

# Übungsaufgaben (Blatt 03)

WPF „Web-Frameworks“

B. Sc. Medieninformatik

Fachbereich Medien

Prof. Manfred Wojciechowski

Node.js, MongoDB und restify		
	<p><u>Vorbereitung:</u></p> <ol style="list-style-type: none"> <li>1. Lade einen Node.js Installer herunter und installiere Node.js auf dem Rechner. Dafür muss zunächst folgende Seite aufgerufen werden: <a href="https://nodejs.org/de/download/">https://nodejs.org/de/download/</a></li> <li>2. Es existieren auf dieser Seite Installer für Windows und Mac Geräte wie auch Binaries für Geräte mit Linux als Betriebssystem um Node.js lokal zu installieren. Hier muss der passende Installer oder die passenden Binaries ausgewählt werden und heruntergeladen werden.</li> <li>3. Danach muss die Installation erfolgen. Zum Testen, ob Node.js erfolgreich installiert wurde muss der folgende Befehl ausgeführt werden: <b><u>node -v</u></b></li> <li>4. Die Skripte können über: <b><u>node {Skriptname}.js</u></b> aufgerufen werden.</li> </ol>	
1	Lernziele: Kennenlernen des Event-Loop. Umsetzen kleiner ereignisgesteuerter Tasks im Event-Loop.	
	a	Erstelle eine globale Funktion ,logWithTime(logText, counter)', welche den Text, den Counter, sowie die aktuelle Zeit in der Konsole ausgibt.
	b	Erstelle eine globale Funktion ,immediateFunction(counter), welche die Funktion ,logWithTime' aufruft und den Text ,Immediate', sowie den counter übergibt

	c	Erstelle eine globale Funktion ,timeOutFunction(counter), welche die Funktion ,logWithTime' aufruft und den Text ,Timeout', sowie den counter übergibt	
	d	Erstelle eine globale Funktion ,doHeavy', welche eine künstliche Last erzeugt und dadurch die Programmausführung um einige ms verzögert, z.B. eine große Zählschleife.	
	e	<p>Erstelle eine globale Funktion ,readFile(fileName, counter). Diese soll eine über ,fileName' angegebene Datei aus dem Dateisystem öffnen und in einem ReadStream auslesen. Ergänzt darin folgende Callback-Funktionen zum Reagieren auf folgende Events:</p> <ul style="list-style-type: none"> <li>- open: Rufe zunächst ,doHeavy' und dann die Funktion 'logWithTime' auf und übergebe den Text 'Open', sowie counter</li> <li>- data: Rufe zunächst ,doHeavy' und dann die Funktion ,logWithTime' auf und übergebe den Text 'Data', sowie counter. Bei jedem dritten Aufruf soll zusätzlich in der Phase ,check' die Funktion ,immediateFunction' mit Übergabe von counter aufgerufen werden.</li> <li>- end: Rufe zunächst ,doHeavy' und dann Funktion ,logWithTime' auf und übergebe den Text 'End', sowie counter.</li> <li>- close: Rufe zunächst ,doHeavy' und dann die Funktion ,logWithTime' auf und übergebe den Text 'Close', sowie counter.</li> </ul>	
	f	<p>Rufe im Skript nun in folgender Reihenfolge die folgenden Funktionen auf:</p> <ul style="list-style-type: none"> <li>- Übergabe der Funktion timeOutFunction(1) in den Event-Loop zur Ausführung in der Phase ,timers' nach 100ms</li> <li>- Übergabe der Funktion timeOutFunction(2) in den Event-Loop zur Ausführung in der Phase ,timers' nach 3s</li> <li>- Aufruf von readFile(&lt;fileName&gt;, 1);</li> <li>- Aufruf von readFile(&lt;fileName&gt;, 2);</li> </ul>	

		<ul style="list-style-type: none"> <li>- Aufruf von <code>readFile(&lt;fileName&gt;, 3);</code></li> <li>- Aufruf von <code>readFile(&lt;fileName&gt;, 4);</code></li> <li>- Aufruf von <code>readFile(&lt;fileName&gt;, 5);</code></li> </ul> <p>Als <code>&lt;fileName&gt;</code> soll der Pfad und Name einer auf dem Rechner existierenden Datei angegeben werden, welche ausgelesen werden kann.</p>	
2		<p>Lernziele: Umsetzen eines kleinen datenbankgestützten REST-Service mit Hilfe von Node.js, MongoDB und restify</p> <p>Installiere als Erstes restify:</p> <ol style="list-style-type: none"> <li>1. Lege ein Node Projekt im Projektordner an: <b><u>npm init</u></b></li> <li>2. Installiere restify: <b><u>npm i restify -save</u></b></li> </ol> <p>Installiere zunächst MongoDB und starte die Datenbank. Danach muss der Treiber für MongoDB in Node installiert werden. Nachfolgend wird eine empfohlene Anleitung genannt:</p> <ol style="list-style-type: none"> <li>1. Downloade den MongoDB Installer: <a href="https://www.mongodb.com/try/download/community">https://www.mongodb.com/try/download/community</a></li> <li>2. Nachdem die MongoDB installiert wurde, muss die Applikation MongoDB Compass aufgerufen werden und eine neue Verbindung über New Connection erzeugt werden.</li> <li>3. In MongoDB Compass muss auch eine Datenbank <b><u>„studentDb“</u></b> und Collection <b><u>„students“</u></b> angelegt werden, welche zusammen mit Restify verwendet werden.</li> <li>4. Im Node Projekt, wo Restify installiert wurde, installiere den MongoDB Treiber: <b><u>npm i mongodb -save</u></b></li> </ol>	
	a	<p>Erstelle einen REST-Service zum Verwalten von Studenten. Es sollen REST-Methoden mindestens zum Hinzufügen, Löschen und Auflisten von Studenten erstellt werden. Code bitte aufbewahren, da dieser Service eventuell in einer weiteren Übung verwendet wird.</p>	
	b	<p>Teste im Browser, ob der REST-Service korrekt funktioniert. Du kannst dafür eventuell ein REST-Plugin für deinen Browser verwenden, z.B. REST-Client für Firefox, oder Advanced REST-Client für Chrome.</p>	

