Moduldokumentation

(MOD)

(TINF18C, SWE I Praxisprojekt 2019/2020)

Modul

NetworkService

Project: Profinet DCP Client als WEB-Applikation

Customer: *Rentschler & Ewertz*
Rotebühlplatz 41
70178 Stuttgart

Supplier: *Team 2 (Jannik Schwarz, Sinan Yurttadur, Noah Broß, Nicolas Breuninger, Marvin Sonntag, Rene Scholz)*
Rotebühlplatz 41
70178 Stuttgart

Version	Date	Author	Comment
0.1	28.04.2020	Nicolas Breuninger	First draft
1.0	06.05.2020	Nicolas Breuninger	Added most of the information except tests

1. Content

2. History

Version	Datum	Autor(en)	Kommentare
0.1	28.04.2020	Nicolas Breuninger	First draft
1.0	06.05.2020	Nicolas Breuninger	Added most of the information except tests

3. Scope

The Module Documentation (MOD) describes the architecture, the interfaces and the main features of the module. It also describes the module/component test including the results. It can also serve as a programming or integration manual for the module. If there are some risks related to the module itself, they shall be noted and commented within this document.

Die Moduldokumentation beschreibt die Architektur, die Schnittstellen und die Hauptmerkmale des Moduls. Außerdem werden die Modul bzw. Komponententests einschließlich der Ergebnisse beschrieben und dokumentiert. Die MOD dient bei Bedarf auch als Programmier- oder Integrationshandbuch für das Modul. Wenn bestimmte Risiken direkt mit der Verwendung des Moduls verknüpft sind, so sind sie in diesem Dokument zu benennen und zu kommentieren.

4. Definitions

<Wichtige Abkürzungen, Terminologien, Begriffe und Worte hier erklären>

5. Module Requirements

5.1. User View

This module is used every time communication between the frontend and backend is required. This module is also used to store the information for the GUI.
Every time a device is clicked in the UI, this module requests data for the selected device.

5.2. Requirements

A connection to the backend-API is required to retrieve data and send requests.

5.3. Module Context

This module is important for the GUI, as it holds all the data. And it works with the backend API to retrieve data from the devices in the network.

6. Analysis

This module is dependent on the HttpClient provided by Angular. As well as the backend.

7. Design

This module was designed around the backend-end data structures. It tries to replicate these structures to allow for an easy communication between front- and backend.

It’s a simple HTTP-API for the frontend used to communicate with the backend. It is a internal API of the GUI.

Link to the source files (https://github.com/nicolasbreuni/Tinf18C_Team_2_Profinet_DCP_Client/blob/master/SOURCE/dcp-client-frontend/src/app/services/device.service.ts)

7.1. **Risks**

If this module fails, no communication between front- and backend is possible. Also no data could be displayed by the GUI.

8. Implementation

This module implementation was heavily influenced by the backend specification. It boils down to a recreation of the provided API-calls by the backend.

9. Module Test

< Wie wird die Komponente getestet? White-Box und Black-Box-View! Dokumentation von Vorgehen und Ergebnissen. Bei Bedarf entsprechend erweitern.>

9.1. Component Testplan

Test-ID	Feature ID	Test Specification (Description or TCS)

9.2. Component Testreport

Test-ID	Pass/Fail	If failed: Test Observation	Date	Tester

10. Summary

Strengths:

No third-party-software was used.

Weaknesses:

This module is not reactive.

Possible extensions:

The communication between frontend and backend could be reactive. If this module was implemented in a reactive way, it would still function, without any changes to the other modules.

11. Appendix

11.1. References

11.2. Interface Definitions

11.3. Code

< Code nur, wenn es dem Verständnis dienlich ist >

11.4. Module Test Cases

Test-ID	Name	Description	Test Steps		
			Step	Action	Expected Result