

Dynamische Formulare mit JSON-Schema und Ajv

Web-Formulare auf Basis von JSON-Schema und Ajv generieren und validieren lassen.

06.08.2023



Vorstellung



Christian Klihm

Frontend Developer und Ausbilder

c.klihm@constructiva.de



Constructiva Solutions GmbH

Irmintrudisstraße 13

53111 Bonn

Herausforderung

Dynamische Formulare auf beliebigen Landingpages mit einem Formular-Widget ermöglichen.

Formulare in der Realität

RECHNUNGSADRESSE

Die gekennzeichneten Felder(*) sind Pflichtfelder.

ICH MELDE MICH AN ALS

Privatkunde Geschäftskunde

VORNAME *

NACHNAME *

Bitte geben Sie einen Vornamen an.

STRASSE & HAUSNUMMER *

PLZ *

ORT *

LAND *

TELEFON

E-MAIL *

Kundenkonto anlegen 

Weiter

[Als Firmenkunde anmelden.](#)

Bitte beachten Sie, dass eine nachträgliche Änderung der Rechnung von privat auf geschäftlich nicht möglich ist.

Formulare in der Realität

```
<label  
  ... for="address-form:firstname"  
  ... class="form-label"  
>Vorname<span class="the-required-asterisk">*</span></label>  
<input  
  ... id="address-form:firstname"  
  ... type="text"  
  ... name="invoice-address-form:firstname"  
  ... class="form-control"  
  ... maxlength="20"  
  ... required  
>  
<span id="address-form:firstname-error"></span>
```

RECHNUNGSADRESSE

Die gekennzeichneten Felder(*) sind Pflichtfelder.

ICH MELDE MICH AN ALS

Privatkunde Geschäftskunde

VORNAME *

Bitte geben Sie einen Vornamen an.

NACHNAME *

STRASSE & HAUSNUMMER *

PLZ *

ORT *

LAND *

TELEFON

E-MAIL *

Kundenkonto anlegen ▾

Weiter

[Als Firmenkunde anmelden.](#)

Bitte beachten Sie, dass eine nachträgliche Änderung der Rechnung von privat auf geschäftlich nicht möglich ist.

Formulare in der Realität

```
<label  
  ... for="address-form:firstname"  
  ... class="form-label"  
>Vorname<span class="the-required-asterisk">*</span></label>  
<input  
  ... id="address-form:firstname"  
  ... type="text"  
  ... name="invoice-address-form:firstname"  
  ... class="form-control"  
  ... maxlength="20"  
  ... required  
>  
<span id="address-form:firstname-error"></span>
```

RECHNUNGSADRESSE

Die gekennzeichneten Felder(*) sind Pflichtfelder.

ICH MELDE MICH AN ALS

Privatkunde Geschäftskunde

VORNAME *

NACHNAME *

Bitte geben Sie einen Vornamen an.

STRASSE & HAUSNUMMER *

PLZ *

ORT *

LAND *

TELEFON

E-MAIL *

Kundenkonto anlegen 

Weiter

[Als Firmenkunde anmelden.](#)

Bitte beachten Sie, dass eine nachträgliche Änderung der Rechnung von privat auf geschäftlich nicht möglich ist.

Zielformular

Meine Daten:

Anrede* Herr Frau

Vorname*

Nachname*

Straße / Hausnr.*

Postleitzahl / Ort*

Land* ▾

E-Mail-Adresse*

Telefon

Zusätzlicher Text

Mobilfunknr.

DUMMY


Zielformular

Meine Daten:


Anrede* Herr Frau

Vorname*


Nachname*

Straße / Hausnr.* 


Postleitzahl / Ort*

Land* 

E-Mail-Adresse*

Telefon 

Zusätzlicher Text

Mobilfunknr. 

DUMMY

Zielformular

Meine Daten:

Anrede* Herr Frau

Vorname*

Nachname*

Straße / Hausnr.*

Postleitzahl / Ort*

Land*

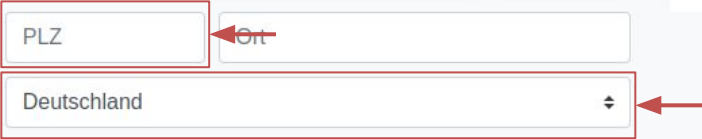
E-Mail-Adresse*

Telefon

Zusätzlicher Text

Mobilfunknr.

DUMMY





The image shows a registration form with several fields. Two red boxes highlight the 'Postleitzahl / Ort*' field (containing 'PLZ') and the 'Land*' field (containing 'Deutschland'). A red arrow points to the 'Ort' sub-field, and another red arrow points to the 'Land*' dropdown menu.

Dynamische Validierung

Straße / Hausnr.*	<input type="text" value="Straße"/>	<input type="text" value="Hausnr."/> ✖
Hausnr. ist ein Pflichtfeld		
Postleitzahl / Ort*	<input type="text" value="123"/> ✖	<input type="text" value="Ort"/>
Die eingegebene Postleitzahl muss 5 Stellen haben.		
Land*	<input type="text" value="Deutschland"/> ▾	

Dynamische Validierung

Straße / Hausnr.*	Straße	Hausnr.
Postleitzahl / Ort*	123 	Ort
Die eingegebene Postleitzahl muss 4 Stellen haben.		
Land*	Österreich 	

Zielformular

Meine Daten:

Anrede* Frau Herr

Vorname*

Nachname*

Straße / Hausnr.*

Postleitzahl / Ort*

Land* ▾

E-Mail-Adresse*

Telefon

Zusätzlicher Text

Mobilfunknr.

Please send my order to:

Company Name*

Title* ▾

First Name*

Last Name*

Job Title*

Postcode*

Address 1*

Address 2

City*

Country* ▾

E-Mail-Address*

Tel.

Anforderungen

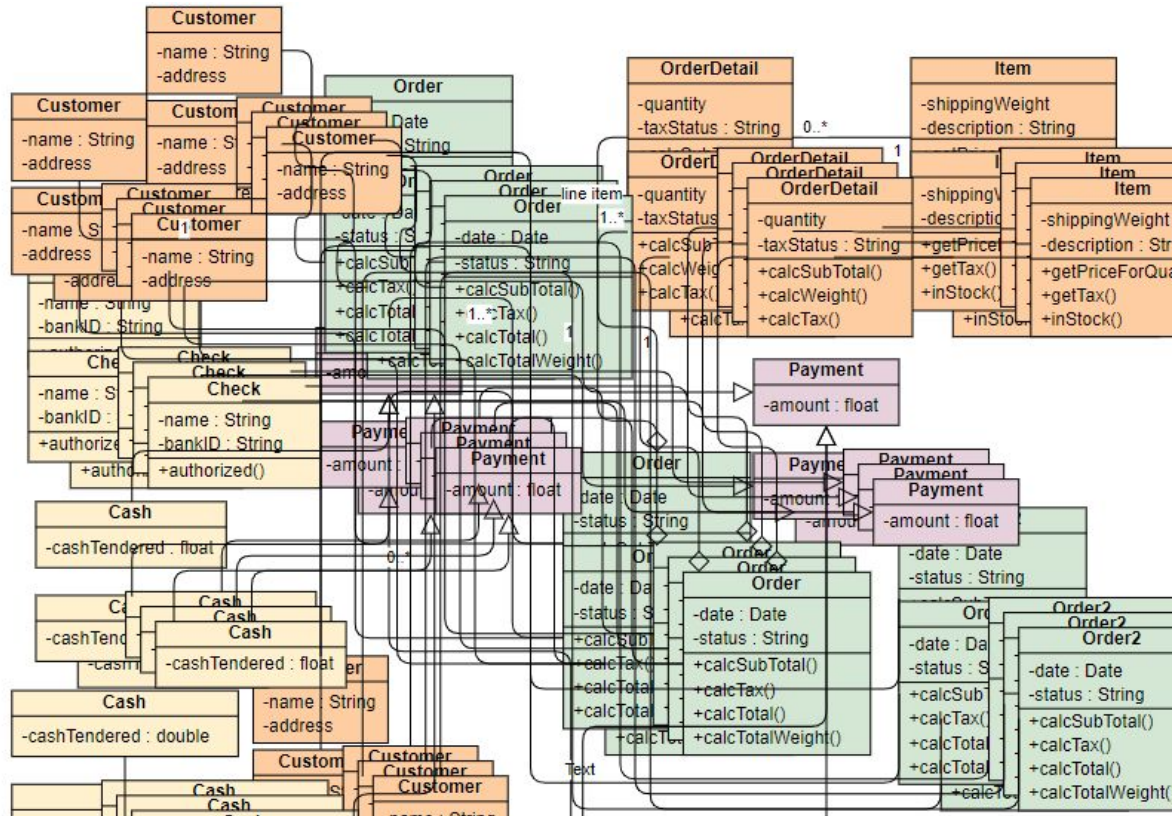
- Felder variieren je nach Zielgruppe, Lokalisierung und/oder Eingaben
- Pflichtfelder variieren nach Zielgruppe, Lokalisierung und/oder Eingaben
- Validierungsregeln variieren je nach Lokalisierung und/oder Eingaben
- Backend und Frontend Validierung in sync halten



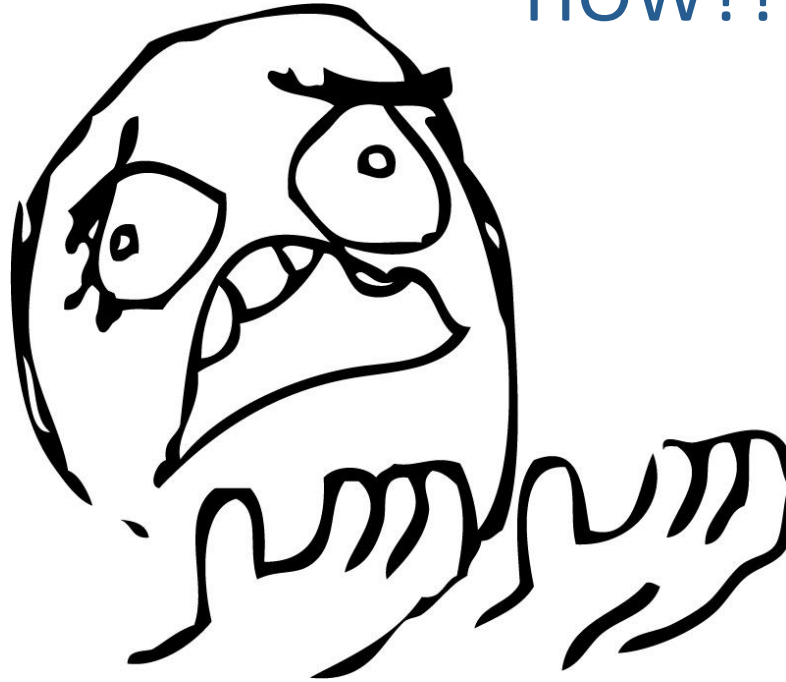
Anforderungen



Was kann schon schief gehen?



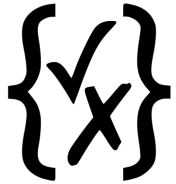
HOW?!



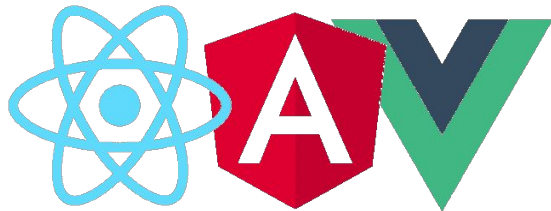


JSON-Forms

- Basiert auf JSON-Schema



- Support für Angular, React und Vue



JSONForms



JSON-Schema

```
Schema {  
  "type": "object",  
  "properties": {  
    "productId": {  
      "type": "integer"  
    },  
    "productName": {  
      "type": "string"  
    },  
    "price": {  
      "type": "number"  
    },  
    "tags": {  
      "type": "array",  
      "items": {  
        "type": "string"  
      }  
    }  
  }  
}
```



JSON-Schema

Schema

```
{
  "type": "object",
  "properties": {
    "productId": {
      "type": "integer"
    },
    "productName": {
      "type": "string"
    },
    "price": {
      "type": "number"
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

Data

```
{
  "productId": 1,
  "productName": "A green door",
  "price": 12.50,
  "tags": [ "home", "green" ]
}
```



Ajv (Another JSON Schema Validator)

Schema

```
{
  "type": "object",
  "properties": {
    "productId": {
      "type": "integer"
    },
    "productName": {
      "type": "string"
    },
    "price": {
      "type": "number"
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

Data



```
{
  "productId": 1,
  "productName": "A green door",
  "price": 12.50,
  "tags": [ "home", "green" ]
}
```

Ajv (Another JSON Schema Validator)

Schema

```
{
  "type": "object",
  "properties": {
    "productId": {
      "type": "integer"
    },
    "productName": {
      "type": "string"
    },
    "price": {
      "type": "number"
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

Data

X

```
{
  "productId": "6d97358c-7883-420c-8d18-180e6e397e46",
  "productName": "A green door",
  "price": "$12.50",
  "tags": "home,green"
}
```

dependentRequired

```
{  
  "type": "object",  
  "properties": {  
    "name": { "type": "string" },  
    "credit_card": { "type": "number" },  
    "billing_address": { "type": "string" }  
  },  
  "required": ["name"],  
  "dependentRequired": {  
    "credit_card": ["billing_address"]  
  }  
}
```



If-Then-Else

```
{
  "type": "object",
  "properties": {
    "street_address": {
      "type": "string"
    },
    "country": {
      "default": "United States of America",
      "enum": ["United States of America", "Canada"]
    }
  },
  "if": {
    "properties": { "country": { "const": "United States of America" } }
  },
  "then": {
    "properties": { "postal_code": { "pattern": "[0-9]{5}(-[0-9]{4})?" } }
  },
  "else": {
    "properties": { "postal_code": { "pattern": "[A-Z][0-9][A-Z] [0-9][A-Z][0-9]" } }
  }
}
```

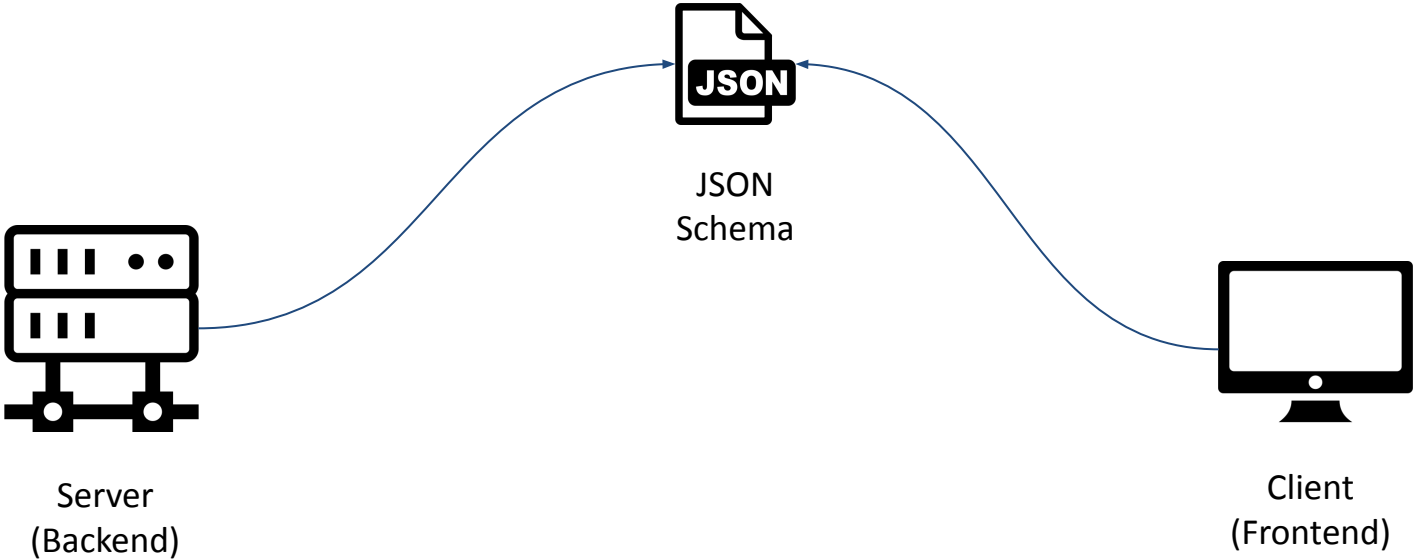

Implication

```
{
  "type": "object",
  "properties": {
    "restaurantType": { "enum": ["fast-food", "sit-down"] },
    "total": { "type": "number" },
    "tip": { "type": "number" }
  },
  "anyOf": [
    {
      "not": {
        "properties": { "restaurantType": { "const": "sit-down" } },
        "required": ["restaurantType"]
      }
    },
    { "required": ["tip"] }
  ]
}
```

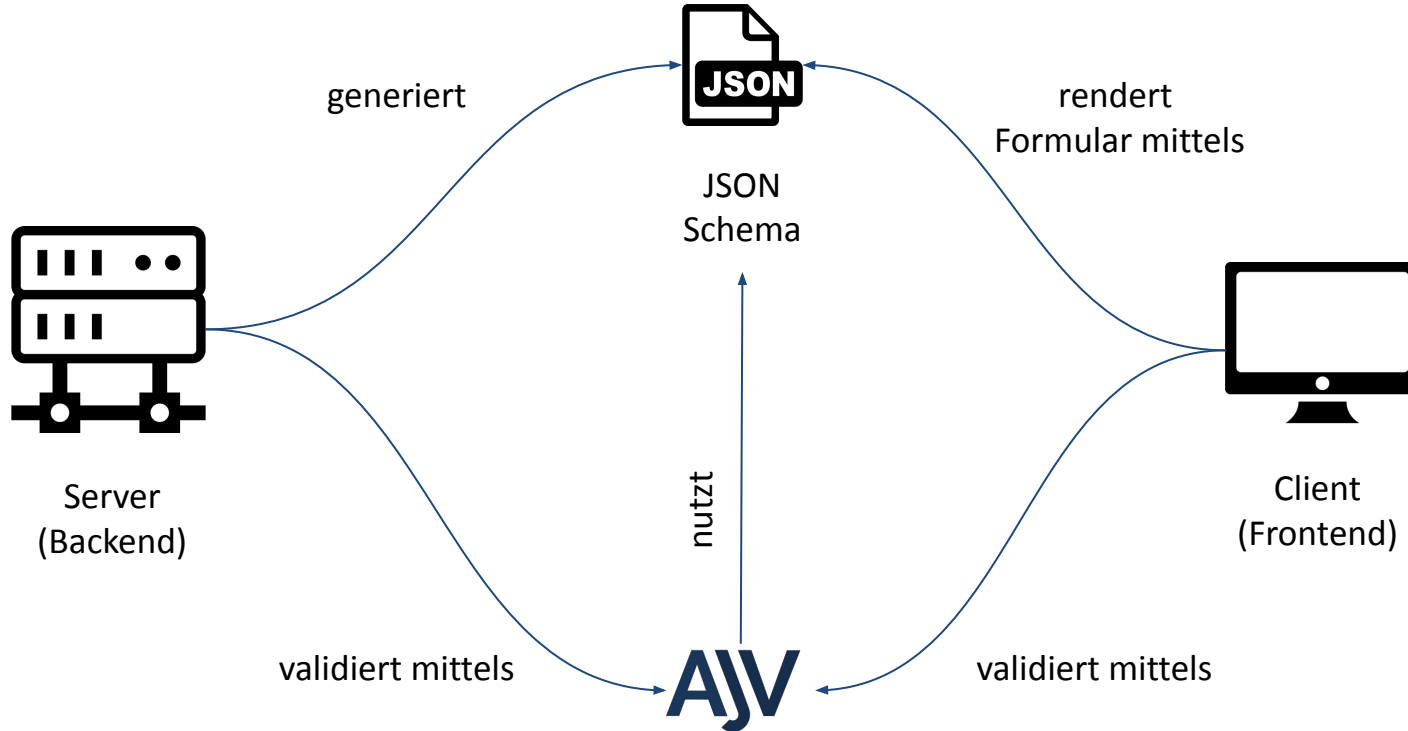
JSON-Schema + JSON-API + Another JSON Schema Validator + JSON



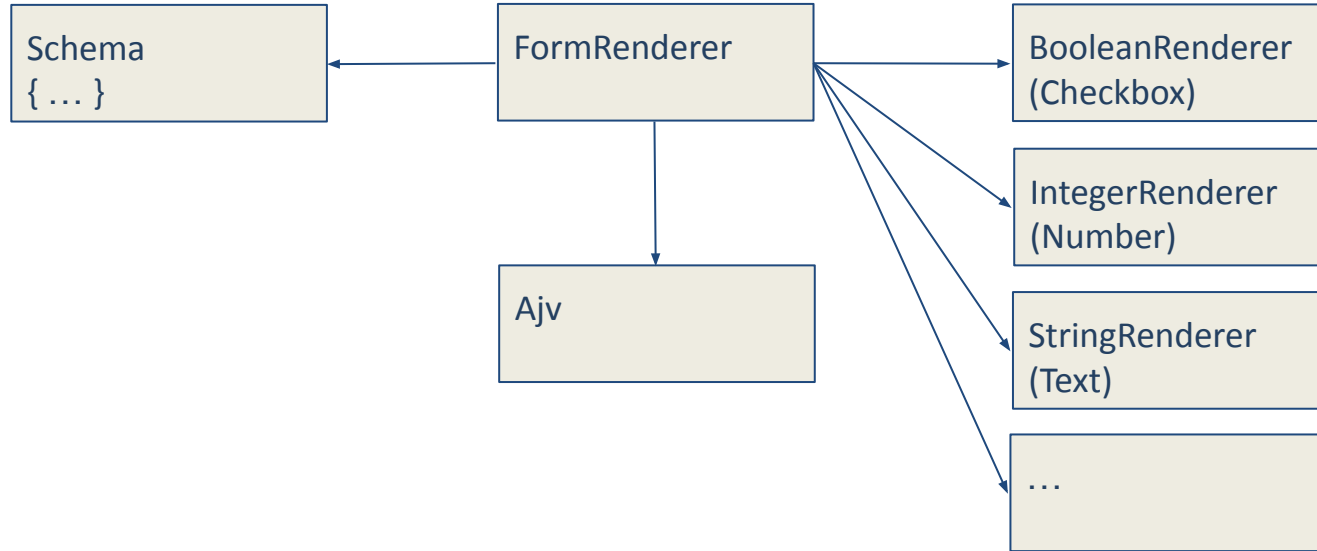
Schematische Darstellung



Schematische Darstellung



Frontend: FormRenderer



Code-Beispiel

IDE wird gestartet, bitte warten...



Fazit

Lohnt sich der Aufwand?

- Kontaktformular oder Admin-Interface
- Fancy User Interface (UI)
- Barrierefreiheit (**A**ccessible **R**ich Internet **A**pplication)
- Menge an Permutationen
- Kopplung zwischen den Feldern



Fazit

Nice 😊

- Backend/Frontend Validierung identisch
- Nahezu universal einsetzbar (nicht nur für Formulare)

But 😞

- State/Error-Handling (touched/pending/valid/invalid)
- Renderer benötigt zusätzliche Informationen zur Darstellung





Links

https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation

<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete>

<https://jqueryvalidation.org/>

<https://json-schema.org/>

<https://json-schema.org/understanding-json-schema/reference/conditionals.html>

<https://jsontypedef.com/>

<https://www.openapis.org/>

<https://jsonapi.org/>

<https://jsonforms.io/>

https://www.w3schools.com/html/html_forms.asp

Code-Beispiel:

<https://gitlab.com/c.klihm/json-schema-ajv-example>

