



JogLog

Sarah M., Leo Liebig,

Idee

- Laufdaten-Tracker
- Tagebuch für Laufdaten
- Daten können eingetragen und anschließend in Form mehrerer Diagramme angesehen werden
- Profil anlegen
- Mögliche Erweiterungen
 - Freunde für Läufe einladen
 - sich mit Freunden vergleichen

Frontend

Sarah Melzer





Tools

- jQuery
 - Umfangreiche JS-Library für diverse DOM-Funktionalitäten
 - nur 32 kB

```
$('#logSelection').change(function () {  
    var selection = $('#logSelection option:selected').text();  
    getNextDiagram(selection);  
    changeDiagrams();  
});  
  
$nav.find('a[href=addrun]').click(function () {  
    $addrun.slideToggle('slow', function() {  
        $addrun.find('input[type=text]').val('');  
    });  
    return false;  
});
```

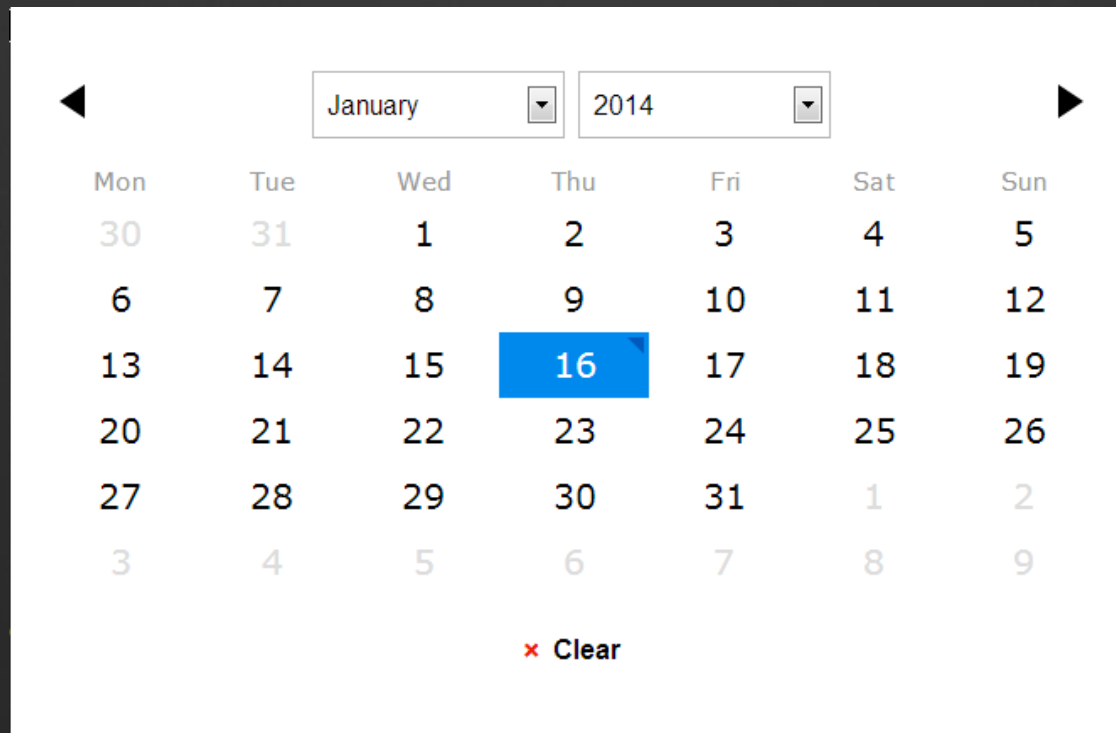
Tools

- pickdate.js
 - für Zeit- und Datumsauswahl
 - Mobiltauglich (responsive + lightweight)

```
$('.datepicker').pickadate({  
  min: new Date(2013,3,20),  
  max: new Date(2013,7,14),  
  firstDay: 1,  
  selectYears: 4,  
  weekdaysShort: ['Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa']  
});
```

Tools

- pickdate.js
 - Zur Selektion von Geburts- und Laufdatum



Tools

- jQuery SVG
 - JS-Library zum Erstellen und Manipulieren von SVGs
 - Für Diagrammzeichnung genutzt

```
this.wrapper.svg({settings: { width: this.width, height: this.height}});
this.svg = this.wrapper.svg('get');
this.svg.text(this.width/2, this.height/2, 'This is a text',
  {fontFamily: 'Verdana', fontSize: '15', fill: 'white'});
this.svg.line(group, x_start, y_start, x_end, y_end,
  { stroke: '#5c9302', strokeWidth: 2 });
this.svg.circle(group, x_pos, y_pos, radius,
  {fill: '#75bb03', stroke: '#558900', strokeWidth: 1});
```

Diagram Module

- JS-Modul zum Anzeigen der Laufdiagramme
- Monatliche oder jährliche Ansicht möglich
- Passt sich dynamisch an Diagrammgröße an

```
caloriesDiagram = diagram.init({  
  wrapper : "#caloriesDiagram",  
  data: orderedData,  
  unit: 'kcal',  
  name: 'calories'  
});
```

Ausblick

- Vollständiges **Responsive Design** für Mobil
- **Druckansicht** optimieren
- **XML-Export** von Laufdaten
- Diagrammgröße im Browser ändern
- Zeit über einen Schieberegler im Diagramm strecken / stauchen
- Ladeanimationen
- Design verbessern

Backend

Leo Liebig

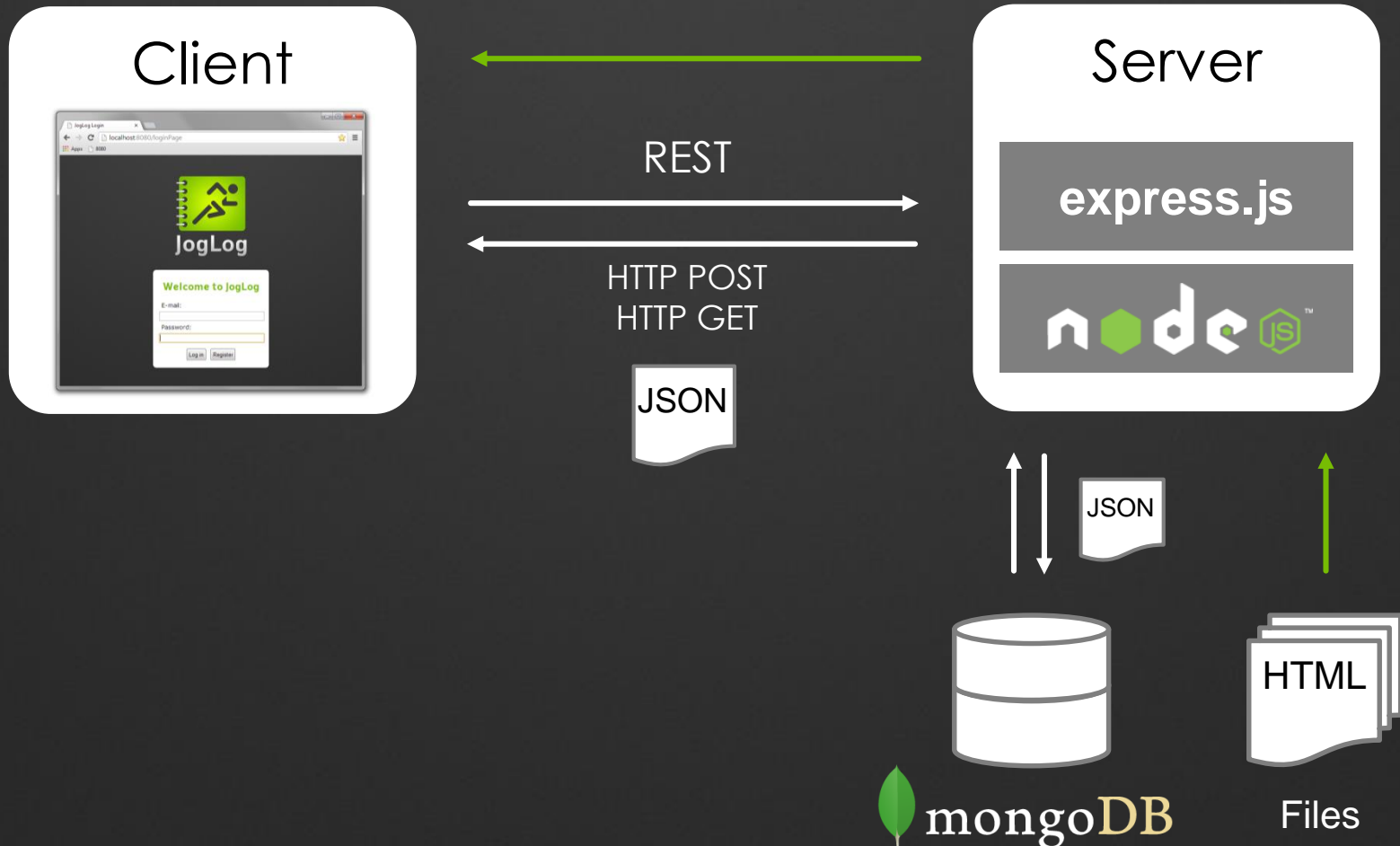
Aufgaben und Funktionalität

- Auslieferung statischer Inhalte
 - HTML, CSS, JS und Grafiken
- Nutzerverwaltung
 - Nutzer hinzufügen, speichern, editieren
 - Authentifizierung
 - Sessions

Aufgaben und Funktionalität

- Laufdatenverwaltung
 - Läufe hinzufügen, speichern und editieren
 - Berechnung zusätzlicher Laufdaten je Lauf
 - verbrauchte Kalorien
 - Durchschnittsgeschwindigkeit (km/h)
 - Zeit pro Kilometer (min/km)
 - Aggregation von Läufen eines Jahres (Jahresübersicht)

Aufbau



Node.js

- Laufzeitumgebung für Netzwerkanwendungen
- Programmierung in JavaScript
- nutzt Googles V8 Java Script Engine

- ereignisgesteuerte Architektur
- nicht-blockierender Event-Loop (I/O asynchron)
- ressourcenschonend, skalierbar, einfache Verwendung

Express

- „Web application framework for node“
- Abstraktionsschicht zur Vereinfachung von Node.js Anwendungen

- Vereinfachtes Routing
- Sessions
- Cookie und HTTP Request Parser

Node.js + Express

```
1 var express = require('express');
2 var routes = require("./routes");
3 var app = express();
4
5 var PORT = 8080;
6
7 //configure middleware
8 app.use(app.router);
9 app.use(express.static(__dirname + "/static"));
10
11 /* ----- STATIC CONTENT DELIVERY ----- */
12 //sends registration.html
13 app.get("/registrationPage", routes.registrationPage);
14
15 /* ----- REST API ----- */
16 //POST: creates a new user or updates an existing one
17 app.post("/user", routes.addUser);
18
19 //GET: returns the user's profile data
20 app.get("/user", routes.restrict, routes.getUserData);
21
22 /* ----- START SERVER ----- */
23 app.listen(PORT);
24 console.log('Server started at port ' + PORT);
```

MongoDB

- nicht-relationale Datenbank
- Daten werden in Dokumenten gespeichert
- Datenstruktur: JSON
- schema-los

- Collections: Container für Dokumente mit semantischem Zusammenhang
- JogLog: users, runs

MongoDB

```
> db.runs.findOne()
{
  "day" : 5,
  "month" : 1,
  "year" : 2012,
  "distance" : 21.44,
  "duration" : 107.2,
  "mood" : "Good",
  "userId" : "52a9d203260dbb306e000001",
  "speed" : 12,
  "pace" : 5,
  "calories" : 1608,
  "_id" : ObjectId("52d2e822703636a81f000001")
}
```

Node.js + Express + MongoDB

- erprobtes Dreamteam
- viele Tutorials → einfacher Einstieg
- wenig Konfigurationsaufwand
- Fokus auf Geschäftslogik:

File	Code-Zeilen
■ server.js	68
■ routes.js	446
■ db.js	249

Ausblick

- Sicherheit
 - HTTPS
 - Hash und Salting von Passwörtern
 - Serverseitiger Parameter-Check
 - Persistieren der Sessions
- Funktionen
 - Profil ändern
 - Runs ändern

Quellen

- <http://amsul.ca/pickadate.js/index.htm>
- <http://jquery.com/>
- <http://keith-wood.name/svg.html>

- <http://www.mongodb.org/>
- <http://www.nodejs.org/>
- <http://expressjs.com/>