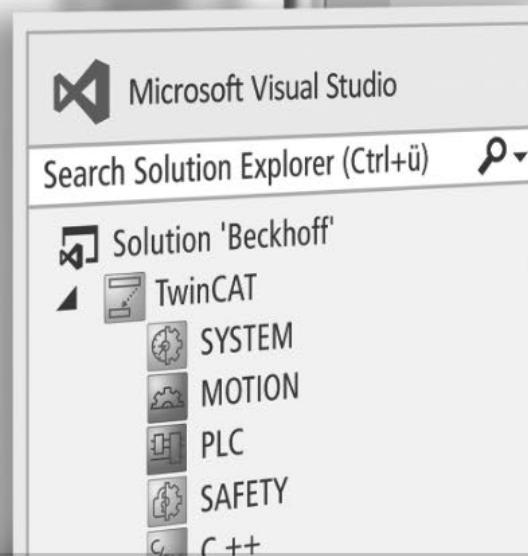


HMI UpDate Systemintegratoren 2023



IHRE MODERATOREN HEUTE

Daniel Hausdorf
HMI / Support

d.hausdorf@beckhoff.com



Gorazd Jerič
Vision / Training

g.jeric@beckhoff.com



Begrüßung	Gorazd Jerič
Einführung / Übersicht Workshop Inhalt	Gorazd Jerič Daniel Hausdorf
PLC / HMI Verbindung	Gorazd Jerič
Kaffee Pause 	—
Workflow bei der Komponenten Entwicklung Teil 1	Daniel Hausdorf
Mittag 	—
Workflow bei der Komponenten Entwicklung Teil 2	Daniel Hausdorf
Kaffee Pause 	—
DevTools / Testing	Gorazd Jerič
Ende	Gorazd Jerič Daniel Hausdorf



HMI VARIABLEN MAPPEN

- Mapping Varianten von PLC Variablen in der HMI
 - Manuell
 - Automappen



MANUELLES MAPPEN

Das Variablen - Mappen manuell über die HMI erstellen

The screenshot shows the 'TwinCAT HMI Configuration' interface. On the left, there's a navigation tree with categories like Server Symbols, Functions, and Users and Groups. The main area displays a table with columns: Name, Datatype, and Mapped to. The table lists symbols from the PLC1 namespace, such as PLC1.Constants, PLC1.Global_Variables, PLC1.Global_Version, PLC1.HmiFunctions, and PLC1.HmiMapping. Under PLC1.HmiMapping, several symbols are listed: nAutoMapping, nAutoMappingHidden, nAutoMappingReadOnly, nAutoMappingUserGroup, nNoAutoMapping, stAutoMapping, and TwinCAT_SystemInfoVarList. A context menu is open over the 'nAutoMapping' symbol, with the 'Map Symbol' option highlighted by a red box.

Name	Datatype	Mapped to
ADS	object	
PLC1	object	
Constants	PLC1.Constants	
Global_Variables	PLC1.Global_Variables	
Global_Version	PLC1.Global_Version	
HmiFunctions	PLC1.HmiFunctions	
HmiMapping	PLC1.HmiMapping	
nAutoMapping	INT	Map Symbol
nAutoMappingHidden	INT	Unmap
nAutoMappingReadOnly	INT	Refresh Mapping
nAutoMappingUserGroup	INT	Rename Mapping
nNoAutoMapping	INT	Map Automap Symbols
stAutoMapping	PLC1.ST_	Edit Permissions
TwinCAT_SystemInfoVarList	PLC1.Twi	



AUTOMAPPEN

Das Automapping wird durch Attributte in der PLC definiert

Syntax

```
1 VAR
2
3 {attribute 'attribute'}
4 Variable_die_Beinflusst_wird : INT;
5
6 END_VAR
```



AUTOMAPPEN

Das Automapping wird durch Attributte in der PLC definiert

Syntax

```
1 VAR  
2  
3 {attribute 'attribute'}  
4 Variable_die_Beinflusst_wird : INT;  
5  
6 END_VAR
```

Attribute

- {attribute 'TcHmiSymbol.AddSymbol'}
- {attribute 'TcHmiSymbol.ReadOnly'}
- {attribute 'TcHmiSymbol.AddSymbol.Hidden'}
- {attribute 'TcHmiSymbol.AddSymbol.UserGroups' := '__SystemAdministrators, Operators'}



AUTOMAPPEN

Kein Mappen aktiv

```
1 VAR  
2  
3   nNoAutoMapping : INT;  
4  
5 END_VAR
```



AUTOMAPPEN

Mappen aktiv

```
1 VAR
2
3 {attribute 'TcHmiSymbol.AddSymbol'}
4 nAutoMapping : INT;
5
6 END_VAR
```



AUTOMAPPEN

Mappen Read Only

```
1 VAR
2
3 {attribute 'TcHmiSymbol.AddSymbol'}
4 {attribute 'TcHmiSymbol.ReadOnly'}
5 nAutoMappingReadOnly : INT;
6
7 END_VAR
```



AUTOMAPPEN

Mappen Hidden

```
1 VAR
2
3 {attribute 'TcHmiSymbol.AddSymbol'}
4 {attribute 'TcHmiSymbol.AddSymbol.Hidden'}
5 {attribute 'TcHmiSymbol.AddSymbol.UserGroups' := '__SystemAdministrators, Operators'}
6 nAutoMappingHidden : INT;
7
8 END_VAR
```

ID:	...	
Name	Value	Datatype
PLC1.HmiMapping.nAutoMapping	INT	
PLC1.HmiMapping.nAutoMappingReadOnly	INT	
PLC1.HmiMapping.nAutoMappingUserGroup	INT	
PLC1.HmiMapping.stAutoMapping	PLC1.ST_HmiMapping	

ID:	...	
Name	Value	Datatype
PLC1.HmiMapping.nAutoMapping	INT	
PLC1.HmiMapping.nAutoMappingHidden	INT	
PLC1.HmiMapping.nAutoMappingReadOnly	INT	
PLC1.HmiMapping.nAutoMappingUserGroup	INT	
PLC1.HmiMapping.stAutoMapping	PLC1.ST_HmiMapping	



AUTOMAPPEN

Mappen UserGroups

```
1 VAR
2
3 {attribute 'TcHmiSymbol.AddSymbol'}
4 {attribute 'TcHmiSymbol.AddSymbol.UserGroups' := '__SystemAdministrators, Operators'}
5 nAutoMappingUserGroup : INT;
6
7 END_VAR
```



AUTOMAPPEN

Mappen Strukturen

```
1 VAR
2
3 {attribute 'TcHmiSymbol.AddSymbol'}
4 {attribute 'TcHmiSymbol.AddSymbol.UserGroups' := '__SystemAdministrators, Operators'}
5 stAutoMapping : ST_HmiMapping := (sBmk :='AutoMapping');
6
7 END_VAR
```



AUTOMAPPEN

In der HMI aktualisieren

1. Keine Mappings vorhanden

Name	Value	Datatype

2. Symbole in der PLC

Name	Type
PLC1.H	object
INT	INT
PLC1.S	object

3. Map Automap Symbols

Map Symbol
Unmap
Refresh Mapping
Rename Mapping
Map Automap Symbols (highlighted)
Edit Permissions
Copy symbol expression

4. Gemappte Symbole

Name	Type
PLC1.HmiMapping.nAut	INT
PLC1.HmiMapping.nAut	INT
PLC1.HmiMapping.nAut	INT
PLC1.HmiMapping.stAut	PLC1.ST_Hm



PLC METHODEN UND PROPERTIES AUFRUFEN

Aus der HMI kann auf folgende Elemente zugegriffen werden

- PLC Properties
- PLC Methoden
- PLC Interfaces



PLC PROPERTIES

Die Properties in der PLC sichtbar machen

```
1 VAR  
2  
3 {attribute 'monitoring' := 'call'}  
4 PROPERTY Prop : BOOL  
5  
6 END_VAR
```

```
1 {attribute 'monitoring' := 'call'}.  
2 PROPERTY ValueA : UDINT.  
3
```

TwinCAT HMI Configuration

The screenshot shows the TwinCAT HMI Configuration interface. On the left is a tree view of symbols under 'Server Symbols'. Under 'ADS', there is a node 'fbHmiAddition' which is expanded to show its properties: 'Result', 'ValueA', and 'ValueB'. These three properties are highlighted with a red box. To the right of the tree is a table with columns 'Name' and 'Datatype'. The table lists various symbols from the ADS tree, including 'PLC1', 'Global_Variables', 'Global_Version', 'HmiFunctionAddition', 'bTestAddition', and 'fbHmiAddition'. The 'fbHmiAddition' row is also highlighted with a blue box. The 'Result', 'ValueA', and 'ValueB' entries under it are also highlighted with a red box.

Name	Datatype
PLC1	object
Constants	PLC1.Constants
Global_Variables	PLC1.Global_Variables
Global_Version	PLC1.Global_Version
HmiFunctionAddition	PLC1.HmiFunctionAddition
bTestAddition	BOOL
fbHmiAddition	PLC1.FB_HmiAddition
Result	PLC1.PROPERTY-GET-UDINT
ValueA	PLC1.PROPERTY-GET-SET-UDINT
ValueB	PLC1.PROPERTY-GET-SET-UDINT
_HmiData	PLC1.ST_HmiAdditionData
iHmiAddition	PLC1.I_HmiAddition
stHmiAdditionData	PLC1.ST_HmiAdditionData



PLC METHODEN

Die Methoden in der PLC sichtbar machen

```
1 {attribute 'TcRpcEnable'}
2 METHOD MyMethod : BOOL
3 VAR_INPUT
4     bInput : BOOL;
5 END_VAR
```

```
1 {attribute 'TcRpcEnable'}.
2 METHOD AdditionValues : UDINT.
3 VAR_INPUT.
4     nValueA : UDINT;
5     nValueB : UDINT;
6 END_VAR
```

```
1 {attribute 'TcRpcEnable'}.
2 FUNCTION_BLOCK PUBLIC FB_HmiAddition IMPLEMENTS T_HmiAddition.
3 VAR INPUT.
4 END_VAR.
```

TwinCAT HMI Configuration

Name	Datatype
ADS	object
PLC1	object
Constants	PLC1.Constants
Global_Variables	PLC1.Global_Variables
Global_Version	PLC1.Global_Version
HmiFunctionAddition	PLC1.HmiFunctionAddition
bTestAddition	BOOL
fbHmiAddition	PLC1.FB_HmiAddition
AdditionValues	PLC1.METHOD-RETURNING-UDINT
AdditionWithProperty	PLC1.METHOD-RETURNING-UDINT
Result	PLC1.PROPERTY-GET-UDINT
ValueA	PLC1.PROPERTY-GET-SET-UDINT
ValueB	PLC1.PROPERTY-GET-SET-UDINT

TwinCAT HMI Configuration

Name	Datatype
PLC1.METHOD-RETURNING-UDINT-PARAMETERS-nValueA-UDINT-nValueB-UDINT	

Context menu options:

- Create new server symbol
- Copy symbol expression
- Unmap
- Toggle 'Use mapping'
- Map to other symbol
- Refresh Mapping
- Edit
- Hierarchy Settings
- Remove Hierarchy Settings
- Write Value to symbol
- Call method
- Edit Permissions



PLC METHODEN

Methoden werden mithilfe JavaScript / TypeScript aufgerufen

- Ohne Parameter

```
1 TcHmi.Symbol.readEx2("%s%PLC1.MAIN.fbTest.MethodWithoutParam%/s%", function (data) {  
2     console.log(data);  
3 });  
4 );
```

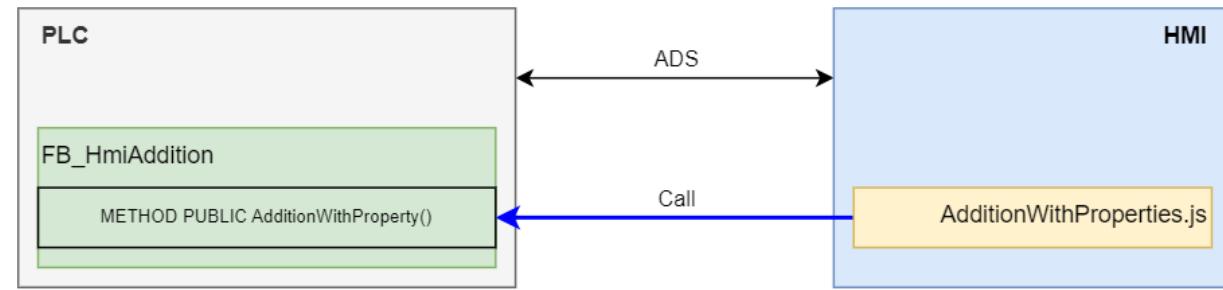
- Mit Parameter

```
1 TcHmi.Symbol.writeEx("%s%PLC1.MAIN.fbTest.MyMethod%/s%", { bInput: true }, function (data) {  
2     console.log(data);  
3 });  
4 );
```



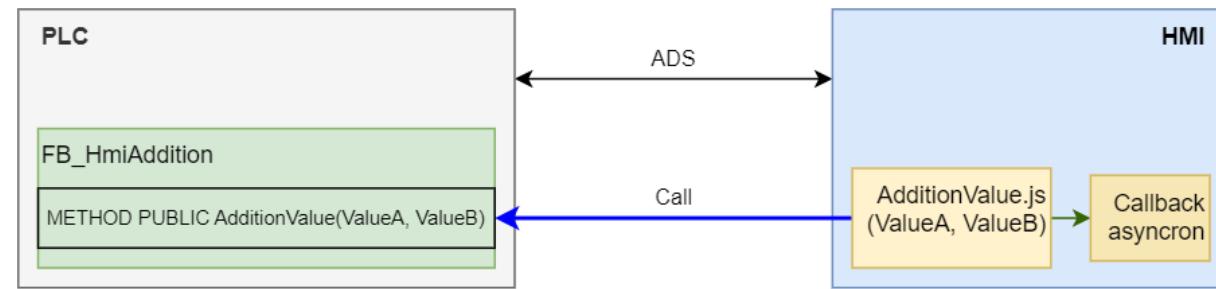
AUFRUFSCHEMA

Ohne Parameter



AUFRUFSCHEMA

Mit Parameter



BEISPIEL “LIZENZEN”

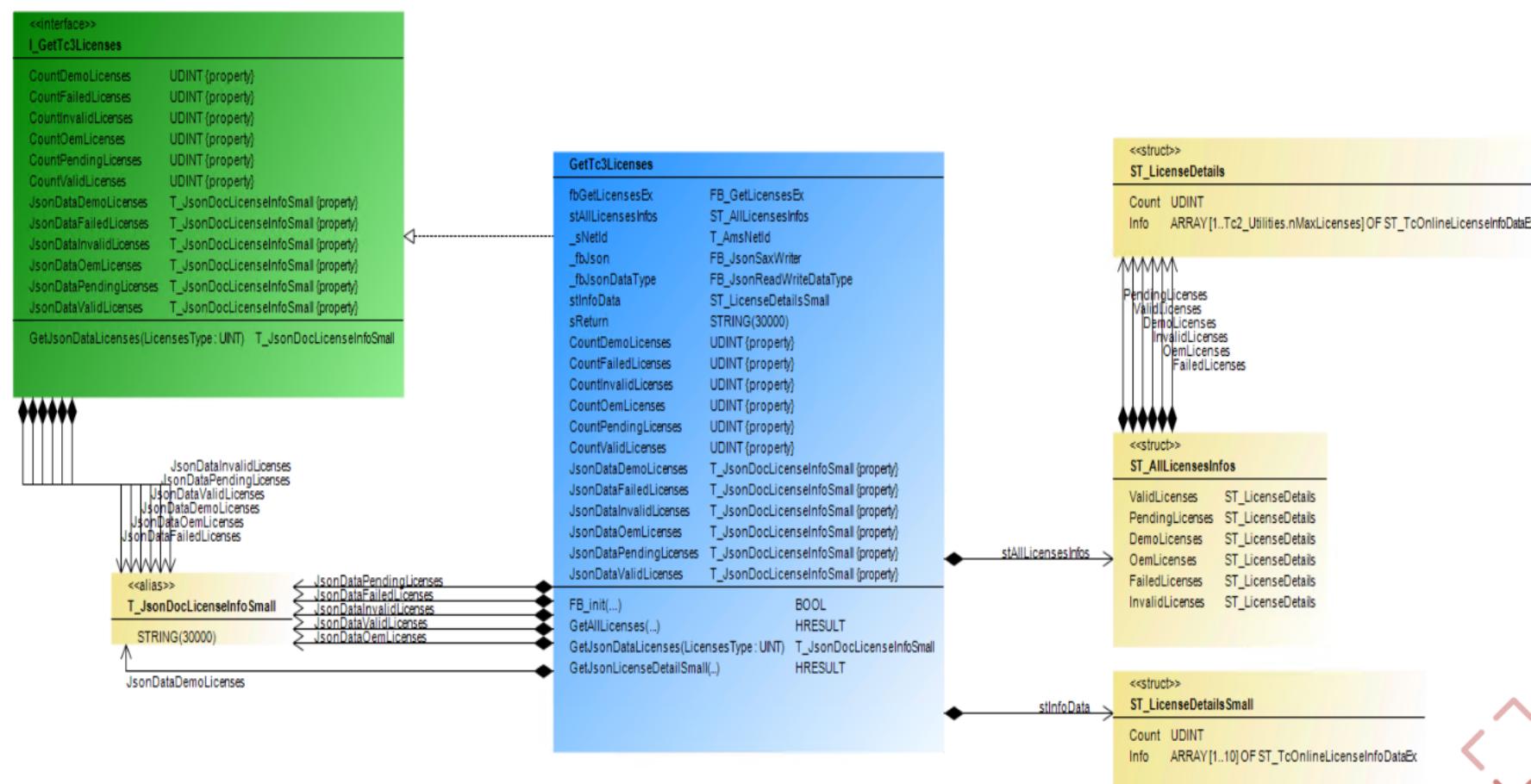
Demo Lizenzen einlesen und anzeigen

```
Demo Lizenz

Count: 7

("Count":7,"Info":[{"stLicenseId":
("Data1":2718506060,"Data2":32004,"Data3":16898,"Data4":
[177,115,153,132,231,18,91,192]),"sLicenseName":"TC3 HMI
Targets","sExpirationTime":
("wYear":2023,"wMonth":7,"wDayOfWeek":2,"wDay":25,"wHour":0,"wMinute
":0,"wSecond":0,"wMilliseconds":0),"sExpirationTime":"2023-07-25-
00:00:00.000","nMaxCount":1,"nUsedCount":0,"eResult":596,"nVolumeNo":0,
"nOptInfo":2,"nRestriction":0,"bOemLicense":false,"bBeckhoffLicense":true,
"bBeckhoffPC":true,"bEtherCATDongle":false,"bUSBxDongle":false,"bGenDev
TypeLic":false}, {"stLicenseId":
("Data1":1277519719,"Data2":59110,"Data3":19189,"Data4":
[189,104,159,122,208,194,0]),"sLicenseName":"TC3
ADS","sExpirationTime":
("wYear":2023,"wMonth":7,"wDayOfWeek":2,"wDay":25,"wHour":0,"wMinute
":0,"wSecond":0,"wMilliseconds":0),"sExpirationTime":"2023-07-25-
00:00:00.000"}, {"stLicenseId":
("Data1":1277519719,"Data2":59110,"Data3":19189,"Data4":
[189,104,159,122,208,194,0]),"sLicenseName":"TC3
HMI Targets","sExpirationTime":
("wYear":2023,"wMonth":7,"wDayOfWeek":2,"wDay":25,"wHour":0,"wMinute
":0,"wSecond":0,"wMilliseconds":0),"sExpirationTime":"2023-07-25-
00:00:00.000"}], "Count": 16}], "Count": 7

Get License Data
```



KOMPONENTEN ENTWICKLUNG TEIL 1

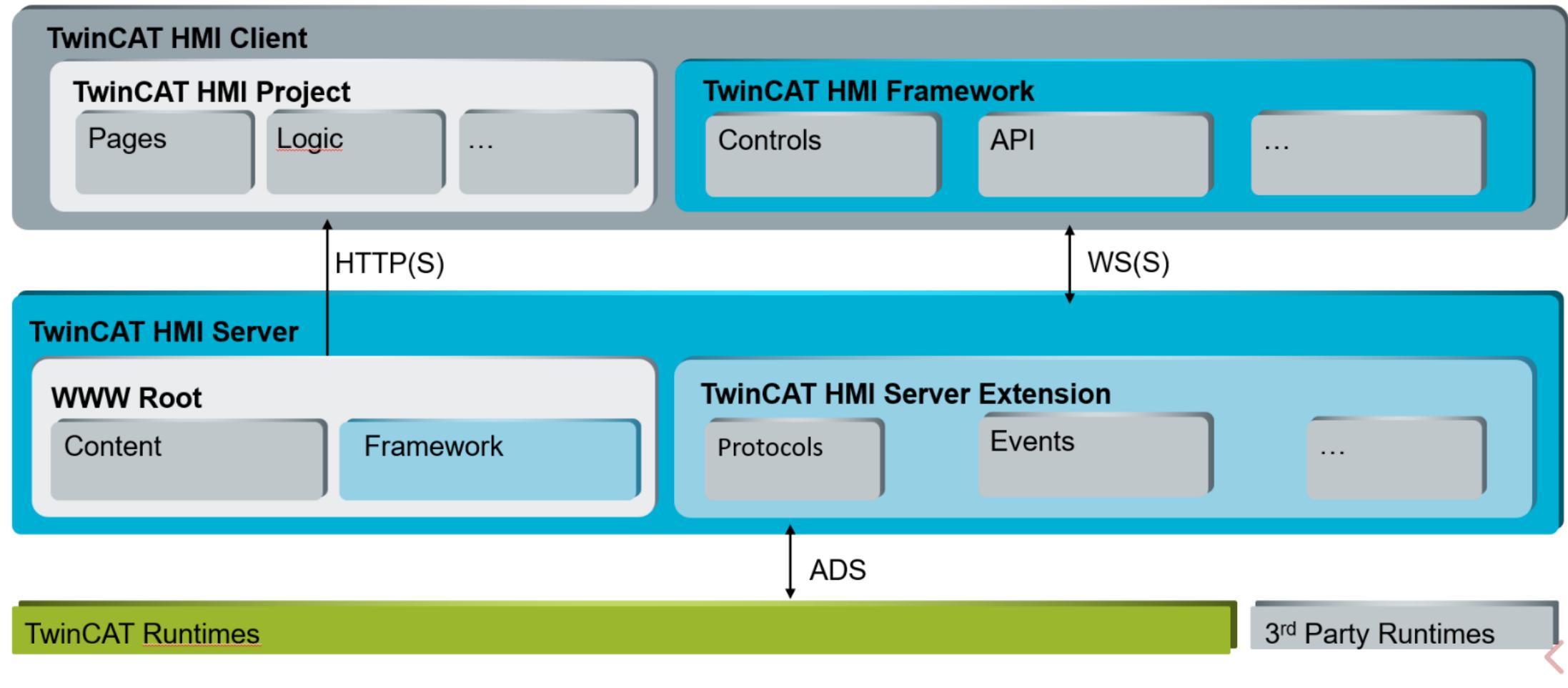
The screenshot shows the browser's developer tools element inspector. On the left, the DOM tree is displayed, showing the structure of the HTML document. On the right, the CSS styles for various elements are listed, including inheritance from parent elements and specific declarations like box-sizing, margin, and padding.

```
<body>
  <div id="app">
    <todo-application>
      #shadow-root (open)
        <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">
        <style>...</style>
        <nav class="navbar navbar-expand-md navbar-dark bg-dark">...</nav>
        <main class="container">
          <todo-form>
            <style>...</style>
            <div class="card todo-form">...</div>
          </todo-form>
          <hr>
          <todo-list ref="list">
            <style>...</style>
            <h2>Tasks:</h2>
            <ul ref="todos" class="list-group">
              <todo-task ref="task-1517176192142" id="task-1517176192142">...</todo-task>
              <todo-task ref="task-1517176320397" id="task-1517176320397">...</todo-task>
              <todo-task ref="task-1517176329096" id="task-1517176329096">...</todo-task>
              <todo-task ref="task-1517176334849" id="task-1517176334849">...</todo-task>
            </ul>
          </todo-list>
        </main>
      </todo-application>
    </div>
    <script src="...">...
```

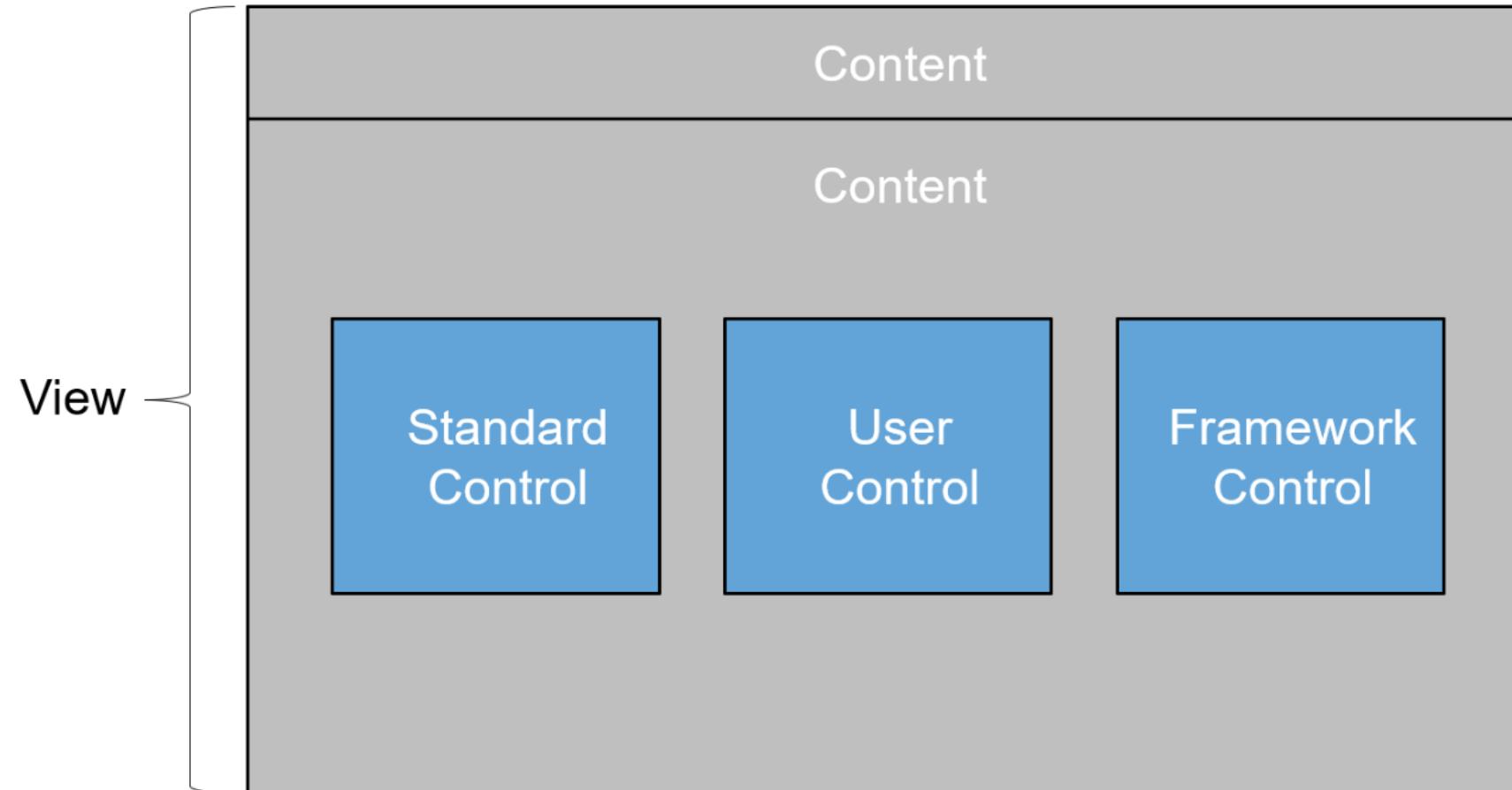
Filter :hover
element.style {
}
*, _reboot,
::after, ::before {
 box-sizing: border-box;
}
Inherited from ul.list
ul, user agent style
menu
, dir {
 display: block;
 list-style-type: none;
 -webkit-margin-before: 1em;
 -webkit-margin-after: 1em;
 -webkit-margin-start: 0px;
 -webkit-margin-end: 0px;
 -webkit-padding-start: 40px;
}
Inherited from div#app
body {
 _reboot.scss
 margin: 0;
}



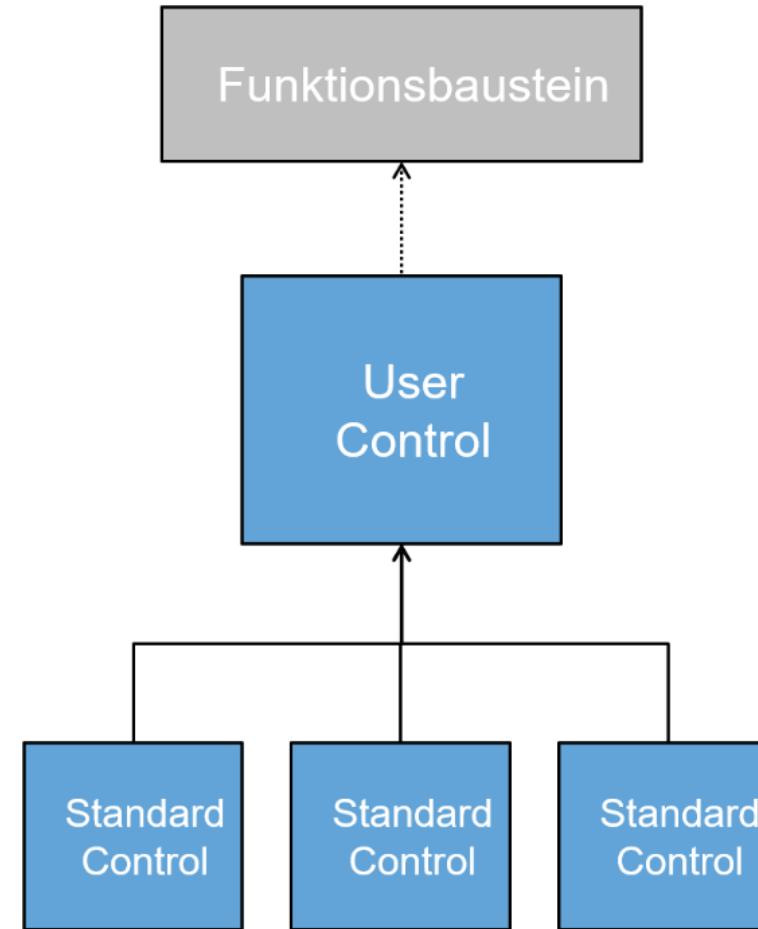
HMI UEBERBLICK



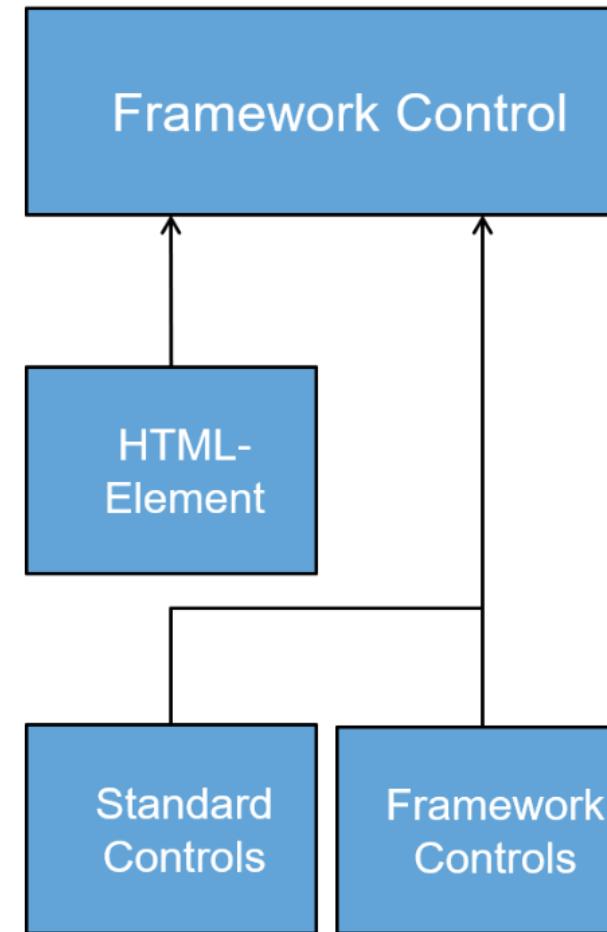
HMI PROJEKT



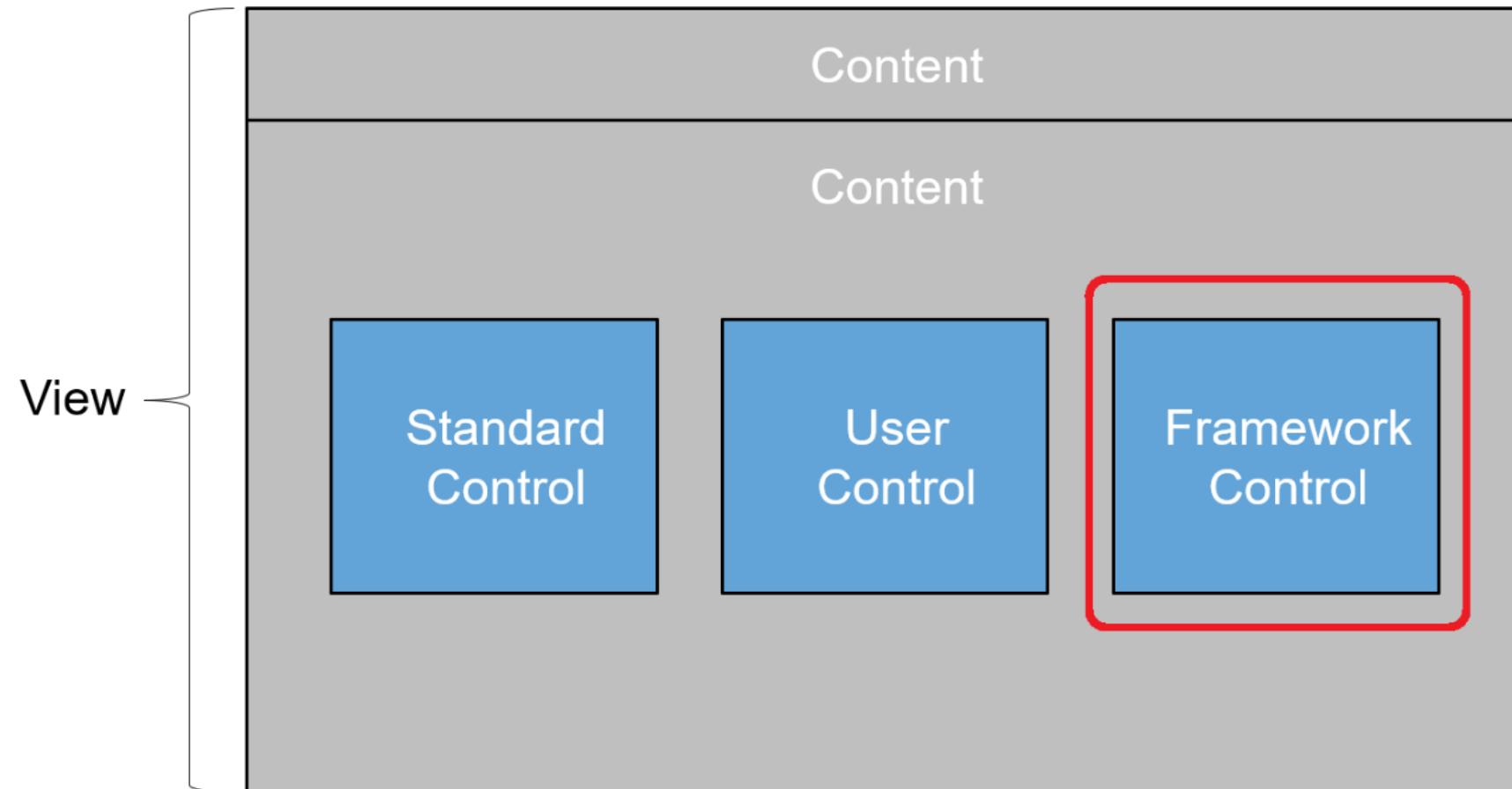
HMI USER CONTROL



HMI FRAMEWORK CONTROL



HMI FRAMEWORK PROJEKT

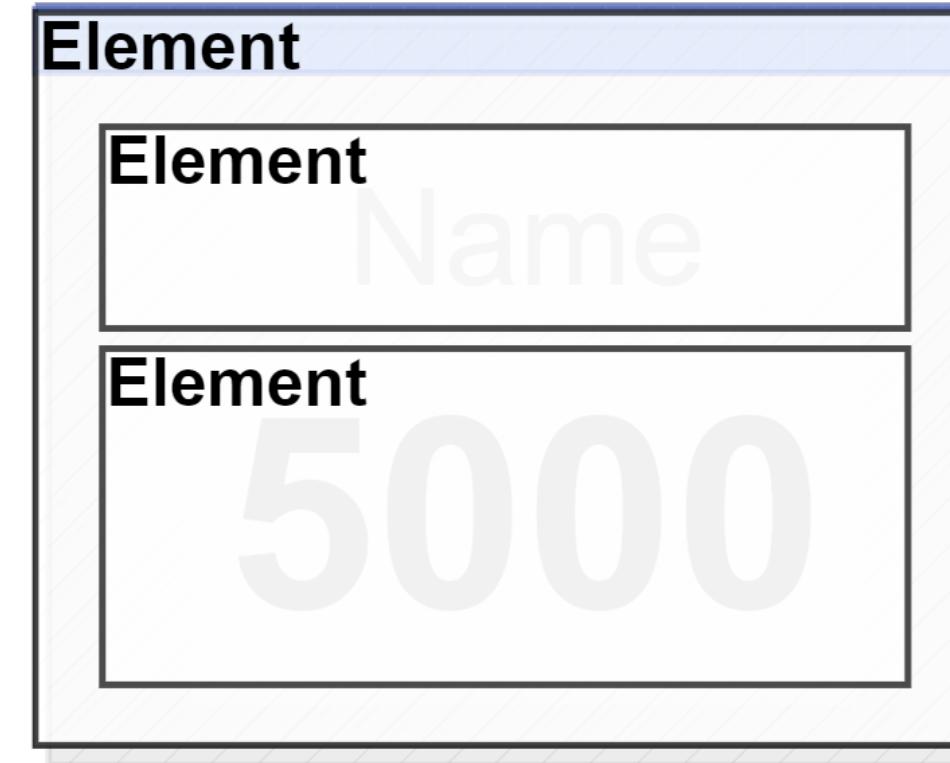


CONTROL | IDEE

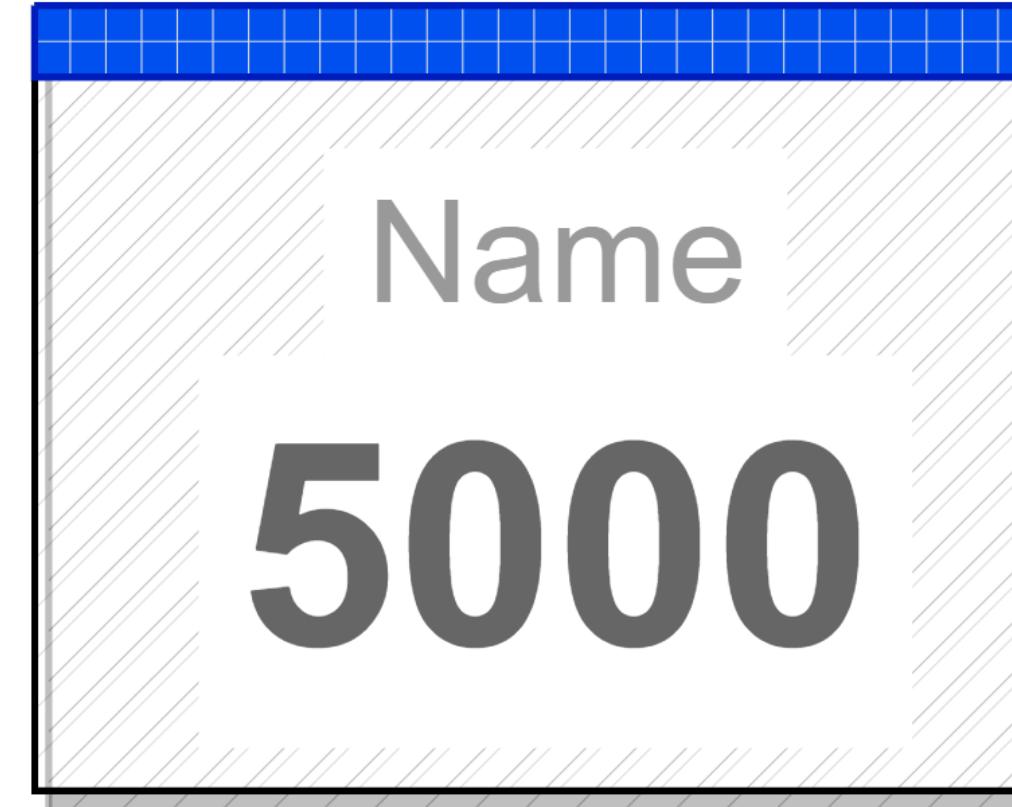


CONTROL | ANFORDERUNGEN

- Card Control
 - Ziel: Dashboard
 - Parameter: Name
 - Parameter: Wert
 - Größe 200 x 150
 - Look & Feel: Modern
 - Rechte



BEISPIEL | CONTROL



CONTROL | GRUNDGERUEST (HTML)

```
1 <div class="tchmi-control-card">
2   <div class="tchmi-control-container">
3     <div class="tchmi-control-name"></div>
4     <div class="tchmi-control-value"></div>
5   </div>
6 </div>
```

Abb: zeigt Template.html



CONTROL | SCHNITTSTELLE (JSON)

Abb: zeigt Description.json

The screenshot shows a software interface for managing control descriptions. On the left, there is a code editor window titled "Description.json" with the schema "./.hmiframework/Schema/ControlDescription.Schema.json". The JSON code is displayed with line numbers from 64 to 101. Two specific sections of the JSON are highlighted with colored boxes: a green box around the "data-tchmi-name" attribute and a blue box around the "data-tchmi-value" attribute. On the right, there is a "Properties" panel for an object named "CardClassic_1". The "Type" is listed as "TcHmi.Controls.HmiUpdateSi2023Classic.CardClassic". The "Data" section contains two entries: "Name" with value "Name" and "Value" with value "9999.0". The "Layout" section shows dimensions: Left (180), Top (339), Width (200), and Height (150). A red decorative icon is located in the bottom right corner of the slide.

```
64      },
65      "attributes": [
66      {
67          "name": "data-tchmi-name",
68          "propertyName": "Name",
69          "propertySetterName": "setName",
70          "propertyGetterName": "getName",
71          "displayName": "Name",
72          "visible": true,
73          "themeable": "Standard",
74          "displayPriority": 10,
75          "type": "tchmi:general#/definitions/String",
76          "category": "Data",
77          "description": "",
78          "readOnly": false,
79          "bindable": true,
80          "heritable": true,
81          "defaultValue": "Name",
82          "defaultValueInternal": "Name"
83      },
84      {
85          "name": "data-tchmi-value",
86          "propertyName": "Value",
87          "propertySetterName": "setValue",
88          "propertyGetterName": "getValue",
89          "displayName": "Value",
90          "visible": true,
91          "themeable": "Standard",
92          "displayPriority": 10,
93          "type": "tchmi:general#/definitions/String",
94          "category": "Data",
95          "description": "",
96          "readOnly": false,
97          "bindable": true,
98          "heritable": true,
99          "defaultValue": "9999.0",
100         "defaultValueInternal": "9999.0"
101     }
```

Properties

- Identifier: CardClassic_1
- Type: TcHmi.Controls.HmiUpdateSi2023Classic.CardClassic

Search Properties

Arrange by: Category ▾

- Colors
- Data
 - Name: Name
 - Value: 9999.0
- Layout
 - Left: 180
 - Top: 339
 - Right:
 - Bottom:
 - Width: 200
 - Height: 150
- Common
- Border
- Background Image

CONTROL | LOGIK I (JAVASCRIPT)

Definition

```
1 constructor(element, pcElement, attrs) {  
2     ...  
3     this.__name = undefined;  
4     this.__value = undefined;  
5 }
```

Abb: zeigt *Control*.js

Selektoren

```
1 __previnit() {  
2     ...  
3     /** element selector */  
4     this.__elementName = this.__elementTemplateRoot.find('.tchmi-control-name').first();  
5     this.__elementValue = this.__elementTemplateRoot.find('.tchmi-control-value').first();  
6     ...  
7 }
```

Abb: zeigt *Control*.js



CONTROL | LOGIK II (JAVASCRIPT)

Setter & Getter Funktion

```
1  getName() {
2      return this.__name;
3  }
4
5  setName(valueNew) {
6      var convertedValue = TcHmi.ValueConverter.toString(valueNew);
7      var convertedValue = valueNew;
8      if (convertedValue === null) {
9          convertedValue = this.getAttributeDefaultValueInternal('Name');
10     }
11     if (tchmi_equal(convertedValue, this.__name)) return;
12     this.__name = convertedValue;
13     TcHmi.EventProvider.raise(this.__id + '.onPropertyChanged', { propertyName: 'Name' });
14     this.__processName();
15 }
16
17 __processName() {
18     let name = this.__name;
19     if (name === null) return;
20     this.__elementName[0].innerText = name;
21 }
```

Abb: zeigt *Control*.js



CONTROL | STYLING (CSS)

```
1 .tchmi-control-card {  
2     width: 100%;  
3     height: 100%;  
4     border: .5px solid rgb(231, 231, 231);  
5     border-radius: 10px;  
6     ...  
7 }  
8 .tchmi-control-container {  
9     position: relative;  
10    height: 100%;  
11    ...  
12 }  
13 .tchmi-control-name {  
14     color: #6c6c6c;  
15     font-size: 2.1em;  
16     ...  
17 }  
18 .tchmi-control-value {  
19     font-weight: 600;  
20     font-size: 3.6em;  
21     color: #323c43;  
22     ...  
23 }
```



KOMPONENTEN ENTWICKLUNG TEIL 2

The screenshot shows the browser's developer tools element inspector. On the left, the DOM tree is displayed with various components like `<body>`, `<div id="app">`, and `<todo-application>`. On the right, the CSS styles for a specific element are shown, including inheritance from `ul.list` and `div#app`, and a detailed breakdown of properties like `border`, `list-style-type`, and `margin`.

```
<body>
  <div id="app">
    <todo-application>
      #shadow-root (open)
        <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">
        <style>...</style>
        <nav class="navbar navbar-expand-md navbar-dark bg-dark">...</nav>
        <main class="container">
          <todo-form>
            <style>...</style>
            <div class="card todo-form">...</div>
          </todo-form>
          <hr>
          <todo-list ref="list">
            <style>...</style>
            <h2>Tasks:</h2>
            <ul ref="todos" class="list-group">
              <todo-task ref="task-1517176192142" id="task-1517176192142">...</todo-task>
              <todo-task ref="task-1517176320397" id="task-1517176320397">...</todo-task>
              <todo-task ref="task-1517176329096" id="task-1517176329096">...</todo-task>
              <todo-task ref="task-1517176334849" id="task-1517176334849">...</todo-task>
            </ul>
          </todo-list>
        </main>
      </todo-application>
    </div>
    <script src="...">...
```

Filter :hover
element.style {
}
*, _reboot,
::after, ::before {
 box-sizing: border
 box;
}
Inherited from ul.list
ul, user agent style
menu
, dir {
 display: block;
 list-style-type: none;
 -webkit-margin-before: 1em;
 -webkit-margin-after: 1em;
 -webkit-margin-start: 0px;
 -webkit-margin-end: 0px;
 -webkit-padding-start: 40px;
}
Inherited from div#app
body {
 _reboot.scss
 margin: 0;
}



UEBERBLICK FRONTEND FRAMEWORK

- React
 - Schnelle Updates
 - Unterstützt von Facebook
 - Kompatibel mit vielen JS-Bibliotheken
- Angular
 - Zwei-Wege-Datenbindung
 - Komponentenbasierte Architektur
 - Unterstützt von Google
- Vue
 - Schnell und klein
 - Anfängerfreundlich
 - Zwei-Wege-Datenbindung



VUE | FRAMEWORK

- Dokumentation
 - Framework: <https://vuejs.org>
 - Paketquellen: <https://vuejs.org/guide/quick-start.html#using-vue-from-cdn>
- Paketquellen: <https://unpkg.com/browse/vue@3.3.4/dist/>



VUE | INTEGRATION IN TWINCAT HMI

- neues HMI-Framework Projekt erstellen
 - Ordner erstellen Libaries
 - Datei erstellen vue.min.js
 - Inhalt des Downloads speichern: <https://unpkg.com/vue@3.3.4/dist/vue.global.prod.js>
 - Manifest.json erweitern

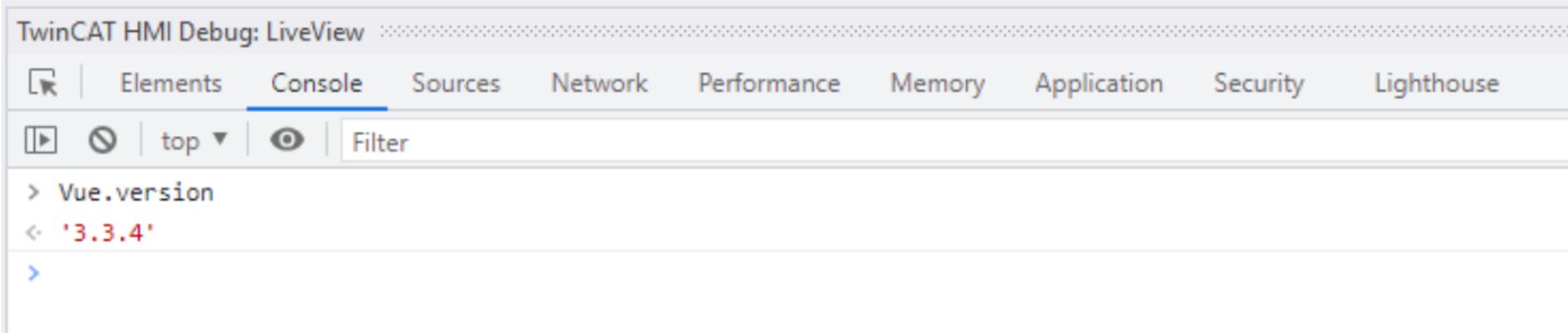
```
1  {
2    ...
3    {
4      "type": "Resource",
5      "path": "./Libaries/vue.min.js"
6    },
7    ...
8  ],
9  "provideMetadata": {
10   ...
11 },
12 "$schema": ".hmiframework/Schema/Manifest.Schema.json",
13 "apiVersion": 1
14 }
```

Abb: zeigt Manifest.json



VUE | INTEGRATIONS TEST

- Live-View Debugger
 - Console öffnen



The screenshot shows the 'Console' tab of the TwinCAT HMI Debug: LiveView interface. The tab bar includes Elements, Console (which is selected), Sources, Network, Performance, Memory, Application, Security, and Lighthouse. Below the tab bar are controls for play/pause, stop, and filter, along with a 'Filter' input field. The main area displays the output of a command: '> Vue.version' followed by '<' and the value '3.3.4' in red text, indicating it's a constant or environment variable.

```
> Vue.version
< '3.3.4'
```

Abb: zeigt Debugger Console

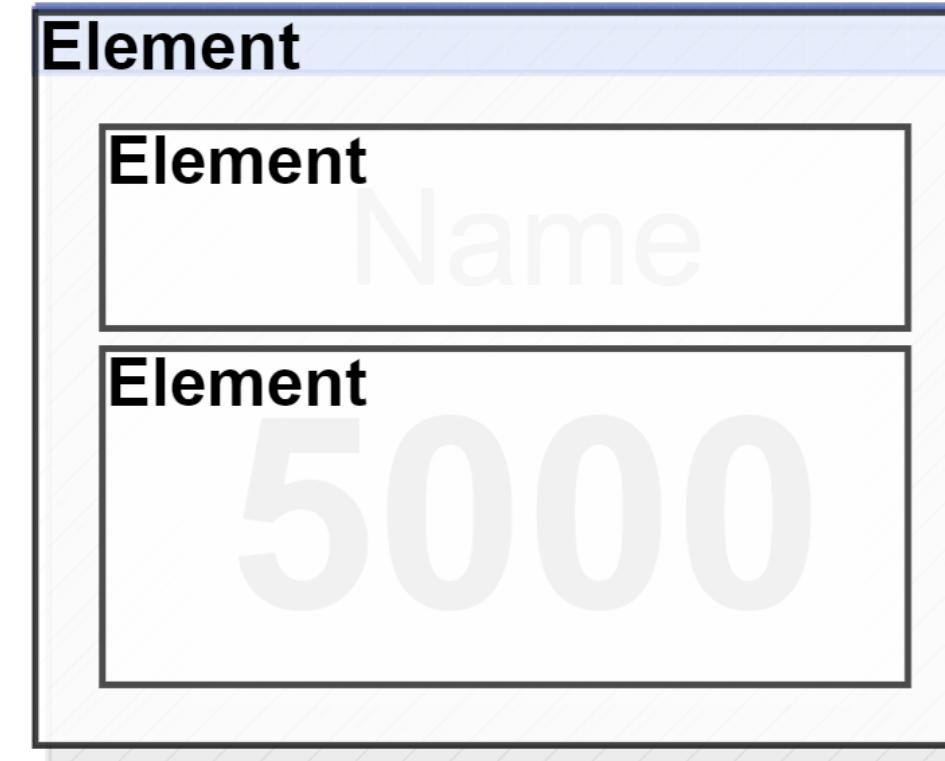


CONTROL | IDEE

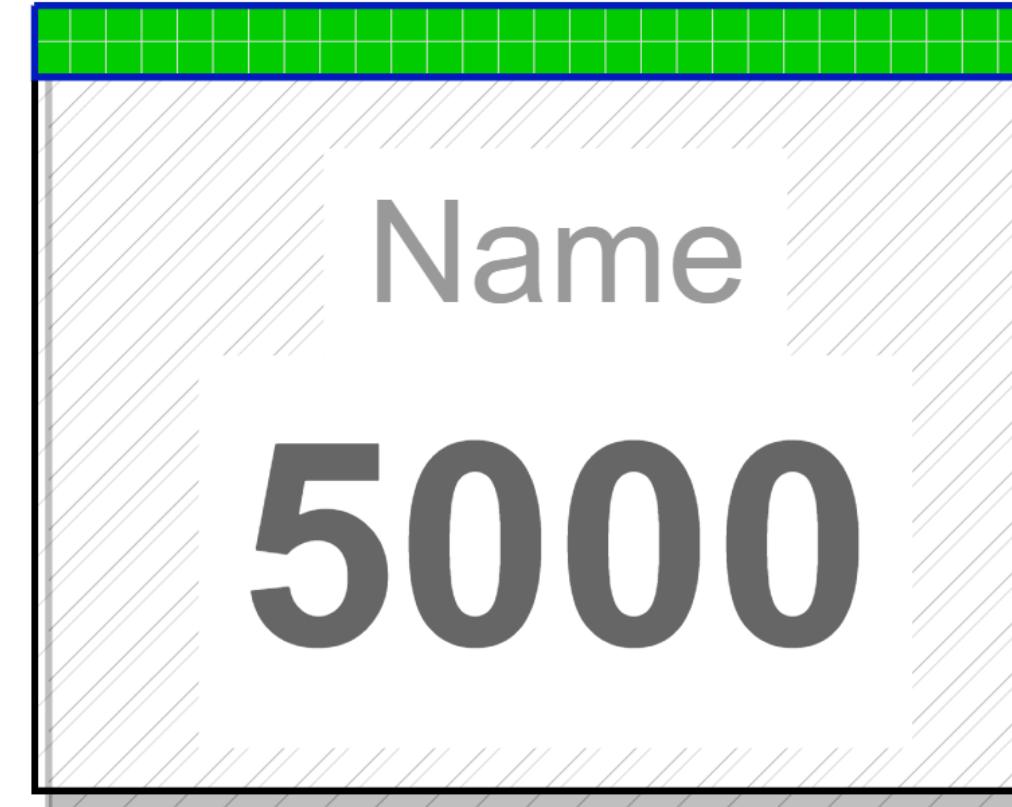


CONTROL | ANFORDERUNGEN

- Card Control
 - Ziel: Dashboard
 - Parameter: Name
 - Parameter: Wert
 - Größe 200 x 150
 - Look & Feel: Modern
 - Rechte



BEISPIEL | CONTROL



CONTROL | GRUNDGERUEST (HTML)

```
1 <div class="tchmi-control-card-vue">
2   <div class="tchmi-control-container-vue">
3     <div class="tchmi-control-name-vue">{{name}}</div>
4     <div class="tchmi-control-value-vue">{{value}}</div>
5   </div>
6 </div>
```

Abb: zeigt Template.html



CONTROL | SCHNITTSTELLE (JSON)

Abb: zeigt Description.json

The screenshot shows a software interface for managing control descriptions. On the left, there is a code editor window titled "Description.json" with the schema "./.hmiframework/Schema/ControlDescription.Schema.json". The JSON code is displayed with line numbers from 64 to 101. Two specific sections of the JSON are highlighted with colored boxes: a green box around the "data-tchmi-name" attribute and a blue box around the "data-tchmi-value" attribute.

The right side of the interface shows the "Properties" panel for a component named "CardClassic_1". The "Type" is listed as "TcHmi.Controls.HmiUpdateSi2023Classic.CardClassic". The "Data" section contains two entries: "Name" with value "Name" and "Value" with value "9999.0". The "Layout" section shows dimensions: Left (180), Top (339), Width (200), and Height (150). Other sections like "Common" and "Border" are also visible.



CONTROL | LOGIK I (JAVASCRIPT)

Definition

```
1 constructor(element, pcElement, attrs) {  
2     ...  
3     this.__name = Vue.ref();  
4     this.__value = Vue.ref();  
5     /** vue instance*/  
6     this.__app = undefined;  
7 }
```

Initialisierung

```
1 __attach() {  
2     ...  
3     let _this = this  
4     //create new vue instance  
5     this.__app = Vue.createApp({  
6         data() {  
7             return {  
8                 name: _this.__name,  
9                 value: _this.__value  
10            }  
11        },  
12    }).mount('#' + _this.__id);  
13 }
```



CONTROL | LOGIK III (JAVASCRIPT)

Setter & Getter Funktion

```
1  getName() {  
2      return this.__name;  
3  }  
4  
5  setName(valueNew) {  
6      var convertedValue = TcHmi.ValueConverter.toString(valueNew);  
7      var convertedValue = valueNew;  
8  
9      if (convertedValue === null) {  
10         convertedValue = this.getAttributeDefaultValueInternal('Name');  
11     }  
12  
13     if (tchmi_equal(convertedValue, this.__name.value)) {  
14         return;  
15     }  
16  
17     this.__name.value = convertedValue;  
18     TcHmi.EventProvider.raise(this.__id + '.onPropertyChanged', { propertyName: 'Name' });  
19  }  
20 }
```

Abb: zeigt *Control*.js



CONTROL | STYLING (CSS)

```
1 .tchmi-control-card-vue {  
2     width: 100%;  
3     height: 100%;  
4     border: .5px solid rgb(231, 231, 231);  
5     border-radius: 10px;  
6     ...  
7 }  
8 .tchmi-control-container-vue {  
9     position: relative;  
10    height: 100%;  
11    ...  
12 }  
13 .tchmi-control-name-vue {  
14     color: #6c6c6c;  
15     font-size: 2.1em;  
16     ...  
17 }  
18 .tchmi-control-value-vue {  
19     font-weight: 600;  
20     font-size: 3.6em;  
21     color: #323c43;  
22     ...  
23 }
```

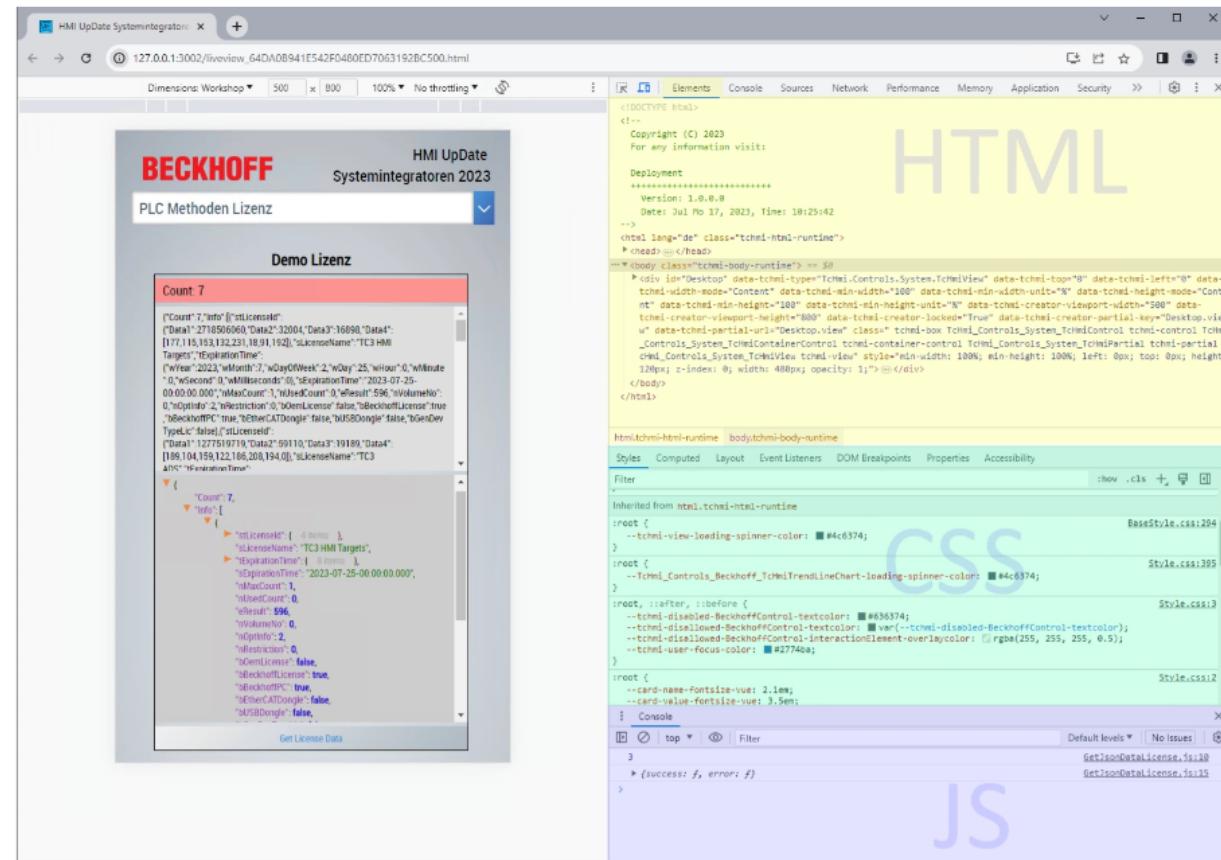
Abb: zeigt Style.css



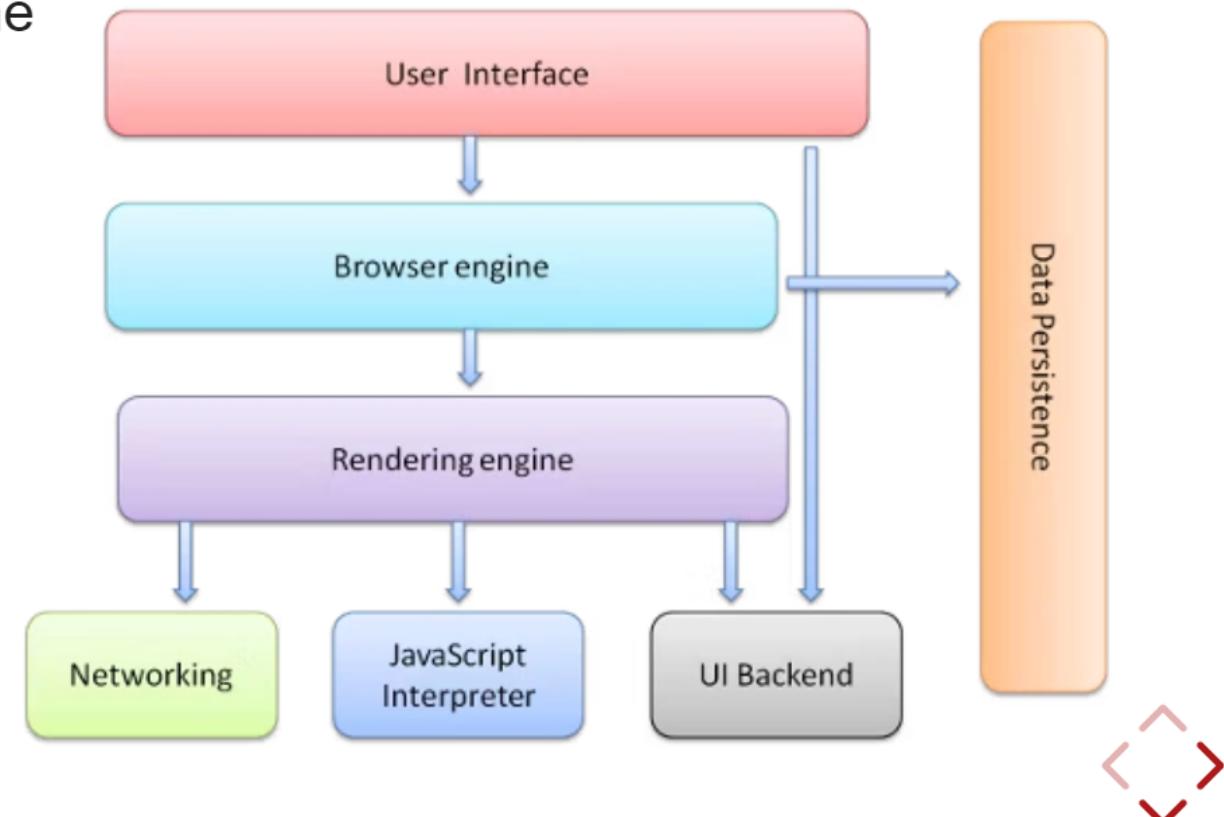
DEVELOPER TOOLS

Mit den Entwicklertools (DevTools) von Chrome können Entwickler

- das Layout von Websites anpassen
- Screenshots erzeugen
- Responsive Design testen



- User Interface
 - Adressleiste
 - Zurück/Vorwärts
- Browser Engine
 - Verbindung Oberfläche / Rendering Engine
- Rendering Engine
 - Anzeige der HTML, CSS Inhalte
- Networking
 - Bearbeitung der Netzwerkzugriffe
- UI-Backend
 - Plattformspezifisch Schnittstelle
- JavaScript interpreter
 - Parsen von JavaScript Code
- Data storage
 - Speichermechanismen (Cookies, ect.)



ENTWICKLERTOOLS ÖFFNEN

Die Tools können über verschiedene Wege geöffnet werden

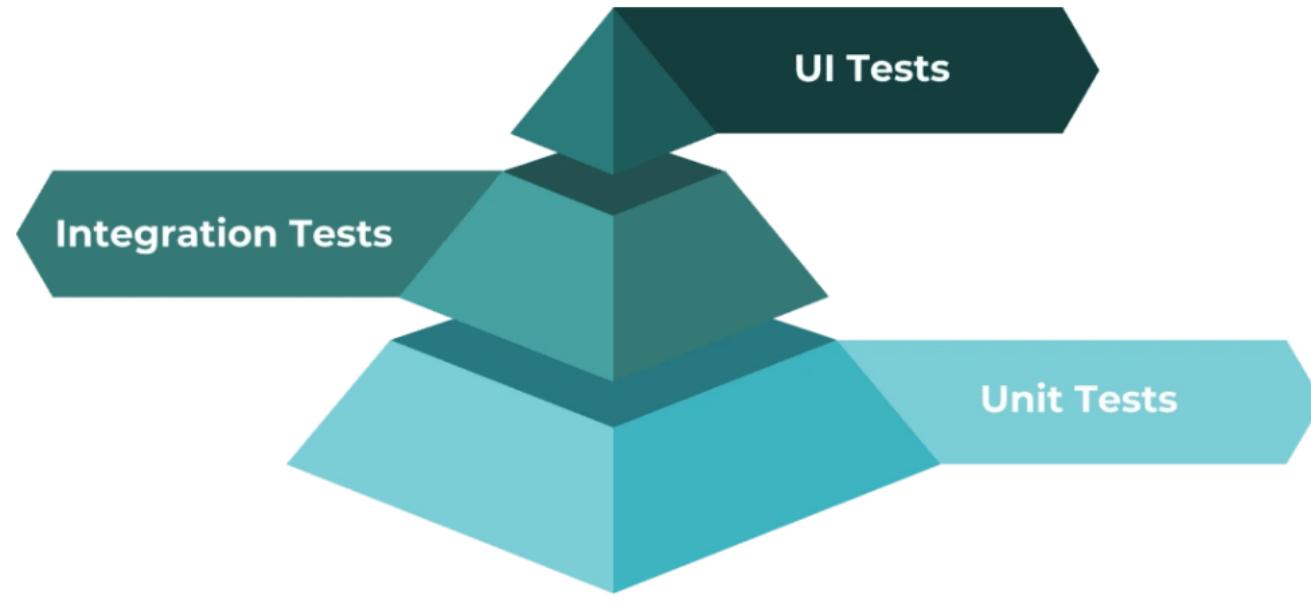
- Maus Rechtsklick -> Context -> Untersuchen
- F12
- Strg+Shift+I



TESTPYRAMIDE

Es gibt 3 Haupt Test Ebenen

- Unit Test
- Integrations Test
- UI Test



UI TEST

UI Test werden auch als E2E (End to End) Tests bezeichnet

FRAMEWORKS

Die bekanntesten Frameworks

- Selenium (Fa. ThoughtWorks)
- Puppeteer (Google)
- Cypress (cypress.io)
- Playwright (Microsoft)



PLAYWRIGHT

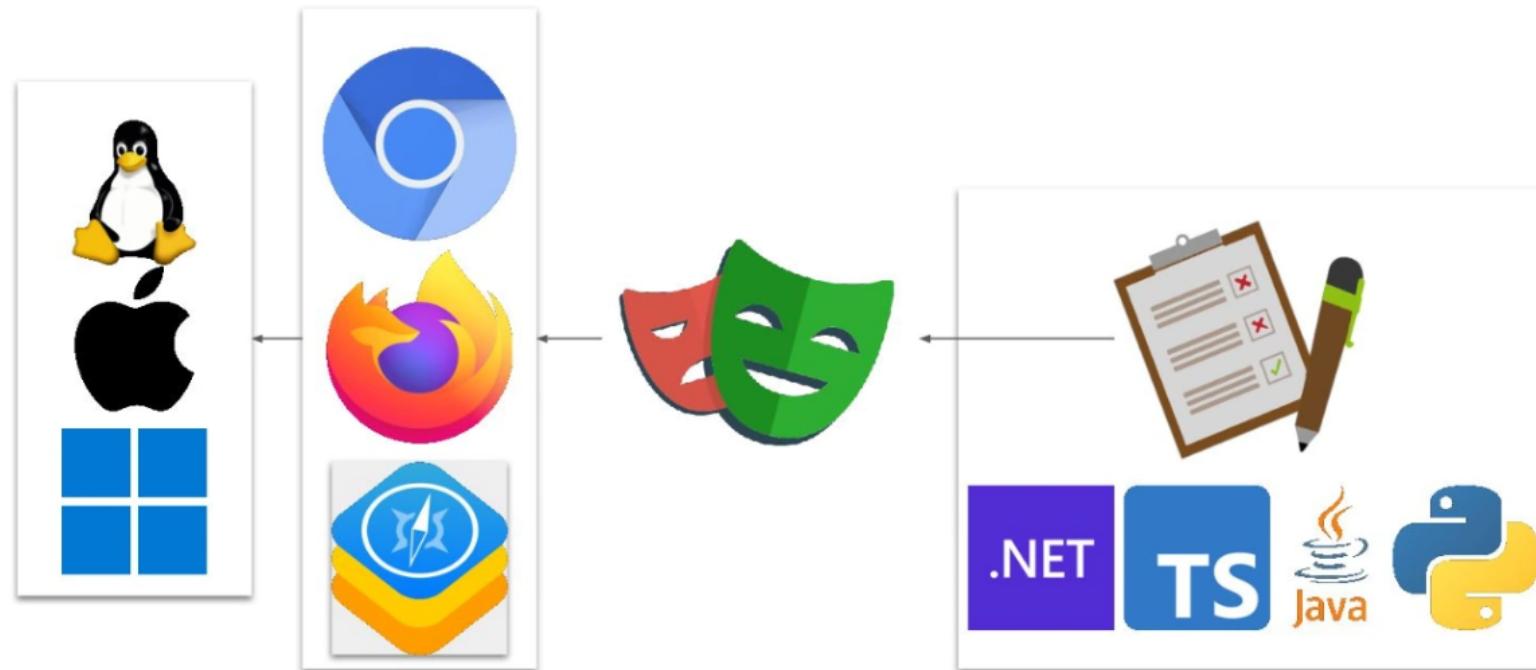
End to End Test Framework

Unterstützte Sprachen

- Java,
- .NET,
- Python,
- JavaScript und
- TypeScript



ARCHITEKTUR



SYSTEMVORRAUSETZUNG

- Node.js 16+
- Windows 10+, Windows Server 2016+ oder Windows Subsystem for Linux (WSL).
- MacOS 12 Monterey or MacOS 13 Ventura.
- Debian 11, Ubuntu 20.04 or Ubuntu 22.04.



WORKFLOW

- Installation

```
1 npm init playwright@latest
```

- Test starten

```
1 npx playwright test
```

- Test starten mit Oberfläche

```
1 npx playwright test --ui
```

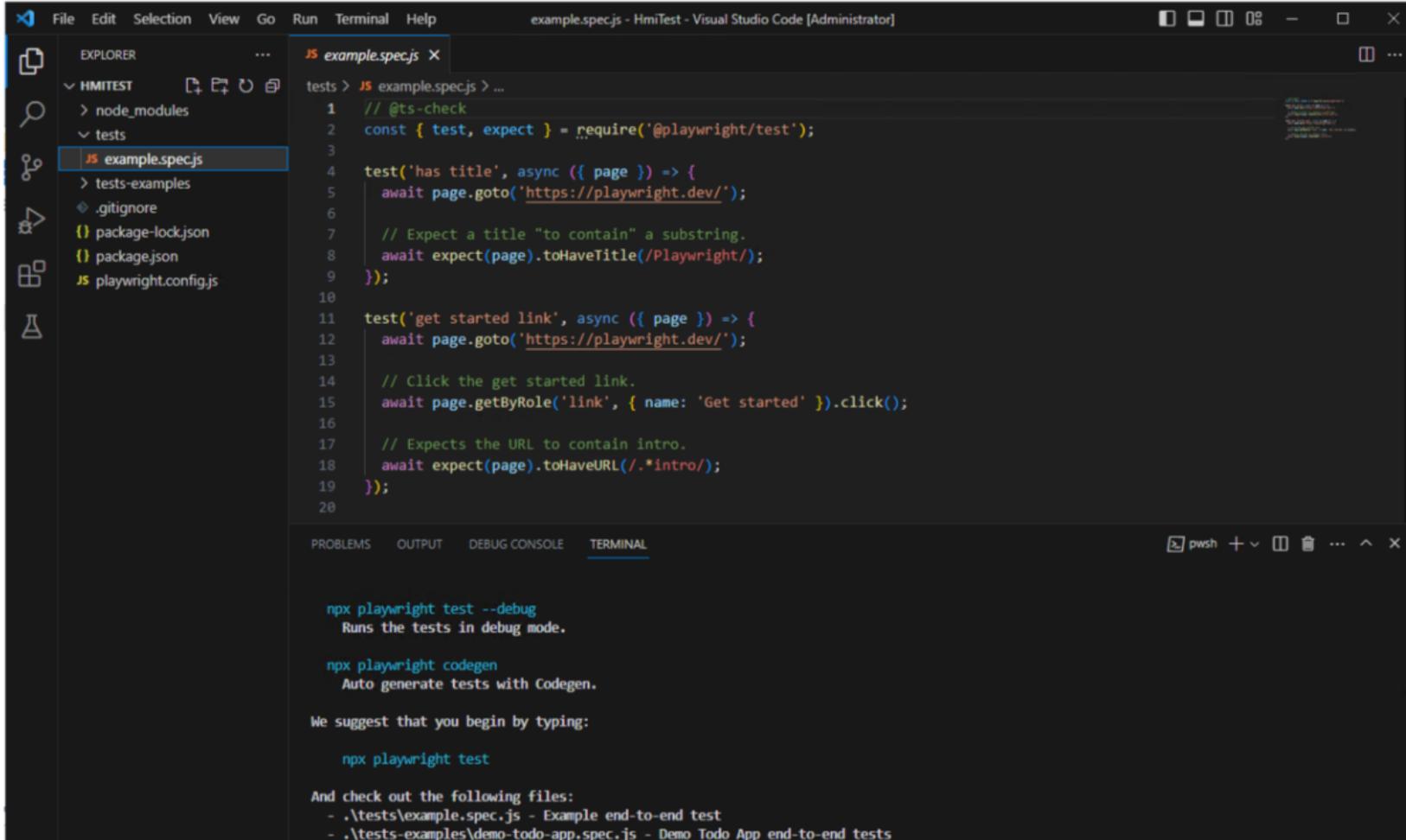
- HTML Report

```
1  
2 npx playwright show-report
```



ERSTELLEN

Playwright ist in Visual Studio Code (VSCode) integriert



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a project structure with a folder named "HMITEST" containing "node_modules", "tests", and files like ".gitignore", "package-lock.json", "package.json", and "playwright.config.js". The file "example.spec.js" is currently selected.
- Code Editor (Center):** Displays the content of "example.spec.js":

```
// @ts-check
const { test, expect } = require('@playwright/test');

test('has title', async ({ page }) => {
  await page.goto('https://playwright.dev/');
  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/Playwright/);
});

test('get started link', async ({ page }) => {
  await page.goto('https://playwright.dev/');
  // Click the get started link.
  await page.getByRole('link', { name: 'Get started' }).click();
  // Expects the URL to contain intro.
  await expect(page).toHaveURL(/.*intro/);
});
```
- Terminal (Bottom):** Shows command-line instructions:

```
npx playwright test --debug
  Runs the tests in debug mode.

npx playwright codegen
  Auto generate tests with Codegen.

We suggest that you begin by typing:
  npx playwright test

And check out the following files:
  - ./tests/example.spec.js - Example end-to-end test
  - ./tests-examples/demo-todo-app.spec.js - Demo Todo App end-to-end tests
```



AUSFÜHRUNG VON TEST'S

Im Visual Studio Code

The screenshot shows a Visual Studio Code interface with a browser window and a code editor.

Browser Window: The title bar says "HMI UpDate Systemintegratoren". The address bar shows "127.0.0.1:3002/liveview_64DA0B941E542F0480E". The content of the browser shows two examples of PLC Methoden Addition:

- Top example: Input fields show 33 and 10, result is 43. Below it is a button labeled "AdditionWithProperties()".
- Bottom example: Input fields show 0 and 0, result is 43. Below it is a button labeled "AdditionValue()".

Code Editor: The file "PlcMethodenAddition.spec.ts" is open. It contains a test script using the playwright library to interact with the HMI application:

```
import { test, expect } from '@playwright/test';

test('AdditionWithProperties', async ({ page }) => {
    await page.goto('http://127.0.0.1:3002/liveview_64DA0B941E542F0480ED7063192BC500.html');
    await page.waitForTimeout(1000); - 100ms
    await page.getByRole('img').click(); - 82ms
    await page.getText('PLC Methoden Addition').click(); - 63ms
    await page.waitForTimeout(1000); - 1003ms
    await page.getByPlaceholder('Wert').first().click(); - 48ms
    await page.getByPlaceholder('Wert').first().dblclick(); - 61ms
    await page.getByPlaceholder('Wert').first().press('ArrowLeft'); - 10ms
    await page.getByPlaceholder('Wert').first().press('Shift+ArrowRight'); - 8ms
    await page.getByPlaceholder('Wert').first().press('Shift+ArrowRight'); - 7ms
    await page.getByPlaceholder('Wert').first().fill('33'); - 14ms
    await page.waitForTimeout(1000); - 1003ms
    await page.getByPlaceholder('Wert').nth(1).click(); - 45ms
    await page.getByPlaceholder('Wert').nth(1).dblclick(); - 62ms
    await page.getByPlaceholder('Wert').nth(1).fill('3'); - 9ms
    await page.getByPlaceholder('Wert').nth(1).press('End'); - 6ms
    await page.getByPlaceholder('Wert').nth(1).fill('10'); - 11ms
    await page.getByPlaceholder('Wert').nth(1).press('Enter'); - 8ms
    await page.waitForTimeout(2000); - 2004ms
    await page.locator('#bt01MethodAdd div').click(); - 43ms
});
```

VS Code UI: The sidebar shows "TEST EXPLORER" with "1/1 tests passed (100%)". The bottom left has a "PLAYWRIGHT" section with checkboxes for "Show browser" (checked), "Show trace viewer", "Pick locator", "Record new", and "Record at cursor". The bottom right shows a terminal window with the command "npx playwright test --ui" and a warning about deprecated configuration options.

REPORT VON TEST'S

HMI UpDate Systemintegrator x Playwright Test Report x +

localhost:9323/#?testId=ec76f4271cd9987bc91e-ec9588a258f41a8a53c1

All 1 Passed 1 Failed 0 Flaky 0 Skipped 0

AdditionWithProperties

PlcMethodenAddition.spec.ts:3

chromium

Run

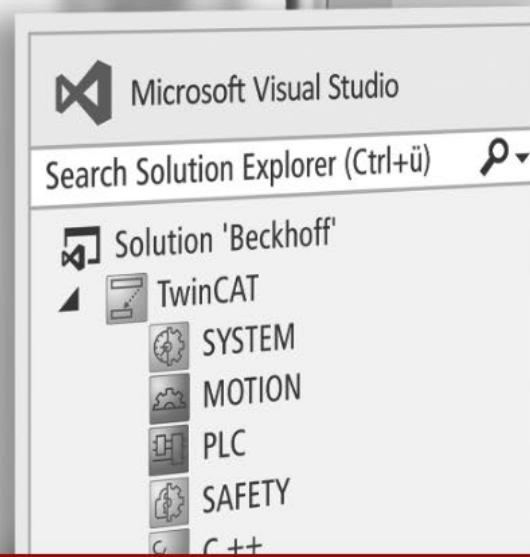
Test Steps

- > ✓ Before Hooks 417ms
- < ✓ page.goto('http://127.0.0.1:3002/liveview_64DA0B941E542F0480ED7063192BC500.html') — PlcMethodenAddition.spec.ts:4 316ms
 - 3 | test('AdditionwithProperties', async ({ page }) => {
 - 4 | await page.goto('http://127.0.0.1:3002/liveview_64DA0B941E542F0480ED7063192BC500.html');
 - 5 | await page.waitForTimeout(1000);
- > ✓ page.waitForTimeout — PlcMethodenAddition.spec.ts:5 1.0s
- > ✓ locator.getByRole('img').click — PlcMethodenAddition.spec.ts:6 82ms
- > ✓ locator.getText('PLC Methoden Addition').click — PlcMethodenAddition.spec.ts:7 63ms
- > ✓ page.waitForTimeout — PlcMethodenAddition.spec.ts:8 1.0s
- > ✓ locator.getByPlaceholder('Wert').first().click — PlcMethodenAddition.spec.ts:9 48ms
- > ✓ locator.getByPlaceholder('Wert').first().dblclick — PlcMethodenAddition.spec.ts:10 61ms
- > ✓ locator.getByPlaceholder('Wert').first().press(ArrowLeft) — PlcMethodenAddition.spec.ts:11 10ms
- > ✓ locator.getByPlaceholder('Wert').first().press(Shift+ArrowRight) — PlcMethodenAddition.spec.ts:12 8ms
- > ✓ locator.getByPlaceholder('Wert').first().press(Shift+ArrowRight) — PlcMethodenAddition.spec.ts:13 7ms
- > ✓ locator.getByPlaceholder('Wert').first().fill — PlcMethodenAddition.spec.ts:14 14ms
- > ✓ page.waitForTimeout — PlcMethodenAddition.spec.ts:15 1.0s
- > ✓ locator.getByPlaceholder('Wert').nth(1).click — PlcMethodenAddition.spec.ts:16 45ms
- > ✓ locator.getByPlaceholder('Wert').nth(1).dblclick — PlcMethodenAddition.spec.ts:17 62ms



HMI UpDate Systemintegratoren 2023

Danke für Ihre Teilnahme



Feedback

