

ErstiWeb Pseudo-CMS Anleitung

Alle Inhalte sind eingebettet im Verzeichnis `_content`. Es existieren die Dateien *bodies.json*, *calendar.json*, *consulting.json*, *contacts.json*, *infos.json*, *notes.json*, *proposasl.json*, *rooms.json*, *search.json*, *tutors.json*

Diese Dateien dienen der Verknüpfung der Web-Komponenten mit den darzustellenden Inhalten. Alle Inhalte werden generell dynamisch gerendert, Hinzufügen oder Löschen eines Datenpunkts wird vom System automatisch übernommen. Wenn Inhalte geändert werden kann es unter modernen Browsern vorkommen, dass diese nicht unmittelbar dargestellt werden, hierfür ist das Caching des Browsers oder Webservers verantwortlich. Dies kann durch Leeren des Caches umgangen werden, dieser wird aber üblicherweise nach einigen Minuten automatisch invalidiert.

Alle Dateien für das ErstiWeb Pseudo-CMS sind im JSON Format gespeichert wofür online diverse Hilfeseiten existieren, bspw. https://www.w3schools.com/js/js_json_datatypes.asp. Da JSON ein striktes Datenformat fordert und Fehler schnell passieren können empfehle ich die Validierung der Inhalte, bspw. über <https://jsonlint.com/>, falls die Seite Inhalte nicht mehr darstellt.

Manche Bereiche des ErstiWebs lassen HTML-Inhalte zu. Für einen Einstieg in html empfehle ich https://www.w3schools.com/html/html_elements.asp. Hierbei sollte beachtet werden, dass nicht alle HTML-Tags für alle Inhalte formatiert sind. Die Nutzung von `<p>` (Paragraphs) `<h1>` bis `<h6>` (Headings groß bis klein) und `
` (Erzwungener Zeilenumbruch) wird empfohlen.

bodies.json

Schema:

```
[
  {
    "content": String,
    "link": String
  }
]
```

Beispiel:

```
[
  {
    "content": "FASTA FB Design",
    "link": "https://www.fh-dortmund.de/de/fb/2/fasta/index.php"
  }
]
```

Anmerkung:

- Eine Notizblock-Komponente

calendar.json

Schema:

```
[
  {
    "title": String,
    "category": String,
    "place": String,
    "date": String,
    "time": String
  }
]
```

Beispiel:

```
[
  {
    "title": "Tag der Deutschen Einheit",
    "category": "Feiertage",
    "place": "Aula / R100",
    "date": "2020-10-03",
    "time": "17:00 Uhr"
  }
]
```

Anmerkung:

- Kalendereinträge können eine Beliebige Kategorie haben und werden dementsprechend gruppiert.
- Das Datum „date“ sollte vom System interpretierbar sein. Dafür eignen sich ISO Dates (2015-03-25) Short Dates (03/25/2015) oder Long Dates (Mar 25 2015) oder (25 Mar 2015)

consulting.json

Schema:

```
[  
  {  
    "content": String,  
    "link": String  
  }  
]
```

Beispiel:

```
[  
  {  
    "content": "Studienbüro & Prüfungsangelegenheiten",  
    "link": "https://www.fh-dortmund.de/de/studi/studbuero/index.php"  
  }  
]
```

Anmerkung:

- Eine Notizblock-Komponente

contacts.json

Schema:

```
[
  {
    "title": String,
    "firstname": String,
    "lastname": String,
    "category": String,
    "contactInfos": [
      {
        "key": String,
        "value": String
      }
    ]
  }
]
```

Beispiel:

```
[
  {
    "title": "Prof.",
    "firstname": "Roger",
    "lastname": "Walk",
    "category": "Fachbereich Design",
    "contactInfos": [
      {
        "key": "Org.-Einheit",
        "value": "Fachbereich Design"
      },
      {
        "key": "Raum",
        "value": "Mal hier mal Dort"
      }
    ]
  }
]
```

Anmerkung:

- Der Titel wird in der Übersichtsliste nicht angezeigt, aber in der Detailansicht
- Alle Werte, inklusive der *contactInfos* sind durchsuchbar
- Es können beliebig viele *contactInfos* hinzugefügt werden.

infos.json

Schema:

```
[
  {
    "category": String,
    "title": String,
    "date": String,
    "content": (HTML-)String
  }
]
```

Beispiel:

```
[
  {
    "category": "Bachelor",
    "title": "Test",
    "date": "Hier kann eigentlich stehen was will",
    "content": "<h1>Eine Überschrift</h1><p>Und wichtiger Text</p>",
  }
]
```

Anmerkung:

- Die Inhalte werden nach der Kategorie gruppiert
- *Title* und *Date* sind prinzipiell beliebige Werte und haben abgesehen von der Darstellung keine Funktion
- *Content* kann beliebiger HTML-Inhalt sein

notes.json

Schema:

```
[  
  String  
]
```

Beispiel:

```
[  
  "./assets/Einleger-Leben.png",  
  "./assets/Einleger-Leben2.png"  
]
```

Anmerkung:

- Einfaches Text-Array
- Es werden Bilder erwartet

proposals.json

Schema:

```
[
  {
    "content": String,
    "link": String
  }
]
```

Beispiel:

```
[
  {
    "content": "Internationale Angebote",
    "link": "https://www.google.de/?q=FH+Dortmund+Internationale+Angebote"
  }
]
```

Anmerkung:

- Eine Notizblock-Komponente

rooms.json

Schema:

```
[
  {
    "banner": String,
    "images": [String],
    "text": String
  }
]
```

Beispiel:

```
[
  {
    "banner": " ./assets/mensa.svg",
    "images": [
      " ./assets/Räume/Mensa/DSC_0311-positive.jpg",
      " ./assets/Räume/Mensa/nikonf65tmax20050.jpg"
    ],
    "text": ""<h1>Eine Überschrift</h1><p>Und wichtiger Text</p>"
  }
]
```

Anmerkung:

- Für *banner* und *images* werden Links zu Bildern erwartet
- *Text* kann beliebiger HTML-Inhalt sein

search.json

Schema:

```
[
  {
    "term": String,
    "target": { "component": String } | { "link": String }
  }
]
```

Beispiel:

```
[
  {
    "term": " Test",
    "target": {"component": "tutors"}
  },
  {
    "term": " Google",
    "target": {"link": "https://www.google.com"}
  },
]
```

Anmerkung:

- Der Suchbegriff wird auf *term* gematched. Auch ein Teiltreffer wird hierbei angezeigt.
- Das Ziel kann eine interne Verlinkung sein ({*"component": "tutors"*}). Hier können alle benannten Komponenten eingetragen werden (search, rooms, tutors, consulting, proposals, bodies, notes, info, contact, calendar).
- Das Ziel kann auch eine externe Verlinkung sein ({*"link": "example.com"*}), hierbei kann ein beliebiger Link eingetragen werden.

tutors.json

Schema:

```
[  
  String  
]
```

Beispiel:

```
[  
  "./assets/ Tutoren/Anna_sw.jpg",  
  "./assets/ Tutoren/dilara_sw.jpg"  
]
```

Anmerkung:

- Einfaches Text-Array
- Es werden Bilder erwartet