

Klausurvorbereitung und weiterführende Themen

Vorstellung in den Tutorien am 12.02. - 20.02.

Nach 13 Tutorien sind die Übungen offiziell für die Veranstaltung Webtechnologien beendet. Damit Sie sich gut auf die Abschlussprüfung vorbereiten können, gibt es im ersten Teil dieses Zusatzblatts ein paar freiwillige Aufgaben, die nicht mehr bepunktet werden. Diese Aufgaben sollten Sie mit dem erlangten Wissen der Veranstaltung bearbeiten können.

Generell ist jedoch alles, was in der Vorlesung besprochen wurde und in den Übungen als wichtig markiert war, prüfungsrelevant. Sie sollten sich deshalb auch mit den vergangenen Aufgaben und Vorlesungsinhalten auseinandersetzen.

Im zweiten Teil des Blattes bieten wir für interessierte Studierende weiterführende Themen an, die in dieser Lehrveranstaltung nicht besprochen wurden. Diese sind **nicht** prüfungsrelevant, jedoch sehr informativ, wenn Sie sich tiefer mit dem Bereich der Webtechnologien beschäftigen wollen. Sie bieten unter Umständen auch alternative Wege zur Lösung von Aufgaben, die vielleicht einfacher und/oder kürzer sind.

In den Wochen vor der Klausur finden am üblichen Termin (Di. 11-13 Uhr TEL 103) Sprechstunden statt. Ebenfalls finden Sprechstunden mittwochs 11-13 Uhr in MAR 4.063 statt.

Teil I

Klausurvorbereitung

1 CSS: Box-Modell und Farben

Welche Bereiche hat das Box-Modell? Mit welchen Eigenschaften geben Sie die Größen der verschiedenen Bereiche an? Welche der Größen-Eigenschaften funktionieren bei *Inline*-Boxen nicht? Mit welchen Eigenschaften können Sie die verschiedenen Bereiche einfärben?

2 CSS: Button stylen

Stylen Sie den folgenden Button, so dass die gegebenen Anforderungen erfüllt sind.

```
<button type="submit">Klick mich</button>
```

- eine weiße, fette Schrift, ohne Serifen, in der Größe 16px und einen Zeilenabstand von 30px haben
- einen Abstand zwischen Text und Rand von Links/Rechts 20px und Oben/Unten 10px haben
- einen blauen Hintergrund
- ein visuelles Feedback geben, wenn er geklickt wird. Dabei soll die Schriftfarbe blau werden und der Hintergrund weiß.

3 CSS, JS: Bildershow

Wenn auf heutigen Webseiten mehrere Bilder eingebettet sind, wird bei einem Klick das entsprechende Bild häufig im Vordergrund zentriert groß dargestellt. Durch einen Klick auf das Bild oder einen entsprechenden Button wird dann das nächste Bild geladen und so kann durch die Bildergalerie navigiert werden.

Stellen Sie dieses Verhalten mit CSS und JavaScript nach:

- Beim Klick auf das Bild soll dieses Element kopiert und im Viewport fest fixiert und zentriert dargestellt werden.
- An der linken und rechten Seite sollen Bereiche mit einem Pfeil nach Links bzw. Rechts auftauchen.
- Beim Klick auf diese Pfeile soll das nächste bzw. vorherige Bild geladen und im Vordergrund zentriert dargestellt werden.

4 JS: Conway's Game of Life

In dieser Aufgabe sollen Sie das Game of Life¹ programmieren. Bei dieser Simulation wird durch die Anwendung bestimmter Regeln eine Folgegeneration bestimmt.

Das Feld soll eine Größe von 10x10 haben und die Startgeneration eine zufällige Verteilung von 60% lebenden Zellen besitzen. (Sie können auch gerne andere, interessante Objekte ausprobieren und nachvollziehen, ob diese das beschriebene Verhalten aufweisen.) Der Rand soll durch tote Zellen belegt sein. Durchlaufen Sie eine variable Anzahl von Generationen, z.B. 15.

Die Ausgabe soll auf der Console von Chrome erfolgen oder auf einer Webseite. Es soll die Generationsnummer und die entsprechende Generation dargestellt werden. Zwischen den Ausgaben der Generationen soll 2 Sekunden gewartet werden.

Für ein besseres Verständnis des Spiels und eine sinnvolle Herangehensweise können Sie sich die ersten 15 Folien folgender Präsentation der FU Berlin durchlesen: http://page.mi.fu-berlin.de/block/pi3_2012/proinf_tag_4.pdf.

5 JS: Eigener Audio-Player

Hinweis: Sie brauchen für die Abschlussprüfung nicht die API der Medienelemente kennen, jedoch sollte Ihnen die Umsetzung mit der Dokumentation keine Probleme bereiten.

Mit JavaScript können Sie durch das DOM HTML-Elemente auswählen und diese manipulieren. Dabei besitzen einige Elemente eine eigene API, z.B. die Media-Elemente `<audio>` und `<video>`.

Da die standardmäßige Oberfläche des Audio-Players (Attribut: `controls`) nicht sonderlich umfangreich und browserabhängig ist, ist es Ihre Aufgabe einen eigenen Player zu erstellen. Folgende Anforderungen sollten erfüllt sein:

1. Es werden frei verwendbare Symbole benutzt.
2. Die standardmäßige Oberfläche (Attribut: `controls`) ist deaktiviert.
3. Es gibt eine Playlist mit mindestens 4 Musikdateien.
4. Wenn ein Titel abgespielt wurde, fängt automatisch der Nächste an.
5. Folgende Funktionalitäten müssen im Benutzer-Interface mindestens zur Verfügung stehen: Play, Pause, Lautstärkeregelung, Abspielgeschwindigkeit ändern, nächster Titel, vorheriger Titel, freie Titelauswahl durch Selektion, zufällige Reihenfolge der Titel.

Um den Symbolen Ihrer grafischen Oberfläche Funktionalität zu verleihen, schauen Sie sich die APIs der HTML-Medienelemente² und im speziellen des Audioplayers³ im HTML5-Standard an. Dort finden Sie alle benötigten Eigenschaften und Funktionen um Ihren Player mittels JavaScript zum Leben zu erwecken.

¹https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens

²<https://www.w3.org/TR/html/semantics-embedded-content.html#the-media-elements>

³<https://www.w3.org/TR/html/semantics-embedded-content.html#the-audio-element>

6 JS: Präsentation erstellen

In dieser Aufgabe geht es darum eine Präsentation (über ein beliebiges Thema) mit Hilfe von HTML, CSS und JavaScript zu erstellen. Erstellen Sie eine Webseite, bei der folgende Anforderungen erfüllt sind:

- Im HTML-Quelltext können `<article>`-Elemente eingestellt werden – diese entsprechen einer *Slide* der Präsentation.
- Die Slide hat eine Breite und Höhe von 80% des Viewports und wird horizontal zentriert dargestellt.
- Im HTML wird nur an einer Stelle ein `<footer>`-Element angelegt. Dieses erscheint als Footer auf allen Slides.
- Die Hintergrund-Farbe der kompletten Seite unterscheidet sich von dem Hintergrund der Slides.
- Es wird immer lediglich eine Slide zu einem Zeitpunkt angezeigt.
- Es gibt eine Navigationsleiste bestehend aus frei verwendbaren Icons. Die Navigation enthält mindestens folgende Funktionen und Informationen: eine Slide Vor, eine Slide Zurück, zum Anfang, zum Ende, Anzahl der Slides, aktuelle Slide-Nummer
- Die Navigation durch die Slides soll per Mausklick und Tastendruck erfolgen können.

Wissenswertes: Für diesen Use Case gibt es bereits Bibliotheken wie S5⁴, reveal.js⁵ und Impress.js⁶. In dieser Aufgabe sollten Sie allerdings keine dieser fertigen JS-Bibliotheken benutzen, sondern den Sachverhalt mit Hilfe von JavaScript auch selbst umsetzen können.

7 JS: JSON und AJAX

In dieser Aufgabe sollen Sie Vorschläge für Suchbegriffe liefern während der Benutzer etwas in das Suchfeld einträgt (sie kennen das von der Google-Suche). Erledigen Sie dazu folgende Punkte:

1. Erstellen Sie eine JSON-Datei (Dateiendung `.json`), die Sie in einen Ordner namens „Tiere“ verschieben. In diese Datei schreiben Sie im JSON-Datenformat mindestens 15 unterschiedliche Objekte. Jedes Objekt enthält als Eigenschaft die Kategorie eines Tieres, z.B. Hund, Katze, Vogel, etc. Wenn Sie wollen können Sie gerne weitere Eigenschaften dem Objekt hinzufügen und sich mit den in JSON verfügbaren Datentypen auseinandersetzen.

⁴<http://meyerweb.com/eric/tools/s5/>

⁵<http://lab.hakim.se/reveal-js/#/>

⁶<http://impress.github.io/impress.js/#/bored>

- Erstellen Sie eine Webseite mit einem Suchfeld. Hinterlegen Sie für das Suchfeld folgende Funktionalität: Bei **jedem** Buchstaben der eingegeben wird, soll ein AJAX-Request ausgelöst werden an ihre hinterlegte JSON-Datei. Durchlaufen Sie die Antwort nach allen Objekten, deren Kategorie mit dem eingegebenen Buchstaben übereinstimmt. Zeigen Sie die Vorschläge dem Benutzer an, indem Sie ein neues HTML-Element erzeugen, welches an das Suchfeld angeheftet ist. Dieser Block soll den unter dem Suchfeld stehenden Text nicht verschieben, sondern überlagern.

8 REST: API-Dokumentation

In Übung 13 sollten Sie mit Hilfe des grafischen Webeditors Restlet Studio⁷ eine API dokumentieren, welche die Schnittstelle zum Verwalten der Elemente der Themenseite (bzw. Restaurants) Ihrer Webseite darstellt. Erstellen Sie in dieser Aufgabe die REST-Schnittstelle mit den Fluchbuchungen aus dem Beispiel in den Vorlesungsfolien. Da dieser Editor noch keinen ordentlichen Hypermedia-Support beinhaltet, vernachlässigen Sie einfach Stufe 3 des RMM-Modells und fertigen die Beschreibung von Stufe 2 an (Folie 41 und 42).

9 REST: Hypermedia as the Engine of Application State

Hinweis: Sie müssen das Konzept Hypermedia für die Prüfung verstanden haben, jedoch nicht den Aufbau oder die Implementierung eines speziellen Repräsentationsformates beherrschen.

In den bisherigen Übungen sollten Sie oft eigene REST-basierte Schnittstellen auf der Serverseite zur Verfügung stellen. Dabei durften Sie der Einfachheit halber die dritte Schicht des Richardson Maturity Model „Hypermedia as the Engine of Application State“ außer Acht lassen. Jedoch behauptet selbst Roy Fielding, dass APIs nur RESTful sind, wenn dieser Aspekt beachtet ist. Lesen Sie den in ISIS hochgeladenen Artikel zu RESTful Hypermedia APIs. Verstehen Sie die grundlegende Idee hinter diesem Konzept und implementieren Sie eines der Repräsentationsformate in einer ihrer bereits erstellten REST-basierten Schnittstellen. Unter npmjs.com können Sie verschiedene Bibliotheken für NodeJS finden, die Ihnen bei der Entwicklung helfen.

10 JS Module

Um JavaScript-Code zu strukturieren, modularisieren und wiederzuverwenden können funktional zusammengehörende Bereiche des Quellcodes ausgelagert werden. Lesen Sie Kapitel 10 des Buches Eloquent JavaScript⁸ über Module. Erklären Sie anhand von Sourcecode-Beispielen:

- Namensräume in JavaScript
- das von Node.js verwendete Pattern *CommonJS*: die Benutzung und die generelle Implementierung einer `require`-Funktion in einer JS-Laufzeitumgebung
- den Nachteil des CommonJS-Pattern, wenn es im Browser genutzt wird
- worauf beim Interface-Design geachtet werden sollte

⁷<https://studio.restlet.com>

⁸http://eloquentjavascript.net/10_modules.html

11 Architektur-Beispiel einer Webanwendung

Unter <http://martinfowler.com/articles/orderPad/> finden Sie eine interessante Präsentation über die Architektur einer Webanwendung zur Bestellung von Waren für das Lager eines Supermarktes. Lesen Sie diese durch und verstehen Sie sie.

Hinweis: Die Serverseite enthält einige Java-spezifische Technologien, die nicht aus dieser Veranstaltung bekannt sind. Auch das im Artikel angesprochene Pattern namens Segregated DOM⁹, welches eine Strukturierung des Codes auf Client-Seite vornimmt und in ähnlicher Art oft wiederzufinden ist, ist nicht klausurrelevant.

12 Model-View-Controller Pattern

Hinweis: Sie müssen das MVC-Konzept für die Prüfung verstanden haben, jedoch nicht den Umgang mit den angesprochenen Tools beherrschen.

Erklären Sie das Model-View-Controller Pattern. Installieren Sie danach den Express-Generator¹⁰ und Yeoman mit generator-express¹¹. Erstellen Sie mit beiden Tools, wie in den *Getting started*-Anleitungen beschrieben, jeweils zwei MVC-Anwendungen (einen CSS Preprozessor oder eine Datenbank brauchen Sie bei Yeoman nicht auswählen). Vergleichen Sie nun die erstellten Strukturen der beiden Apps mit der auf den Folien vorgestellten MVC-Struktur.

Hinweis: Auf der Seite <http://yeoman.io/> finden Sie generelle Informationen über das Tool. Um die generierten JS-Projekte auszuführen, empfiehlt es sich auch die Tools bower¹² und grunt-cli¹³ mittels npm global zu installieren.

13 Notiz-Webseite

In dieser Aufgabe sollen Sie eine kleine Notiz-Webseite erstellen. Die Webseite, die Ihr Server ausliefert, soll lediglich ein Eingabefeld, einen Speichern-Button und einen Bereich für die Notizen besitzen. Mittels JavaScript wird eine kleine Funktionalität zur Verfügung gestellt: Wenn im Eingabefeld Text steht (die Notiz), dann wird dieser beim Klick auf den Speichern-Button gelöscht und als neue Notiz mit aktuellem Datum in den vorgesehenen Bereich eingefügt. Außerdem sollen die Daten per XMLHttpRequest-Objekt an den Server übermittelt werden. Folgende Abbildung zeigt einen möglichen Aufbau:

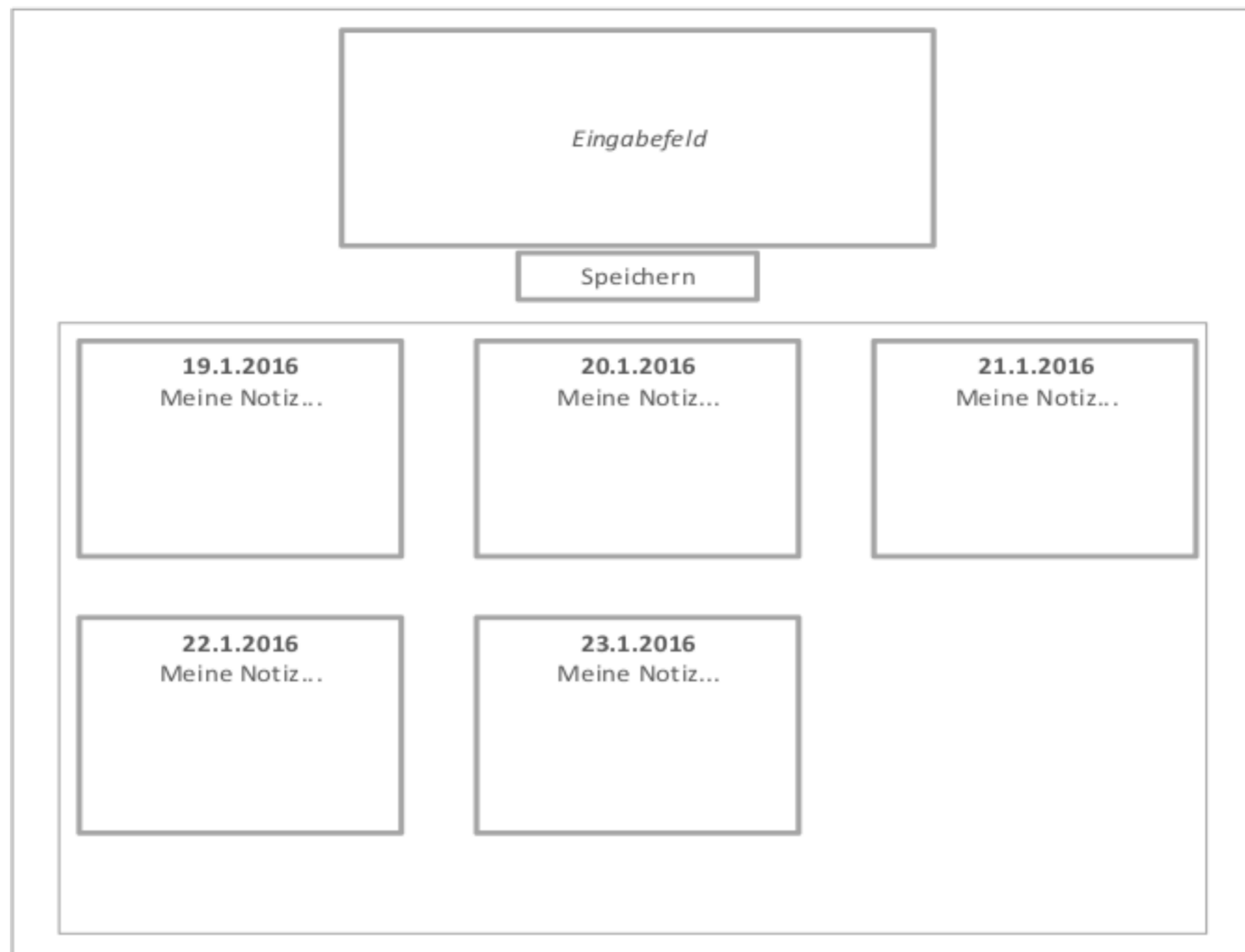
⁹<http://martinfowler.com/bliki/SegregatedDOM.html>

¹⁰<http://expressjs.com/en/starter/generator.html>

¹¹<https://github.com/petecoop/generator-express#getting-started>

¹²<http://bower.io/>

¹³<http://gruntjs.com/>



In dieser Aufgabe sollen Sie den Nutzer identifizieren, das heißt wiedererkennen, ob er die Seite schon einmal besucht hat, oder nicht. Erarbeiten Sie sich dazu erst den theoretischen Hintergrund: HTTP ist ein statusloses Protokoll. Mit Hilfe von Cookies können Clients wiedererkannt werden. Das wird u.a. dazu verwendet um mehrere HTTP-Anfragen, die zusammengehören, zu identifizieren und in einer sogenannten Sitzung bzw. Session zu verwalten. Erklären Sie, wie Cookies im HTTP-Protokoll gesetzt werden, welche Attribute sie besitzen und welche Arten es gibt.

Sie können für die Implementierung die Middleware `express-session`¹⁴ nutzen. Achten Sie bei der Konfiguration darauf, dass die Session auch nach einem Browserneustart wiederhergestellt werden soll.

14 Webseite zur Benutzung des Datei-Servers

Mit dem zuvor erstellten Datei-Server (Sourcecode¹⁵) ist es möglich beliebige Dateien mit richtigem *Content-Type*-Header auszuliefern. Erstellen Sie in dieser Aufgabe eine eigene Webseite mit CSS- und JavaScript-Datei, welche der Server auf Anfrage zurück gibt. Der Zweck dieser Webseite soll es sein ihren Datei-Server mittels einer grafischen Oberfläche zu bedienen.

Ihre Webanwendung soll die folgenden Anforderungen erfüllen:

- Die Webseite soll alle Funktionalitäten des Servers (Datei/Verzeichnis lesen, erstellen und löschen) abbilden. Dafür müssen für die jeweiligen Methoden entsprechende Buttons und Eingabefelder zur Verfügung stehen, z.B. um den Dateinamen und den Inhalt festzulegen/anzusehen.

¹⁴<https://github.com/expressjs/session>

¹⁵http://eloquentjavascript.net/code/file_server.js

- Die Interaktion zwischen der an den Browser ausgelieferten Webseite und dem Server soll in JavaScript mittels XMLHttpRequest-Objekt im Hintergrund erfolgen.
- Die URL ihrer Webanwendung ändert sich nie.
- Die gesamte Übertragung der Daten zwischen Server und Client soll in JSON formatiert sein. (Erfordert eine Änderung des Server-Codes.)
- Wenn ein Verzeichnis abgefragt wird, sollen alle Dateien gelistet werden und per Maus auswählbar sein. Dadurch wird eine bestimmte Datei selektiert und der Inhalt angezeigt.
- Strukturieren Sie Ihre Seite mittels CSS übersichtlich und benutzerfreundlich.

Hinweis: Achten Sie darauf, dass es nicht möglich sein soll die Dateien für Ihre „Basis“-Webseite zu verändern. Bei einem Reload könnte es ansonsten passieren, dass keine Webseite mehr ausgeliefert wird, weil Sie diese gelöscht haben.

15 Airline REST-API implementieren

In Aufgabe 8 sollten Sie eine Dokumentation für die REST API einer Fluggesellschaft (Beispiel aus den Folien) erstellen. Setzen Sie diese API mit Express um und befüllen Sie sie mit Pseudodaten. Beachten Sie auch die Pfad- und Query-Parameter (z.B. `/flights/{flightNr}/seats?availability=open`).

16 Fingerprinting mit Node.js oder Express

Die Wiedererkennung von Benutzern, die auf eine Webseite im Internet zugreifen, ist ein wichtiges Thema für viele Unternehmen und Geschäfte im Internet. Die einfachste Methode ist das Setzen von Cookies. Da jedoch viele Benutzer lieber anonym auf Webseiten zugreifen, werden die Cookies häufig nach dem Schließen des Browsers automatisch gelöscht. Dies macht eine Wiedererkennung schwieriger. Allerdings zeigen Veröffentlichungen zu diesem Thema, dass es eine Vielzahl an anderen Techniken gibt, einen Benutzer zu verfolgen. Häufig reicht es bereits aus die Headerfelder der HTTP-Kommunikation auszuwerten, die ein Browser automatisch bei jeder Anfrage setzt. Sie können das selbst für ihre Browser testen: `http://ip-check.info/?lang=de`.

Schreiben Sie ein Node.js-Modul oder eine Express Middleware, welches Sie nutzen können um jeden Besucher ihrer Webseite anhand der HTTP-Felder *User-Agent*, *Accept* und *Accept-Language* zu identifizieren. Speichern Sie diese drei Informationen in eine Liste und setzen Sie einen Zähler pro Eintrag hoch, wenn exakt die selben Informationen bei einem neuen Request noch einmal gesendet werden.

Visualisieren Sie das Ergebnis in einer Tabelle auf einer Webseite, welche bei einer Abfrage zurückgeliefert wird. Als Überschrift der Tabelle schreiben Sie entweder „Hallo, schön Sie wiederzusehen“, wenn die Daten des Request bereits in der Liste enthalten sind, oder „Hallo, schön Sie kennenzulernen“, wenn der Benutzer neu ist.

Teil II

Weiterführende Themen der Webtechnologien

17 Content Negotiation

REST besagt, dass jede Ressource durch unterschiedliche Repräsentationen dargestellt werden kann. In HTTP ist dieses Prinzip mittels *Content Negotiation*¹⁶ umgesetzt.

1. Entwickeln Sie einen Webserver in Node.js, der Ihnen ihre Personendaten (Name, Jobbezeichnung, E-Mail) bei einem HTTP-Request mit spezifizierten *Accept*-Header in dem angegebenen Format zurück gibt. Mindestens unterstützt werden muss `text/plain`, `text/html` und `application/json`. (Als Beispiel können Sie die Ressource `http://eloquentjavascript.net/author` mit Postman nach unterschiedlichen Repräsentationen abfragen. Ihr Webserver soll die selbe Funktionalität bieten.)
2. Schreiben Sie durch Verwendung von Node's `http.request` Funktion einen Client, der Ihren zuvor geschriebenen Server anspricht und Ihre Personendaten in jedem möglichen Format abfragt. Die Antworten sollen auf der Konsole ausgegeben werden.

18 E-Mail mit Aktivierungslink

In dieser Aufgabe sollen Sie einem Benutzer eine E-Mail mit einem Aktivierungslink zuschicken, so wie Sie das im Internet häufig bei Registrierungen für Online-Dienste erleben. Erst nach dem Klick auf den Link kann festgestellt werden, ob die E-Mail-Adresse valide ist und tatsächlich dem Benutzer gehört.

Gehen Sie folgendermaßen vor: Erstellen Sie eine Webseite für die Registrierung von Nutzern. Die Seite muss mindestens den Namen und die E-Mail abfragen. Diese Daten werden an Ihren Server übermittelt, welcher daraufhin eine Aktivierungsmail an den Registrierten verschickt. Nutzen Sie für die Implementierung das Node.js-Modul `Nodemailer`¹⁷. Die Verwendung ist auf der Internetseite beschrieben. Zum Senden sollten Sie einen Mail-Server verwenden, auf dem Sie bereits ein Konto besitzen, z.B. dem der TUB. Hier funktioniert folgende Konfiguration für `Nodemailer 2.7.2`:

```
var transporterObjectConfig = {
  host: 'exchange.tu-berlin.de',
  port: '587',
  secure: false,
  requireTLS: true,
  auth: {
    user: '<anmelde-name>@win.tu-berlin.de',
    pass: '<pass>'
  }
};
```

¹⁶https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation

¹⁷<https://community.nodemailer.com/>

Ersetzen Sie `<anmelde-name>` und `<pass>` mit Ihren TUB-Daten und Sie können in `Node.js` Emails mit Ihrem Account verschicken.

Hinweis: In Cloud9 funktioniert die Verwendung von *nodemailer* derzeit nicht. Arbeiten Sie deshalb für diese Aufgabe lokal auf ihren eigenen Computer.

Übermitteln Sie in der Mail einen sehr langen, nicht zu erratenden und für jeden Registrierer einen anderen Link, z.B.

`http://localhost:8080/regs/861safd64fa412135dsaf463da5s43f...`

Diese URL kann dann für 5 min an ihrem Server angesprochen werden und wird danach ungültig. Wenn der Benutzer im gültigen Zeitraum darauf zugreift, dann wird eine Erfolgsbestätigungs-Webseite angezeigt, andernfalls eine Fehlerbenachrichtigung mit der Möglichkeit zur erneuten Registrierung.

Zusatzfrage: Welche einfache Möglichkeit fällt Ihnen ein (bei Verwendung einer HTML-Mail) zu erkennen, ob der Empfänger bereits die Mail gelesen hat, wenn Sie einen im Internet ansprechbaren Server besitzen?

19 REST: unterschiedlicher Payload

Verschiedene Clients einer API haben häufig unterschiedliche Anforderungen, was es sinnvoll macht unterschiedliche Payloads je Client anzubieten. Zum Beispiel kann der benötigte Payload abhängig von der grafischen Oberfläche und dessen Anwendungsfluss sein. Ein weiteres Kriterium kann der gewünschte Detaillierungsgrad sein, da eine Webseite mehr Informationen darstellen kann als eine Smartphone-App. Gleiches gilt für die möglichen Aktionen. Auch können physikalische Eigenschaften eine Rolle spielen: so ist es für einen Smartphone-Client aufgrund seiner unterbrechbaren Verbindungen zum Internet sinnvoll, ihn nicht zu zwingen mehrere Requests durchzuführen, sondern eine aggregierte Antwort aus mehreren Ressourcen mit einmal zurückzusenden.

Überlegen und diskutieren Sie Möglichkeiten, wie eine API unterschiedliche Payloads für unterschiedliche Clients anbieten könnte!

20 Web Components

Informieren Sie sich, was Web Components sind. Wieso sinnvoll? Welche Hauptfeatures gibt es? Wie erstellt man eine Component? Mögliche Quellen sind u.a. <https://www.webcomponents.org/introduction> und <https://www.polymer-project.org/>

21 ROCA

Ein Ansatz zum Design von Frontend-basierten Webanwendungen aufbauend auf den REST-Prinzipien ist der Stil *Resource-Oriented Client Architecture* (ROCA). Was sind die Architektur-Empfehlungen der ROCA-Webseite¹⁸? Was bedeuten die Begriffe *unobstrusive JavaScript* und *progressive Enhancement*?

¹⁸<http://roca-style.org/>

22 Performante mobile-first Webseiten erstellen mit AMP

Das *Accelerated Mobile Pages*¹⁹ Projekt (AMP) ist ein Ansatz von Google zur Erstellung von performanten Webseiten, insbesondere im Hinblick auf die Benutzung auf mobilen Geräten. Lesen Sie sich den technischen Performance-Artikel²⁰ und die Spezifikation²¹ durch. Bearbeiten Sie außerdem das Quick-Start Tutorial²². Wie will das Projekt den Performance-Gewinn bei (mobilen) Webseiten erreichen? Was sind Vor- und Nachteile dieses Ansatzes?

23 Progressive Web Apps

Neben dem AMP-Projekt promoted Google derzeit auch seine *Progressive Web Apps* (PWA)²³ – eine Möglichkeit Webanwendungen zu erstellen, die der Benutzer bei Wunsch als App auf dem Smartphone installieren kann. Um das native Offline-Gefühl von Smartphone Apps zu erreichen, nutzen PWAs die in HTML5 eingeführten Service Worker²⁴. Zum Starten, lesen Sie sich den Quick Start Guide durch: <https://codelabs.developers.google.com/codelabs/your-first-pwapp/#0>

24 JavaScript Style Guide

Wenn Sie öfter in JavaScript programmieren bzw. kooperativ an einem Projekt arbeiten, empfiehlt es sich eine Konvention einzuhalten. Unter folgender URL sind einige Empfehlungen aufgelistet: <https://github.com/rwaldron/idiomatic.js/>

25 JS Frontend Framework

Im Bereich der Webseitenentwicklung mit JavaScript kann die Erstellung eines etwas größeren Projektes leicht sehr komplex werden. Unter anderem hilft hier natürlich die Auslagerung von Code in Module und die Verwendung von spezialisierten Bibliotheken. Jedoch gibt es auch große JavaScript-Frameworks, die dem Entwickler bei vielen Aspekten behilflich sind. Die bekanntesten sind derzeit wahrscheinlich Angular²⁵ von Google, React²⁶ von Facebook und Ember.js²⁷.

Vergleichen Sie die Vorteile und Herangehensweisen der Frameworks gegeneinander, indem Sie die Quick-Start-Tutorials durcharbeiten und externe Vergleiche suchen. Außerdem ist

¹⁹<https://www.ampproject.org/>

²⁰https://www.ampproject.org/docs/get_started/technical_overview.html

²¹<https://www.ampproject.org/docs/reference/spec.html>

²²https://www.ampproject.org/docs/get_started/create_page.html

²³<https://developers.google.com/web/progressive-web-apps/>

²⁴<https://developers.google.com/web/fundamentals/getting-started/primers/service-workers>

²⁵<https://angular.io/>

²⁶<https://facebook.github.io/react/>

²⁷<http://emberjs.com/>

die folgende Artikel über die Entwicklung der wichtigsten JS-Frameworks im Jahr 2017 sehr interessant: <http://developer.telerik.com/topics/web-development/javascript-2017-libraries-frameworks/>

26 Domain Name System

Beantworten Sie die folgenden Fragen und bearbeiten Sie die Aufgaben. Nutzen Sie dafür das native DNS Lookup Tool ihres Betriebssystems (`nslookup`, `dig` oder `host`) um die DNS-Einträge auf der Kommandozeile abzufragen. (Ein Blick in die Manuals ist meistens für die Bedienung ausreichend. Bei `nslookup` sollten Sie mittels `set debug` die Debug-Option aktivieren, da Sie hierdurch ausführlichere Informationen erhalten.)

1. Auf welcher Schicht des OSI-Referenzmodells befindet sich das DNS-System und welches Transport-Protokoll wird benutzt?
2. Was ist der Unterschied zwischen einer Domäne und einer Zone?
3. Mittels welchem Resource Record Typ werden in einem Nameserver DNS-Namen einer IP zugewiesen? Wie wird auf den zuständigen Nameserver für eine Sub-Domain verwiesen? Was sind Aliase und wie werden Sie im Resource Record ausgewiesen? Wie sieht ein Mail-Server-Eintrag aus? Was gibt der Typ SOA an?
4. Überprüfen Sie mit Ihrem Kommandozeilenprogramm die Domäne `www.snet.tu-berlin.de`. Was für ein Resource Record ist hinterlegt? Wie findet ihr Programm die zugehörige IP-Adresse?
5. Finden Sie von `snet.tu-berlin.de` den zuständigen Mailserver und den Nameserver heraus. Welcher Zone ist diese Domäne deshalb unterstellt?
6. Fragen Sie von `yahoo.com` die IP-Adresse ab (IPv4 ist ausreichend). Wieso werden Ihnen mehrere IP-Adressen zurückgegeben? Welche würde ihr Browser verwenden, wenn dieser die Abfrage getätigt hätte?
7. Was ist ungefähr der Preis, den Sie für einen nicht-vergebenen Domain-Namen unter den TLD `.de`, `.com`, `.org`, `.bayern` und `.berlin` bezahlen müssten?
8. (*Zusatz-Aufgabe für die Experten:* Sie können den standardmäßig verwendeten Nameserver in Ihrem Betriebssystem konfigurieren. Wieso ist es aus Performance-Sicht oft nicht so sinnvoll dies zu machen?)

Hinweis: In den Kommandozeilenprogrammen gibt es die Möglichkeit den abzufragenden Typ festzulegen. Meist ist auch eine Abfrage nach `ANY` möglich, welche normalerweise alle Einträge einer Zonendatei zurückliefern sollte. Dies ist jedoch anscheinend nicht bei allen Nameservern freigeschaltet. Es bleibt die Möglichkeit den gewünschten Typ direkt abzufragen.

27 HTTP/2.0

Nach 15 Jahren ohne Weiterentwicklung wurde 2015 eine Nachfolgeversion des HTTP-Standards veröffentlicht. Lesen Sie Kapitel 12²⁸ des lesenswerten Buches *High Performance Browser Networking*²⁹ durch. Auch die Spezifikationen von HTTP/2.0 sind gut lesbar und verständlich geschrieben. Es lohnt sich deshalb immer ein Blick in die RFCs 7540³⁰ und 7541³¹.

28 REST API Design Guides

Das Erstellen einer REST API für einen Dienst kann relativ aufwändig werden. Insbesondere gibt es unterschiedliche Ansätze, wie eine gute RESTful-API umgesetzt wird. Zum Beispiel ist häufig eine der ersten Fragen: Wie und wo wird eine API versioniert? Hier gibt es unterschiedliche Lösungen: in der URL, im Header oder eventuell gar nicht (lediglich über Hypermedia). Viele Firmen erstellen solche Design Guidelines. Unter <http://apistylebook.com/> sind einige zusammengetragen und analysiert.

29 GraphQL

Facebook entwickelte 2012 eine etwas andere Art um Daten von ihrem Dienst abzufragen. 2015 stellten Sie die *GraphQL* genannte Abfragesprache Open Source und mittlerweile bieten einige großen Dienste ihre APIs damit an. Diese Abfrageart könnte eine spannende Zukunft haben, weshalb sich ein Blick auf die Webseite lohnt: <http://graphql.org/>

30 APIs: Flickr-API mittels JSONP benutzen

Der Bilderdienst Flickr bietet unter anderem eine API an, die mittels der JSONP-Technik benutzt werden kann. Schreiben Sie eine Webseite, in der Sie die neusten Bilder abrufen und darstellen. Jedes Bild soll zwei Bereichen (bei Flickr als „Tags“ bezeichnet, z.B. „Berlin“ und „Sport“) zugeordnet sein. Welche Bereiche das sind, lassen Sie den Nutzer ihrer Seite über eine Eingabe bestimmen. Von allen angezeigten Bildern wird zusätzlich der Titel und Autor dargestellt. Wenn auf das Bild geklickt wird, öffnet sich in einem neuen Fenster die Original-Seite von Flickr.

Sie können wie folgt vorgehen: Lesen Sie zuerst die VL-Unterlagen und starten Sie dann mit einer neuen Webseite. Nachdem der Benutzer ihrer Seite die gewünschten Themenbereiche eingegeben hat, wird eine Funktion aufgerufen, in der ein neues `script`-Element erstellt wird. (*Achtung:* Da Sie diesen Vorgang mehrmals durchführen, sollten Sie das alte `script`-Tag zuerst im DOM löschen!) Benutzen Sie für das `src`-Attribut die URL https://api.flickr.com/services/feeds/photos_public.gne?format=json&jsoncallback=callbackFunction

²⁸<https://hpbn.co/http2/>

²⁹<https://hpbn.co/>

³⁰<https://tools.ietf.org/html/rfc7540>

³¹<https://tools.ietf.org/html/rfc7541>

, die Sie um weitere Parameter ergänzen können. Welche es gibt, ist auf dieser Flickr-Seite³² aufgelistet. Verändern Sie außerdem den Wert der Callback-Funktion und übergeben Sie Ihre eigene. Diese sollte genau einen Parameter entgegennehmen - nämlich ein JSON-Objekt, welches Flickr an diese Callback-Funktion übergibt. Zuletzt muss in der Funktion das neue Script-Element im DOM eingefügt werden.

Analysieren Sie das übergebene JSON-Objekt (ein Blick auf die URL genügt um die erforderlichen Properties des Objektes zu finden). Danach ist der Aufbau Ihrer Callback-Funktion ganz einfach: in einer Schleife das komplette Objekt durchgehen und Bild, Titel bzw. Autor als HTML-Element erstellen. Das Bild wird zusätzlich noch auf die Original-Flickr-Seite verlinkt. Dann müssen Sie ihre Elemente nur an der richtigen Stelle im DOM einbinden und Sie sind fertig.

Hinweis: Wenn es gewünscht ist, finden Sie auf folgender Seite Informationen, wie Sie die URL verändern müssen um das Bild in einer anderen Größe abzugreifen: <https://www.flickr.com/services/api/misc.urls.html>

31 JSONP API anbieten

Express bietet eine Funktion mit der Sie sehr einfach eine JSONP-Schnittstelle anbieten können. In dieser Aufgabe sollen Sie darauf aufbauend eine API zur Verfügung stellen, die Daten über eine spezifizierte Person zurückgibt. Gehen Sie folgt vor:

1. Erstellen Sie eine Datenstruktur für `Personen` und instanziiieren Sie mindestens 5 Objekte.
2. Erstellen Sie eine Route, in der ein Pfad-Parameter für eine ID einer Person enthalten ist.
3. Liefern Sie in der Route mit der Funktion `res.jsonp`³³ die Antwort im JSONP-Format zurück.
4. Schreiben Sie eine Webseite, die die Personendaten mittels JSONP von ihrem Server abrufen und in die Seite integriert

Beispiel Anfrage:

```
http://localhost:8080/person/1?callback=writeOnSite
```

Beispiel Antwort:

```
writeOnSite( {  
  "fullname": "Kai Grunert",  
  "mail": "grunert@snet.tu-berlin.de" } )
```

³²https://www.flickr.com/services/feeds/docs/photos_public/

³³<http://expressjs.com/en/4x/api.html#res.jsonp>

32 Überblick Datenbanken

Lesen Sie sich die dreiteilige Artikel-Serie über die NoSQL-Geschichte durch, welche Sie in ISIS finden. Achten Sie u.a. auf Codd's Kritik der damaligen Systeme, das relationale Datenmodell, Normalformen, referentielle Integrität, ACID, die Schwächen relationaler DB, das CAP-Theorem, Arten und Eigenschaften von NoSQL Datenbanken, Schwächen von NoSQL Datenbanken, Multi-Modell-Datenbanken, Kritik an der Implementierung von relationalen Datenbanken und die Neuerungen von NewSQL Datenbanken. Fassen Sie die aktuellen Ausrichtungen zusammen, indem Sie einen kleinen Leitfaden mit Vor- und Nachteilen über den Einsatz bestimmter Datenbanken für bestimmte Anwendungsfälle geben.

33 Websicherheit

In der Vorlesung wurde aus zeitlichen Gründen das Thema Sicherheit von Webanwendungen nicht angesprochen. Es ist ein sehr umfangreicher Bereich. Trotzdem ist es als Webentwickler wichtig sich damit zu beschäftigen. Auf den Seiten des Open Web Application Security Project³⁴ finden Sie viele Materialien, Projekte, Bücher, Videos und Guides zu diesem Thema. Auch ein Blick in den folgenden Artikel³⁵ ist sehr interessant. Bei der Programmierung von Express-Anwendungen lohnt sich ein Blick auf die Middleware Helmet³⁶, die einige sicherheitskritische Aspekte behandelt.

34 HTTPS

Wenn eine Kommunikation im Internet im Klartext stattfindet, setzen sie sich damit verschiedener Gefahren aus. Es vereinfacht aber auch das massenhafte Verletzen der Privatsphäre durch Regierungen und Geheimdienste. Nicht zuletzt findet deswegen aktuell bei vielen Internetdiensten ein Umdenken statt, standardmäßig auf eine verschlüsselte Kommunikation zu setzen. Mittlerweile werden Webseiten mit einer solchen Eigenschaft sogar in Suchmaschinen besser platziert als ohne.

Lesen Sie sich die Vorlesungsfolien zu HTTPS durch und beantworten Sie folgende Fragen: Welche Sicherheitsfunktionen werden mit diesem Protokoll-Stack erreicht? Nennen Sie ein paar Beispiele für Gefahren, die dadurch verhindert werden können.

Was ist eine *asymmetrische* Verschlüsselung? Was sind digitale *Signaturen* und wieso sind sie sinnvoll? Wie funktionieren digitale *Zertifikate* und welches Problem lösen sie?

35 Browser Erweiterungen

Im Laufe des Semesters haben Sie eine Menge Technologien kennengelernt, die im Frontend und im Backend zum Einsatz kommen. Mit Ihnen sind umfangreiche Dinge möglich, die nicht nur

³⁴<https://www.owasp.org>

³⁵<http://martinfowler.com/articles/web-security-basics.html>

³⁶<https://github.com/helmetjs/helmet>

positiv für den Benutzer sind: die webseitenübergreifende Verfolgung von Nutzern ermöglicht gezielt Werbung zu schalten, erfordert aber den Verzicht auf Privatsphäre. JavaScript macht eine Webseite oft erst benutzbar, kann aber ein Sicherheitsrisiko darstellen. Suchen Sie sich Erweiterungen für ihren Browser, die diese Aspekte verbessern/verhindern/lösen, damit Sie beim täglichen Surfen im Internet ein besseres Gefühl haben.

Aspekte die beachtet werden könnten:

- Ausführung von JavaScript-Dateien, welche in einer Seite eingebunden sind
- Kontrolle über die Cookies, die eine Seite oder eine verlinkte Seite (Drittanbieter) setzt
- Daten im Referrer-Headers
- Daten im User-Agent
- Blockierung von Werbe- bzw. Privatsphäre-verletzenden-Skripten
- Obligatorische Benutzung von sicherer HTTPS-Kommunikation