Fachhochschule Bielefeld
Lehrgebiet Softwaretechnik und Programmierung
Prof. Dr. Jörg Brunsmann

Übungen zur Vorlesung

Spezielle Gebiete zum Software Engineering

Sommersemester 2015

Aufgabenblatt 5: Node.js

Hinweis: Benutzen Sie für die Aufgaben 1 und 2 weder das Express- noch das Mongoose-Framework.

Aufgabe 5.1: HTTP-Server und WebSockets mit Node.js (2 Punkte)

Schreiben Sie auf Basis von node.js einen HTTP-Server, der die folgenden Funktionalitäten bietet:

a. Ermittlung der aktuellen Zeit: startet die URL eines HTTP-GET-Requests mit „`currenttime`“, so wird die aktuelle Uhrzeit ermittelt und es wird ein JSON-Dokument erzeugt, das die Sekunde, Minute, Stunde der aktuellen Zeit enthält.
b. Abbildung der statischen Struktur eines Verzeichnisses: wird in der URL ein Dateiname angegeben, so soll der Inhalt dieser Datei aus einem Verzeichnis zum Client transportiert werden. Achten Sie darauf, den korrekten MIME-Type (HTML, CSS, JS, JSON, JPG, sonst "text/plain") und den Response-Code (200, 404, 500) zu setzen.
c. Streaming des HTTP-Logs an interessierte Web-Clients: die Zugriffe der Teilaufgaben von a. und b. sollen in einem HTTP-Log protokolliert werden, das zum Beispiel folgenden Aussehen haben kann:

```
GET 200 /ordner/index.html
GET 404 /ordner/favicon.ico
GET 200 /ordner/icon.png
```

„Streamen“ Sie diesen Log über WebSockets an interessierte Clients, die sich zuvor beim Server über zu implementierende Javascript-Logik beim node.js-Server angemeldet haben. Zeigen Sie mit mindestens zwei Clients, dass der HTTP-Log (der auch Client-IP-Adresse, Timestamp sowie Größe der Datei enthalten kann) an alle Clients gesendet wird.

Aufgabe 5.2: Node.js und MongoDB (3 Punkte)

Installieren Sie den MongoDB-Node.js-Driver von der URL

```
https://github.com/mongodb/node-mongodb-native/
```

oder alternativ das folgende Node.js Modul:

```
https://github.com/mafintosh/mongojs
```

Implementieren Sie in Javascript eine Software, die eine Verbindung zu MongoDB mit der Collection „zips“ aus dem vorhergehenden Aufgabenblatt 4 herstellt.

Schreiben Sie dann auf Basis von node.js einen HTTP-Server, der auf URLs der Form

```
http://localhost:1234/zips/[city]
```

antwortet, wobei „[city]“ der Name einer Stadt ist, also beispielsweise

```
http://localhost:1234/zips/YOUNGSTOWN
```

Es sollen alle Dokumente aus der MongoDB-Collection ermittelt werden, die zu einer Stadt vorhanden sind. Das Ergebnis dieser MongoDB-Anfrage soll dann an den Webbrowser im JSON-Datenformat gesendet werden. Ist kein Dokument zu einer Stadt in der Collection vorhanden, soll im Webbrowser eine entsprechende Meldung angezeigt werden.

In der Benutzeroberfläche, die im Webbrowser dargestellt wird, soll der Name der Stadt in einem Eingabefeld definiert werden können. Der Webbrowser soll dann das Ergebnis einer Anfrage in einer Liste darstellen, wobei die Darstellung alle gespeicherte Informationen (zip, loc, etc.) zur definierten Stadt enthalten soll.

Benutzen Sie zur Implementierung der Browser-Funktionalität und der graphischen Benutzeroberfläche die von Polymer zur Verfügung gestellten Core-Elemente wie z.B. `<core-ajax>`, `<core-list>` sowie die Paper-Elemente wie z.B. `<paper-button>`, `<paper-input>` etc.

Aufgabe 5.3: Verwendung von Mongoose und Express für eine E-Mail-Anwendung (5 Punkte)

Aus den Lösungen für das Aufgabenblatt 4 besitzen Sie eine MongoDB-Collection, die eine Menge von E-Mail-Dokumenten enthält. Entwerfen Sie mit dem Mongoose-Framework ein Schema, das die E-Mail-Collection abbildet. Erstellen Sie dann mit dem Express-Framework eine REST-API, die die folgenden Funktionalitäten implementiert:

REST API	Beschreibung
Get folders	Ermittelt alle existierenden Folder
Get folder messages	Ermittelt alle Nachrichten eines Folders
Delete folder	Löscht einen existierenden Folder mit allen Nachrichten
Update folder name	Aktualisiert den Namen eines Folders
Read message	Liest die Detailinformationen einer einzelnen Nachricht
Delete message	Löscht eine einzelne Nachricht
Create message	Erzeugt eine neue einzelne Nachricht
Update message	Verschiebt eine Nachricht von einem Folder in einen anderen

Überlegen Sie, welche Eingabeparameter die jeweilige REST-API-Funktion benötigt und welche JSON-Dokumente als Resultat zurückgegeben werden sollen. Benutzen Sie Express und Mongoose zur Implementierung der geforderten REST-API-Funktionalität.