Plaschko, inf20093@lehre.dhbw-stuttgart.de  
Maximilian Meier, inf20084@lehre.dhbw-stuttgart.de  
Lucas Kaczynski, inf20147@lehre.dhbw-stuttgart.de  
Isabel Schwalm, inf20085@lehre.dhbw-stuttgart.de  
Rotebühlplatz 41  
70178 Stuttgart

Version	Date	Author	Comment
0.1	29.04.2021	Yannis Plaschko	created
0.9	02.05.2021	Yannis Plaschko	Filled in details
1.0	04.05.2022	Yannis Plaschko	Finalized document
1.1	05.05.2022	Yannis Plaschko	Document format fix

## CONTENTS

<b>1. Scope .....</b>	<b>3</b>
<b>2. Abbreviations.....</b>	<b>3</b>
<b>3. Module Requirements .....</b>	<b>3</b>
<b>3.1 Demo Website.....</b>	<b>3</b>
<b>3.2 Requirements .....</b>	<b>3</b>
<b>3.3 Module Context .....</b>	<b>3</b>
<b>4. Analysis .....</b>	<b>3</b>
<b>5. Design.....</b>	<b>4</b>
<b>6. Implementation .....</b>	<b>4</b>
<b>7. Module Tests.....</b>	<b>5</b>
<b>7.1 Module Testplan.....</b>	<b>5</b>
<b>7.2. Module Testreport.....</b>	<b>5</b>
<b>8. Summary .....</b>	<b>5</b>

## 1. Scope

This documentation discusses the Graphical User Interface of the lwIP Server to test the functionality of the restful API.

## 2. Abbreviations

- GUI – Graphical User Interface
- API – application programming interface
- lwIP – Lightweight IP

## 3. Module Requirements

### 3.1 Demo Website

This module serves a website to the user, which can be used to communicate with the lwIP Server.

Following Features have been implemented:

- The user can issue a restful API request to “<lwIP-adress>/identification”, which the server correctly recognizes.
- A button exists on the demo website, that can forward the user to the “<lwIP-adress>/identification” endpoint.

### 3.2 Requirements

This module implements the following Project requirements: /NF10/, /F30/.

/NF10/: For this nonfunctional requirement to be met the additional code must be kept to a minimum. This requirement has been met. The design was kept minimal while also being clear, concise and demonstrating the possibility of having a much more stylized.

/F30/: This requirement is based on the User Interface of the Testclient. The Testclient enables the user to interact with the API-Endpoints to receive data from the Server. This requirement has been implemented.

### 3.3 Module Context

This module provides a simple Graphical User Interface to explain how to get a JSON-Object containing Information about the server. It achieves this by a text field containing instructions and a button to execute the request.

## 4. Analysis

To test whether the Rest-API endpoint works and can return a correct JSON-Object a simple GUI must be created which both explains what a user should do as well as what is supposed to happen to pass the test. To achieve this the page needs little more than a text paragraph explaining what the user is supposed to do and the expected outcome and a simple way to start the test.

## 5. Design



### lwIP Test Application

The following link goes to the /identification endpoint to test GET-Request functionality of said endpoint. The Endpoint should return a json object containing information. Among other things it should contain the IP-Address of the lwIP-host application

Get Test

## 6. Implementation

The User Interface gets implemented with the httpserver-netcon.c file. Through this a high compatibility with the remaining lwIP code is achieved and further Improvements to the lwIP code base can be easier developed. The HTML Code is served using a static char array.

```
static const char http_html_hdr[] = "HTTP/1.1 200 OK\r\nContent-type: text/html\r\n\r\n";
```

```
static const char http_index_html[] =
"<html><head><title>Test</title><style>.clickme{background-color: #EEEEEE;padding: 8px 20px;text-decoration:none;font-weight:bold;border-radius:5px;cursor:pointer;}.danger{background-color:#FF0040;color:#FFFFFF;}.danger:hover{background-color:#EB003B;color:#FFFFFF;}</style></head><body style=\"background-color: #dfdfe2;\"><img src=\"https://static.wikia.nocookie.net/lwip/images/b/bc/Wiki.png/\"><h1 style=\"font-family:'Courier New'\">lwIP Test Application</h1><p style=\"background-color: #d0d0d5; font-family:'Courier New'; font-size: 20px;\">The following link goes to the /identification endpoint to test GET-Request functionality of said endpoint. The Endpoint should return a json object containing information. Among other things it should contain the IP-Address of the lwIP-host application</p><a href=\"/identification\" class=\"clickme danger\">Get Test</a></body></html>";
```

## 7. Module Tests

This Section contains Tests, whether the Requirements raised by this module function according to the specifications. Further Descriptions of the Tests can be found in the System test plan.

### 7.1 Module Testplan

Test-ID	Req. - ID	Functionality
4	LF90: Call API /identification from root	The test case verifies whether a click on the link redirects to displays information about the Server.

### 7.2. Module Testreport

Test-ID	PASS / FAIL	When failed: Observation	Tester
1	FAIL	Couldn't be tested – no implementation of WebSocket's / Testclient	Y. Plaschko
2	FAIL	Couldn't be tested – no implementation of WebSocket's / Testclient	Y. Plaschko
3	FAIL	Couldn't be tested – no implementation of WebSocket's / Testclient	Y. Plaschko

## 8. Summary

This modules requirements couldn't be implemented due to time constraints, thus it was impossible to execute the Testplan. In Accordance with Mr. Rentschler the focus was shifted on different requirements and further work on the GUI was discontinued.