

HTTP dynamisch bis hin zum Application Server

Dynamik im Server...

Wenn ein Server nicht stumpf Ressourcen ausliefert (also z.B. einen Request 1:1 mit einer bestimmten Datei beantwortet), dann beginnt Dynamik.

Dazu kann er

- die URL im Request programmatisch auswerten und
- sich zu einem bestimmten Client den *Zustand* merken (z.B. über Cookies).

Werden die Zustände bestimmter Clients auch über Sitzungen hinweg gespeichert (in einer Datenbank) oder folgen Requests/Response einer programmatischen Ablauflogik, dann mutiert der Webserver allmählich zum *Application Server*.

Die „programmatische“ Auswertung der URL eines Requests wurde schon in einem früheren Kapitel angedeutet.

... und im Client.

Der Browser kann natürlich auch dynamisch sein und nicht nur die HTML Seite gut darstellen,

Unter anderem gibt es mit *JavaScript* eine Möglichkeit, wie in das HTML-Dokument Code eingebettet werden kann, der nach Aufbau der HTML-Darstellung durch den Browser interpretiert wird und mit dem

- der Inhalt der Darstellung oder die Form der Webseite nachträglich/fortlaufend verändert werden kann,
- der Nutzer der Webseite interagieren kann oder
- im Hintergrund weitere Ressourcen vom Webserver abgefragt und in die aktuelle Webseitendarstellung eingebaut werden kann.

Beispiel

(ein zugegeben etwas einfältiges...)

Szenario: Die Webseite soll die **Uhrzeit** anzeigen!

- **Lösungsversuch1:** Statischer Webserver liefert *uhrzeit.html* aus. Dann stimmt die Uhrzeit

genau einmal pro Tag



```
<html>
```

```
<body>  
12:05:31  
</body>  
</html>
```

- **Lösungsversuch 2:** Dynamischer Webserver liefert die Uhrzeit zum Zeitpunkt des Request/Response aus: <https://app.qgelm.de/clock>

Der kleine Teil des Programmcodes sieht dann wie folgt aus (das ganze ist eingebettet in ein Python Web-Application-Framework namens *web2py*):

```
import datetime  
  
# ---- index page ----  
def index():  
    clock = datetime.datetime.utcnow().strftime("%H:%M:%S")  
    return dict(message=clock)
```

- **Lösungsversuch 3:** Statischer Webserver liefert JavaScript-Code dazu aus:
<http://schulung.qgelm.de/uhrzeit.html>

```
<!DOCTYPE html>  
<html>  
  
<body onload="startTime()">  
  
<div id="clock"></div>  
  
<script>  
function startTime() {  
    const today = new Date();  
    let hh = today.getHours();  
    let mm = today.getMinutes();  
    let ss = today.getSeconds();  
    hh = (hh < 10) ? "0" + hh : hh;  
    mm = (mm < 10) ? "0" + mm : mm;  
    ss = (ss < 10) ? "0" + ss : ss;  
  
    document.getElementById('clock').innerHTML = hh + ":" + mm + ":" + ss;  
    setTimeout(startTime, 1000);  
}  
</script>  
  
</body>  
</html>
```

From:
<https://schnipsl.qgelm.de/> - **Qgelm**

Permanent link:
<https://schnipsl.qgelm.de/doku.php?id=schulung:ht...>

Last update: **2022/01/15 14:27**

