

# Web-Applikationen

Formulare



↳ Manuel Richardt, Dominik Rupprecht

↳ 22.03.23

**Hochschule Fulda**  
University of Applied Sciences



↳ Rückfragen zur SU/Übung?

„Ein Webformular ist eine organisierte Sammlung von Elementen, die dem Anwender erlauben, Daten anzugeben, zu verändern, auszuwählen oder mit diesen zu interagieren.“

Quelle: [https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulare/Was\\_ist\\_ein\\_Webformular%3F](https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulare/Was_ist_ein_Webformular%3F)

↙(Web)-Formulare sind ein komplexes und umfangreiches Thema

- ↙ Eingabe von Daten
- ↙ Veränderung von Daten
- ↙ Interaktion mit Webseite

Post -> für PW

```
<form action="/suchen" method="get">
  <label for="search">Suchbegriff:</label>
  <input type="text" id="search" name="search" placeholder="Suchtext">
  <input type="submit" value="Finden">
</form>
```

Suchbegriff:  Finden

- ↳ Form-Tag
  - ↳ Definiert ein Formular zur Benutzereingabe
  - ↳ Attribute: `action` und `method`
- ↳ Ein Formular kann multiple Elemente enthalten
  - ↳ Label, Input, Button, Select, Option, Optgroup, fieldset, Textarea, Output
  - ↳ Attribut `type` ist essentiell
- ↳ <https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulare>
- ↳ [https://www.w3schools.com/tags/tag\\_form.asp](https://www.w3schools.com/tags/tag_form.asp)

View .css

- ↳ action definiert Ziel des Formulars
  - ↳ Wohin soll der Inhalt des Formulars gesendet werden?
- ↳ method definiert die Methode mit welcher gesendet wird
  - ↳ Kann den Wert get oder post annehmen

```
<form action="/machwas" method="get">
```

Submit



↳ Sendet Daten des Formulars

```
<form action="/suchen" method="get">
  <label for="search">Suchbegriff:</label>
  <input type="text" id="search" name="search" placeholder="Suchtext">
  <input type="submit" value="Finden">
</form>
```

Suchbegriff: Suchtext

- ↳ Beschriftung von Eingaben
- ↳ for-Attribut bezieht sich auf die id oder name des Elements

```
<form action="/suchen" method="get">
  <label for="search">Suchbegriff:</label>
  <input type="text" id="search" name="search" placeholder="Suchtext">
  <input type="submit" value="Finden">
</form>
```

Suchbegriff:

- ↳ Formulare ohne Beschriftungen sind sinnlos!

- ↳ Nimmt Eingaben des Benutzers entgegen
- ↳ type-Attribut bestimmt Eingabemöglichkeit und Aussehen

```
<input type="button">  
<input type="checkbox">  
<input type="color">  
<input type="date">  
<input type="datetime-local">  
<input type="email">  
<input type="file">  
<input type="hidden">  
<input type="image">  
<input type="month">  
<input type="number">
```

```
<input type="password">  
<input type="radio">  
<input type="range">  
<input type="reset">  
<input type="search">  
<input type="submit">  
<input type="tel">  
<input type="text"> (default value)  
<input type="time">  
<input type="url">  
<input type="week">
```

## ↳ Drop-Down Inputs

## ↳ Option-Element bestimmt Items

```
<!DOCTYPE html>
<html>
<body>
    <h1>The select element</h1>
    <p>The select element is used to create a drop-down list.</p>
    <form action="/action_page.php">
        <label for="cars">Choose a car:</label>
        <select name="cars" id="cars">
            <option value="volvo">Volvo</option>
            <option value="saab">Saab</option>
            <option value="opel">Opel</option>
            <option value="audi">Audi</option>
        </select>
        <br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

## The select element

The select element is used to create a drop-down list.

Choose a car:

# Web-Applikationen

Request & Response



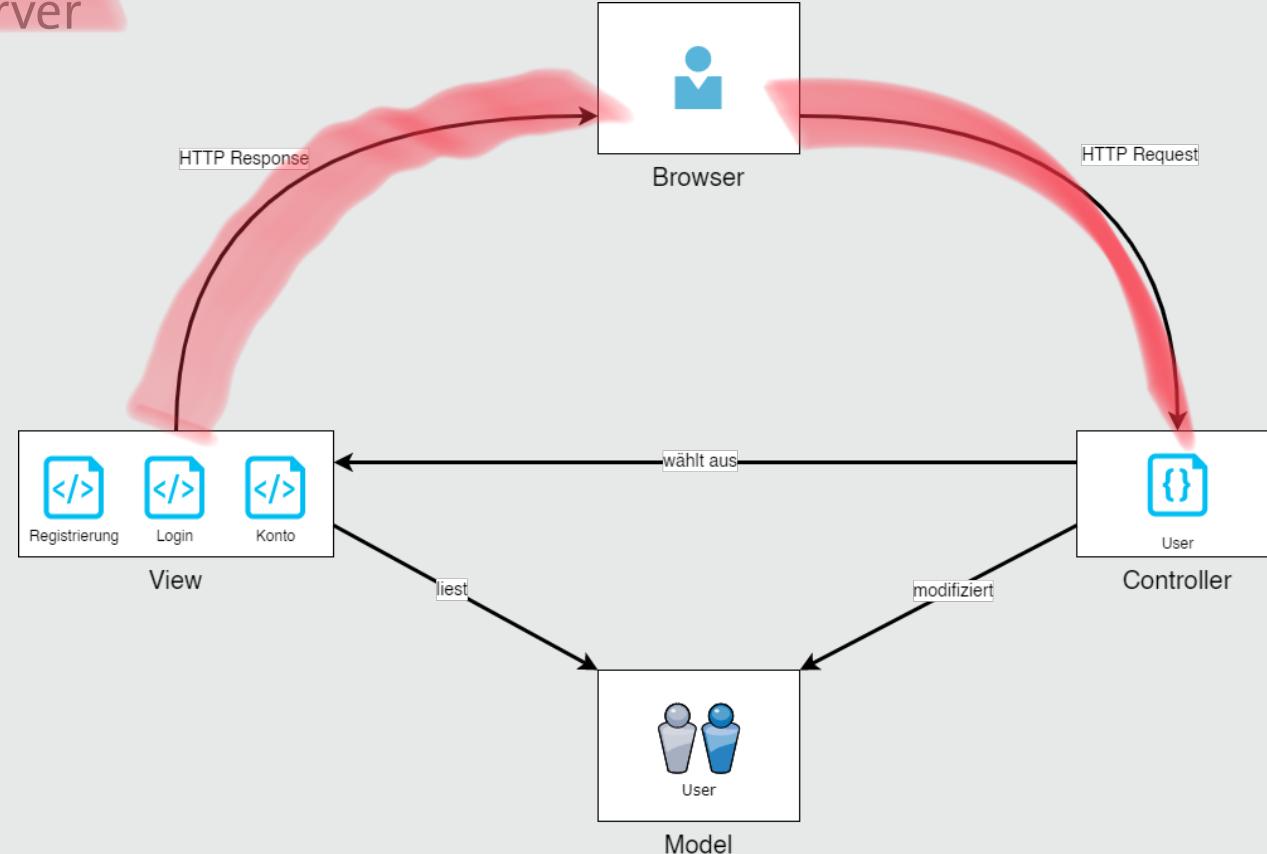
**Hochschule Fulda**  
University of Applied Sciences



# Request & Response

AI

- Request ist Anfrage an den Server
- Response seine Antwort



# Request & Response in node.js

AI

## ↳ router.get

- ↳ Reagiert auf Route (Adresse)
- ↳ Get-Methode

## ↳ req

- ↳ Request
- ↳ Kann Daten von einem Formular beinhalten

## ↳ res

- ↳ Response des Servers

## ↳ console.log

- ↳ Ausgabe auf Console
- ↳ Debug!

```
router.get('/', function(req, res, next) {
  res.render('index', {
    subject: 'EJS template engine',
    name: 'our template',
    link: 'https://google.com'
  });
});
router.get('/manuel', function(req, res, next) {
  console.log("Hello, Manuel");
  res.render('index', { title: 'Manuel' });
});
router.get('/dominik', function(req, res, next) {
  console.log("Hello, Dominik");
  res.render('index', { title: 'Dominik' });
});
```

# Request

AI

```
routes > JS index.js > ...
  8
  9 router.get('/altersberechnung', function(req, res, next) {
10   /* Werte aus Request auslesen*/
11   let fname = req.query.vorname;
12   let bdate = req.query.birthdate;
13
14   /* Umwandlung zu Timestamp -> In Millisekunden*/
15   let date_birth = new Date(bdate).getTime();
16   let date_now = new Date(Date.now()).getTime();
17
18   let days = /* Hier kommt noch eine Umrechnung hin*/
19   console.log(days);
20
21   let greet = fname + ' ist ' + days + ' Tage alt.';
22   res.render('index', { title: 'Express', greeting: greet });
23 });
24 module.exports = router;
25 |
```

```
views > HTML index.ejs > ...
  1  <!DOCTYPE html>
  2  <html>
  3    <head>
  4      <title><%= title %></title>
  5      <link rel='stylesheet' href='/stylesheets/style.css' />
  6    </head>
  7    <body>
  8      <h1><%= title %></h1>
  9      <p>Welcome to <%= title %></p>
10      <p><%= greeting %></p>
11      <form action="/altersberechnung" method="get">
12        <label for="vorname">Vorname:</label>
13        <input type="text" id="vorname" name="vorname">
14        <label for="birth">Geburtsdatum:</label>
15        <input type="date" id="birthdate" name="birthdate">
16        <input type="submit" value="Submit">
17      </form>
18    </body>
19  </html>
```

↳ Neben `res.render` gib es noch andere Response Objekte

## Response methods

The methods on the response object (`res`) in the following table can send a response to the client, and terminate the request-response cycle. If none of these methods are called from a route handler, the client request will be left hanging.

Method	Description
<code>res.download()</code>	Prompt a file to be downloaded.
<code>res.end()</code>	End the response process.
<code>res.json()</code>	Send a JSON response.
<code>res.jsonp()</code>	Send a JSON response with JSONP support.
<code>res.redirect()</code>	Redirect a request.
<code>res.render()</code>	Render a view template.
<code>res.send()</code>	Send a response of various types.
<code>res.sendFile()</code>	Send a file as an octet stream.
<code>res.sendStatus()</code>	Set the response status code and send its string representation as the response body.

- ↳ Formulare sind für die Interaktion mit dem Benutzer essentiell
- ↳ Form-Element muss `action` und `method` definieren und ein Submit beinhalten
- ↳ Sinnvolle Beschreibung der Eingabemöglichkeiten
- ↳ Request beinhaltet Route und ggf. Parameter
- ↳ Response ist die Antwort des Servers auf eine Anfrage