Aufgabenblatt 8

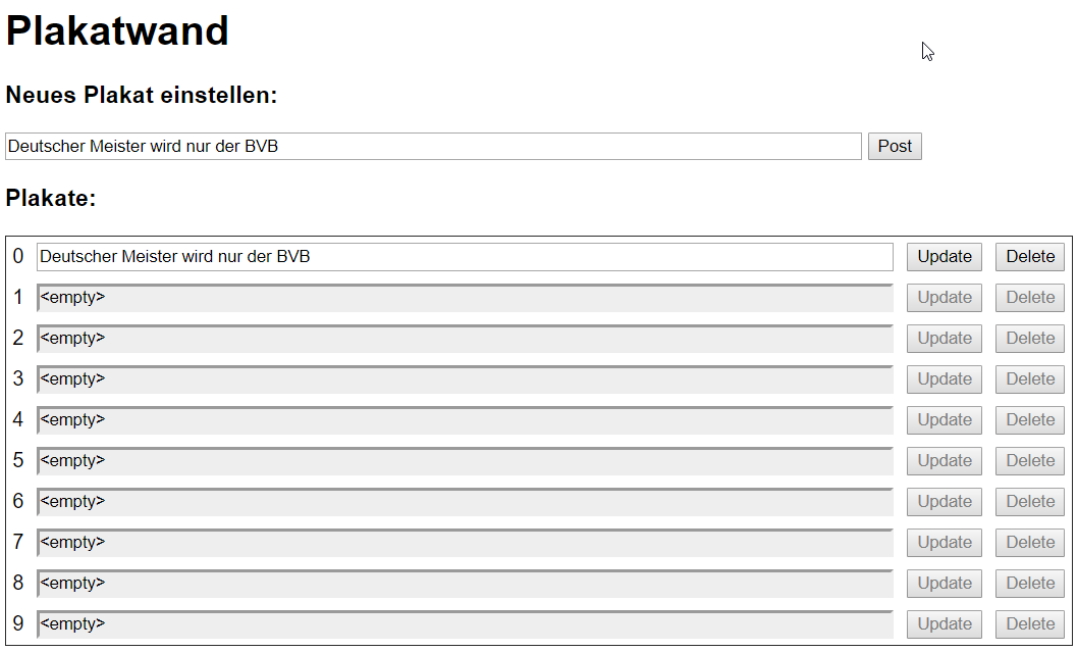
Aufgabe – RESTful Web-Services und AJAX

In dieser Aufgabe sollen RESTful Web-Services und der asynchrone Zugriff darauf via JavaScript/AJAX untersucht werden. Insbesondere geht es darum, sog. CRUD-Operationen auf im Server gehaltene Datenobjekte durch die bekannten http-Verben (GET, POST, PUT, DELETE) umzusetzen.

CRUD-Operation	http-Methode	Funktion
Create	**`POST`**	Erzeugen eines Eintrages, hier für eine Plakatwand.
Read	**`GET`**	Lesen des aktuellen Zustands der Plakatwand.
Update	**`PUT`**	Aktualisierung des aktuellen Zustands. Hier: eines Plakates auf der Wand.
Delete	**`DELETE`**	Löschen eines Objektes. In diesem Fall: Löschen eines Plakats auf der Wand.

Als Beispiel wird hier ein sog. *Billboard* (dt. Plakatwand) untersucht:

- Nutzer können ihre *Posts* (= Plakate) dort einstellen.
- Die Fläche ist begrenzt, weshalb ggf. existierende Posts durch neue ersetzt werden müssen.
- Nutzer können alle Posts sehen, die eigenen editieren und löschen.



Im OSCA-Bereich zur Vorlesung finden Sie die Grundstruktur der Applikation. Die Read/Get-Methode ist hier exemplarisch umgesetzt. Wie man sieht, wird der Inhalt des Billboards mit der Read/Get-Methode zwischen Service und Client in Form einer html-Tabelle ausgetauscht.

Ihre Aufgabe besteht darin,

1. die fehlenden CRUD-Operationen zu implementieren und
2. den Datentransfer von html auf XML oder JSON umzustellen. Dazu müssen Sie ggf. (je nach verwendetem JDK) externe Bibliotheken dazu. Das ist auf Basis der Maven-Schablone recht einfach, Sie müssen einfach eine Dependency zur Projektdatei **`pom.xml`** hinzufügen. Für die Nutzung von JSON ist dies:

```
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20200518</version>
</dependency>
```

3. Reduzieren Sie das Datenaufkommen, in dem Sie die Anzahl der Requests und die Größe der Server-Responses minimieren. Beschreiben Sie Ihre Maßnahmen im Protokoll.

Hinweise:

- Es werden keinerlei Frameworks (z.B. JAX-RS) zur Erstellung der Web-Services benötigt.
- Aus Gründen der Übersichtlichkeit wurden einige Aspekte (Nutzer-Management, Sitzungen, Sicherheit,…) nicht berücksichtigt. Die Applikation ist so gesehen wirklich minimal.
    - Beispielsweise erfolgt die Entscheidung, ob eine Post editiert oder gelöscht werden kann, auf Basis der IP-Adresse des abfragenden Clients.
    - Gern können Sie die fehlenden Funktionen zur Authentifizierung und Autorisierung auf Basis der in der Vorlesung gelernten Techniken ergänzen (freiwillige Aufgabe).
- Die Aktualisierung des Billboards erfolgt in der zur Verfügung gestellten Version per Polling. Als weitaus elegantere Lösung sollten idealerweise nur die geänderten Einträge per Ajax-Callback modifiziert werden (Long-Polling).
- Beachten Sie, dass Web-Browser mit Caches ausgestattet sind. Insbesondere bei Änderungen der JavaScript-Datei muss ein Neuladen erzwungen werden, damit die Änderungen sichtbar werden.

Für das Testat ist ein doppelseitiger Ausdruck (inkl. Namen der Gruppenmitglieder) der Dokumentation abzugeben.

Testierung: 14./15.6.2021