

# MOD01: Server connection

TINF22F, Software Engineering I  
Praxisproject 2023/24

## *Project*

AAS-Webclient

## *Customer*

Markus Rentschler, Ivan Bogicevic  
Lerchenstraße 1, 70178 Stuttgart

## *Supplier*

Armin Taktar (Project Leader)  
Lara Lorke (System Architect)  
David Bauer (Product Manager)  
Ümmühan Ay (Documentation)  
Rafael Sancho Pernas (Developer)  
Kyle Zieher (Test Manager)

## *Author*

David Bauer 06.05.2024

# CONTENTS

<b>1 SCOPE .....</b>	<b>3</b>
<b>2 GLOSSARY .....</b>	<b>3</b>
<b>3 MODULE REQUIREMENTS .....</b>	<b>3</b>
3.1 USER VIEW.....	3
3.2 REQUIREMENTS.....	3
<b>4 ANALYSIS .....</b>	<b>3</b>
<b>5 DESIGN .....</b>	<b>3</b>
<b>6 IMPLEMENTATION .....</b>	<b>4</b>
<b>7 MODULE TESTS .....</b>	<b>4</b>

## 1 Scope

The AAS web client is a display interface for presenting the data taken from an AASX server. Therefore, correct generation and display of Nameplates are important.

## 2 Glossary

- AAS: Asset Administration Shell
- AASX: file format to store an asset
- NPG: Nameplate Generator

## 3 Module Requirements

### 3.1 User View

The user can view the Nameplate for the specific Asset at will, they are able to generate and download the Nameplate in the form of a SVG-File. This is triggered by a button, that is present in every Asset.

### 3.2 Requirements

REQ/ NPG-API (The API of the NPG must be available)

## 4 Analysis

A API request must be made to the NPG, a valid response must be received. The data from the response must be displayed correctly and a download feature must be present.

## 5 Design

The image shows a partial view of a web application interface. It features a light gray sidebar on the left. The main content area has a white background. At the top, there is a white rectangular input field. Below it, a horizontal line separates the header from the main content. Further down, there are two rounded rectangular buttons stacked vertically. The top button is labeled 'JSON downloaden' and the bottom button is labeled 'Nameplate sehen'. Both buttons have a subtle shadow and a thin border.

When the User clicks on the <Nameplate sehen>-Button the SVG-File will be shown fullscreen and can be downloaded from there.

## 6 Implementation

In the file `assetBody.js` with the function `openNameplateGen` the API from the Nameplate Generator gets send a request with the Asset ID.

In the previous project, this functionality wasn't implemented. There was no basis for the function.

The `openNameplateGen` creates a link that is automatically pressed to go to the response SVG-File.

```
{
const data = await
fetch("http://localhost:8080/Nameplate/GenerateByReference/"+shellBody.hide.URL)
.then(() => {
    const link = document.createElement('a')
    link.href = data;
    link.download = 'nameplate.svg';
    document.body.appendChild(link);
    link.click();

    document.body.removeChild(link);
}).catch()
}
```

Sometimes the API returns no response, this is likely due to the Nameplate Generator server being down or unreachable.

## 7 Module Tests

Testing for this module is conducted through usability tests. An Asset is chosen and then the <Nameplate sehen>-Button is pressed. The expected result is a redirect to a new page where the SVG-File that has been returned by the API of the specific Asset is shown. There it can be downloaded.

