SwimFit

Ein Traingsanalyse Tool für Schwimmer

Meilenstein 4

EIS

Entwicklung Interaktiver Systeme

ausgearbeitet

von

Franziska Schmidt

Inhaltsverzeichnis

1.0 Datenstruktur..................................................................................... S. 3

2.0 WBA Modellierung............................................................................. S. 4

**1.0 Datenstruktur**

Es wurde sich für das Datenformat Javascript Object Notation (JSON) entschieden. Die Daten werden in einer MongoDB Datenbank abgespeichert. "Die Grundlagen der Speicherung von Informationen in dieser Datenbank basiert auf Dokumenten, die im BSON- Format vorliegen"[[1]](#footnote-1). Die Datenbank wird nochmal untergliedert in collections in denen einzelne Dokumente hinterlegt sind. Im folgenden werden die verschiedenen Collections mit einem Beispiel Dokument dargestellt.

1.1 Collection: Schwimmer

{

"\_id" : ObjectId("4dcd3ebc9278000000005158"),

"date" : ISODate("2011-05-13T14:22:46.777Z"),

"name" : "Sophie Müller",

"geburtstag" : "26.06.1991",

"Kenntnisse" : "Kurs 1",

"Verein" : "Campussport",

"Ruhepuls" : "90",

"Belastungspuls" : "175"

}

**2.0 WBA Modellierung**

Alle Ressourcen der REST API werden mit dem anwendbaren HTTP Methoden wie folgt beschrieben:

/schwimmer

GET: Die Schwimmer-Ressource mit der eindeutigen ID {schwimmer} wird im Response gesendet.

POST: Eine neue Schwimmer-Ressource wird unter eindeutiger, automatisch generierter ID in der Datenbank gespeichert.

PUT: Die Schwimmer-Ressource mit eindeutigen ID {schwimmer} wird die im Request enthaltenen Informationen in der Datenbank aktualisiert.

DELETE: Die Schwimmer-Ressource wird aus der Datenbank gelöscht.

/trainer

GET: Die Trainer-Ressource mit der eindeutigen ID {trainer} wird im Response gesendet.

POST: Eine neue Trainer-Ressource wird unter eindeutiger, automatisch generierter ID in der Datenbank gespeichert.

PUT: Die Trainer-Ressource mit eindeutigen ID {trainer} wird die im Request enthaltenen Informationen in der Datenbank aktualisiert.

DELETE: Die Trainer-Ressource wird aus der Datenbank gelöscht.

/training

GET: Die Listenressource aller Trainings-Ressourcen wird im Response gesendet

/modul

GET: Die modul-Ressource mit der eindeutigen ID {modul} wird im Response gesendet.

POST: Eine neue modul-Ressource wird unter eindeutiger, automatisch generierter ID in der Datenbank gespeichert.

PUT: Die modul-Ressource mit eindeutigen ID {modul} wird die im Request enthaltenen Informationen in der Datenbank aktualisiert.

DELETE: Die modul-Ressource wird aus der Datenbank gelöscht.

/clubs

GET: Die Listenressource aller Club-Ressourcen wird im Response gesendet

/club

GET: Die club-Ressource mit der eindeutigen ID {modul} wird im Response gesendet.

POST: Eine neue club-Ressource wird unter eindeutiger, automatisch generierter ID in der Datenbank gespeichert.

PUT: Die club-Ressource mit eindeutigen ID {modul} wird die im Request enthaltenen Informationen in der Datenbank aktualisiert.

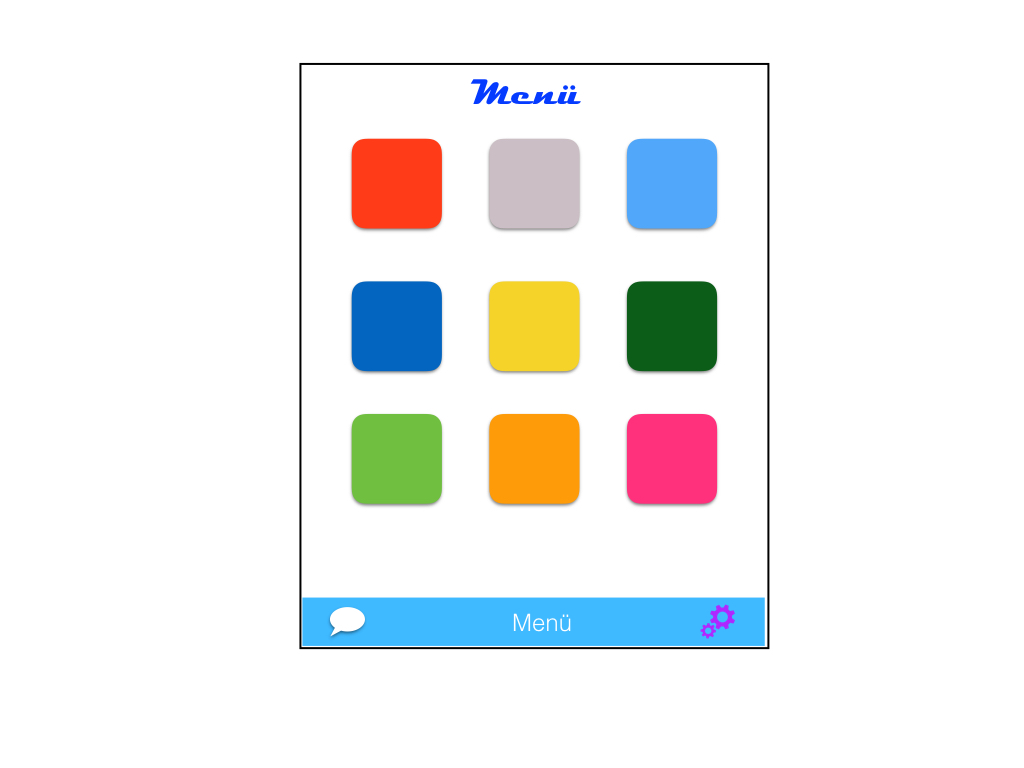
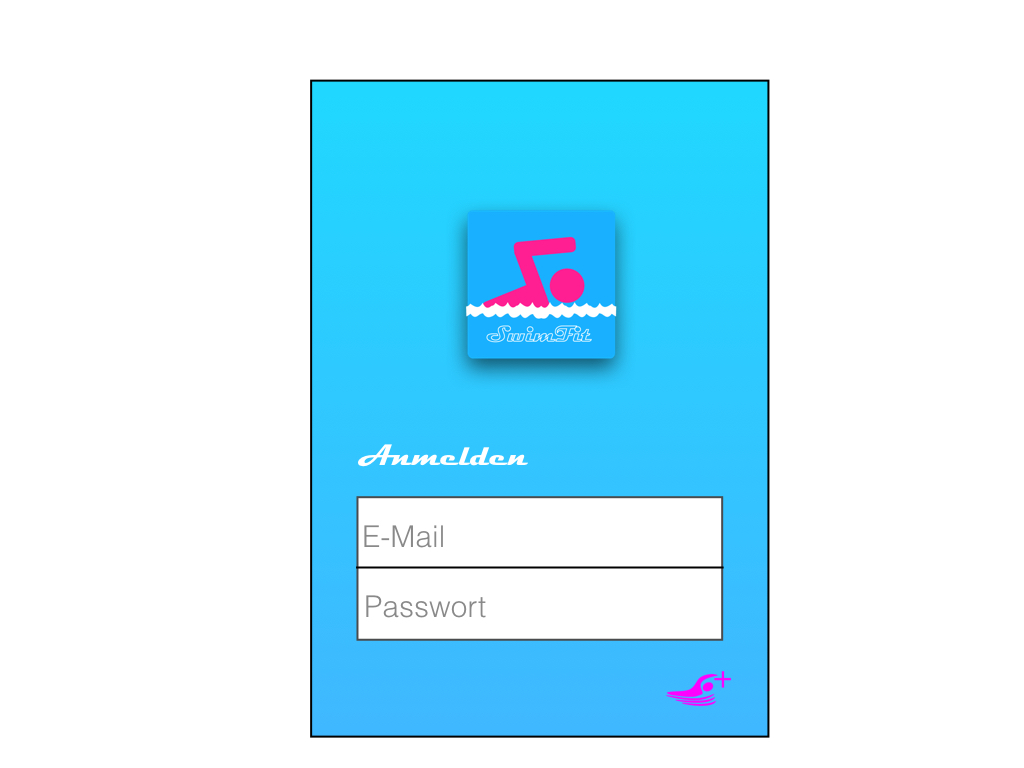
DELETE: Die club-Ressource wird aus der Datenbank gelöscht.

/competition

GET: Die Listenressource aller competition-Ressourcen wird im Response gesendet

/equipment

GET: Die Listenressource aller equipment-Ressourcen wird im Response gesendet



1. Stephan Mattescheck, Erik Lipperts : Node.js S.375 [↑](#footnote-ref-1)