# **DBWT Dossier (Gr. F)**

**Autoren :**

* **Tadjiev, Nodirjon (3527449)**
* **Winata, Daniel (3525700)**

## **Meilenstein 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aufgabe** | **Geschätze Zeit (Minuten)** | **Tatsächliche Zeit (Minuten)** | **Kommentar** |
| 1 | 5 | 3 | Die Aufgaben waren gut und klar strukturiert, deswegen waren wir früher fertig. |
| 2 | 2 | 2 | - |
| 3 | 5 | 5 | Die Tabelle wurde schon in der Vorlesung gezeigt. |
| 4 | 30 | 60 | War schwieriger als erwartet, da man hier viel googeln muss. |
| 5 | 45 | ~90 | Ähnlich letzter Aufgabe und erster Umgang mit Input-Feldern |
| 6 | 60 | 60 | - |
| 7 | 5 | 5 | Paar Kleinigkeiten wurden korrigiert. |
| 8 | 60 | 30-40 | www.css-tricks.com |
| 9 | 120 | ~120 | Hat Spaß gemacht! (erste wirkliche Seite) |
| 10 | 360 | ~500 | Wir hatten am Anfang das Layout mit Flex geschrieben und dann auf Grid geändert, da man mit Grid mehr Flexibilität hat. |

### **Aufgabe 6**

**6.1)**

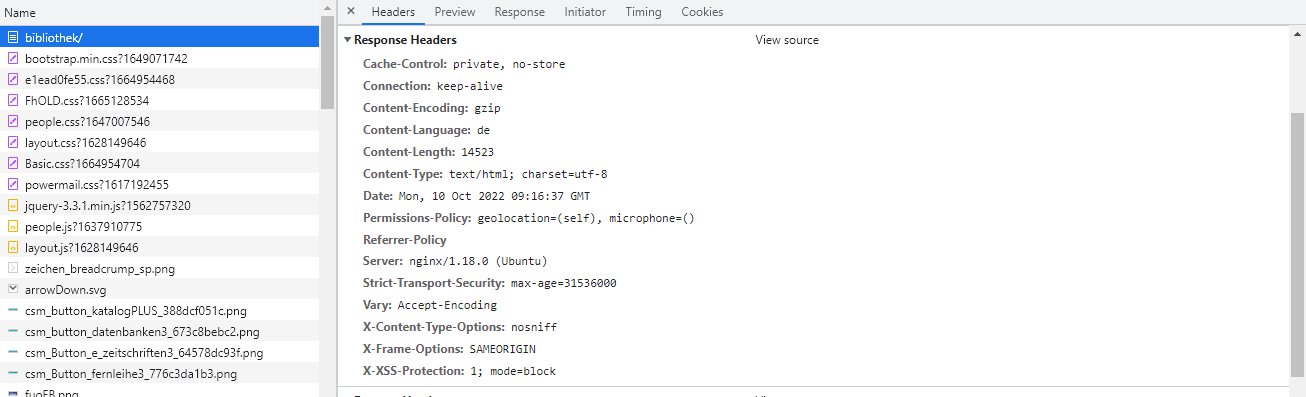
**Anfrage-Header :**



**5 Attributen des Anfrage-Headers :**

1. **Accept :** gibt die Auskunft, welche Formate der Client in der Antwort unterstützt und besteht aus Paaren mit zwei Werten (Fortmat und q-Faktor).
2. **Cache-Control :** bestimmt das Caching-Verhalten des Browsers und beinhaltet einzelne Direktiven für die Steurung des Caches. Z.B. hier max-age = 0.
3. **Connection :** reguliert, ob die Verbindung zwischen dem User Agent und Webserver für weitere Übertragung von Daten/Dateien aufrecht erhalten werden soll.
4. **Cookie :** Cookies ermöglichen das clientseitige Speichern von Information, die auch vom Server stammen können und die bei weiteren Aufrufen für den Benutzer transparent an den Server übertragen werden. Dadurch erleichtern Cookies die Benutzung von Webseiten, die auf Benutzereinstellungen reagieren oder den Aufbau von Sitzungen.
5. **Host :** Host gibt den Host und die Portnummer des Servers an, an den die Anfrage gesendet wird. Wenn es keine Portnummer steht, wird die Default-Portnummer verwendet(HTTP : 80).

Antwort-Header :

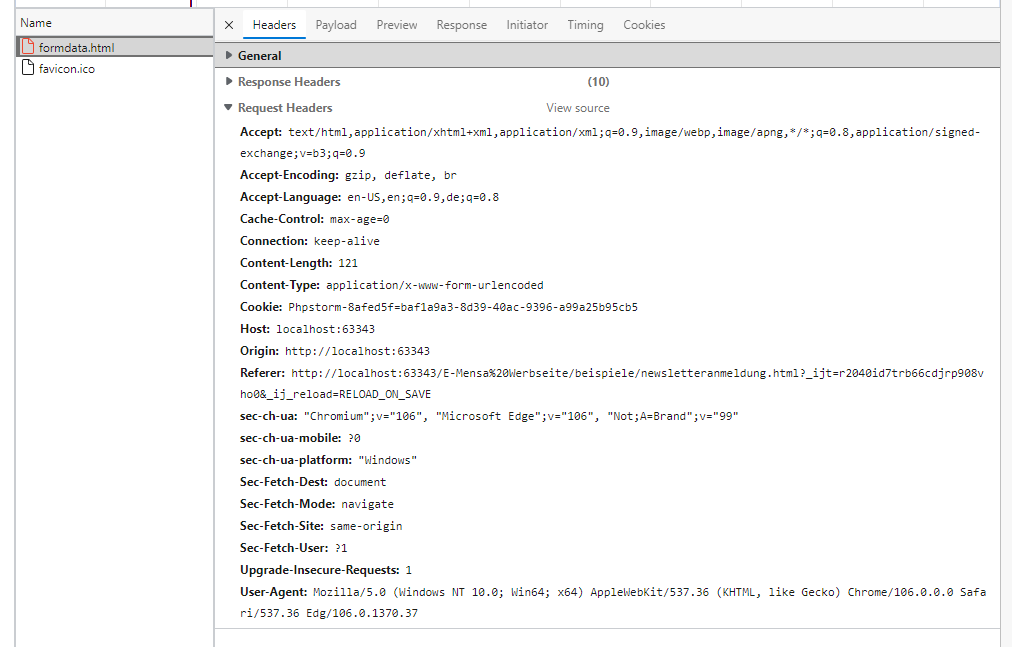


**5 Attributen des Antwort-Headers :**

1. **Cache-Control :** bestimmt das Caching-Verhalten des Browsers und beinhaltet einzelne Direktiven für die Steurung des Caches. Z.B. private und no-store.
2. **Connection :** reguliert, ob die Verbindung zwischen dem User Agent und Webserver für weitere Übertragung von Daten/Dateien aufrecht erhalten werden soll.
3. **Content-Language :** sie wird verwendet, um die Sprache des Inhalts zu beschreiben, das für das bestimmte Publikum bestimmt ist.
4. **Content-Type :**  liefert dem Client den tatsächlichen Inhaltstyp des zurückgegebenen Inhalts (hier .html Datei encodiert in UTF-8).
5. **Date :** gibt die Zeit und das Datum an, wann die Datei entsteht.

**6.2)**

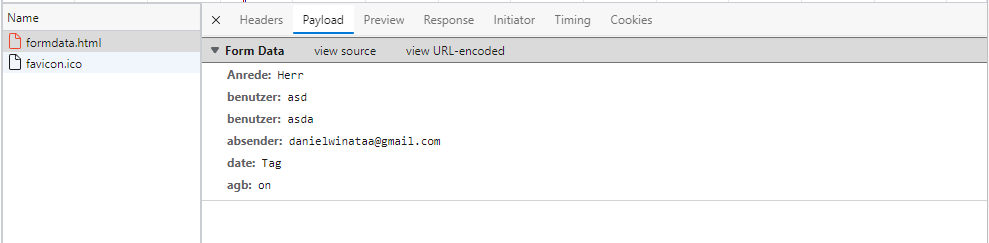
**Anfrage-Header :**



**5 Attributen des Anfrage-Headers :**

1. **Accept :** gibt an, welche Inhaltstypen, ausgedrückt als MIME-Typen, der Client verstehen kann.
2. **Accept-Encoding :** gibt die Inhaltskodierung (in der Regel ein Komprimierungsalgorithmus) an, die der Client verstehen kann.
3. **Accept-Language :** Welche Sprachen der Client akzeptiert. Falls der Server passend eingerichtet ist und die Sprachversionen vorhanden sind, wird über [Content Negotiation](https://wiki.selfhtml.org/wiki/Content_Negotiation) die passende Datei ausgeliefert.
4. **Cache-Control :** bestimmt das Caching-Verhalten des Browsers und beinhaltet einzelne Direktiven für die Steurung des Caches. Z.B. hier max-age = 0.
5. **User-Agent :** ist eine charakteristische Zeichenkette, die es Servern und Netzwerk-Peers ermöglicht, die Anwendung, das Betriebssystem, den Hersteller und/oder die Version des anfragenden User-Agents zu identifizieren.

**Request-Body :**



**Aufgabe 9**

<style>

/\*  
Specifies the body font as Arial and sets serif as a backup for the case of not loading  
Setting the height of the maximum page size  
 \*/  
body {  
 font-family: "Arial", serif;  
 min-height: 100vh;  
}  
  
/\*  
Making the headline bold  
 \*/  
#elektro {  
 font-weight: bold;  
}  
  
/\*  
Tabelle link is bold and coloured to chocolate without underlining  
 \*/  
a {  
 color: chocolate;  
 font-weight: bold;  
 text-decoration: none;  
}  
  
/\*  
The table takes automatic margin and padding distances and has a relative width of 45%  
 \*/  
#etabelle {  
 border: 2px solid chocolate;  
 margin-left: auto;  
 margin-right: auto;  
 width: 45%;  
}  
  
/\*  
Color the header and second row of the table to bisque color  
 \*/  
thead, tbody tr:nth-of-type(2) {  
 background-color: bisque;  
}  
  
/\*  
Each list element has top and bottom margins of 15px  
 \*/  
li {  
 margin-top: 15px;  
 margin-bottom: 15px;  
}  
  
/\*  
The footer is fully wide and its text is aligned to right and bottomed  
 \*/  
footer {  
 width: 100%;  
 text-align: right;  
 bottom: 0;  
}  
  
/\*  
Each word of the list in footer is inlined  
 \*/  
footer ul li {  
 display: inline;  
 margin-right: 1em;  
}  
  
/\*  
The sticks dividing the words in footer  
 \*/  
footer ul li:nth-of-type(1), footer ul li:nth-of-type(2), footer ul li:nth-of-type(3) {  
 border-right: 2px solid chocolate;  
 padding-right: 1em;  
}  
  
/\*  
Text after the table  
 \*/  
.center {  
 border-bottom: 2px solid chocolate;  
 text-align: center;  
}  
  
/\*  
The letter E had underlining. Removed it  
 \*/  
[title="Extra"] {  
 text-decoration: initial;  
}

</style>

## **Meilenstein 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aufgabe** | **Geschätze Zeit (Minuten)** | **Tatsächliche Zeit (Minuten)** | **Kommentar** |
| 1 | 5 | 10 | Schwierigkeiten php.ini zu finden |
| 2 | 30 | 30 | 30 Aufgaben. 1 Aufgabe = 1 min |
| 3 | 360 | 400 | Da erster Einstieg ins php, daher viel Zeit |
| 4 | 5 | 5 | - |
| 5 | 60 | 90 | - |
| 6 | 120 | 150 | - |
| 7 | 60 | 70 | War spannend mit text Dateien zu arbeiten |
| 8 | 120 | 120 | War einfach, weil wir viel davor geübt haben |

### **Aufgabe 3**

1)

Zeile 12: Comma (,) addiert

Zeile 21: Eckige Klammern (]) addiert

Zeile 41: Runde Klammern ()) addiert

Zeile 59: Das Wort "function" addiert

2)

Im Quelltext stehen solche Funktionalitaeten wie: einfache Arrays, verschachtelte Arrays und eine Funktion zur Verfuegung.

3)

**$\_GET** ist ein Array welches alle Parameter aus der vom Besucher aufgerufenen URL enthält. Lautet die aufgerufene URL etwa *http://www.example.com/index.php?****page=article&id=1015&design=normal***, dann würde $\_GET die folgenden Parameter enthalten:

* page = article
* id = 1015
* design = normal

Entsprechend wäre das $\_GET-Array identisch mit *array('page' => 'article', 'id' => '1015', 'design' => 'normal')*. (Alle Parameterschlüssel und -werte werden im $\_GET-Array als Strings abgespeichert.)

Das $\_GET-Array ist eine superglobale Variable, kann also von überall im Skript jederzeit aufgerufen werden. Wurden keine Parameter übergeben, dann ist das Array zwar vorhanden, aber leer.

**strpos**: Gibt die numerische Position des ersten Vorkommens von needle innerhalb der Zeichenkette haystack zurück.

**count**: Zählt alle Elemente eines Arrays, wenn es mit einem Array verwendet wird. Bei Verwendung mit einem Objekt, das die Schnittstelle [Countable](https://www.php.net/manual/de/class.countable.php) nutzt, gibt es den Rückgabewert der Methode [Countable::count()](https://www.php.net/manual/de/countable.count.php) zurück.

: **float**: erwarteter Rückgabewert ist vom Typ float.