



Hochschule  
Albstadt-Sigmaringen

Albstadt-Sigmaringen University

# Klausurvorbereitung



Dipl. Ing. Sven Eppler (FH)  
sodge IT GmbH

# Wie wird die Klausur aussehen?

- Mischung aus praktischen und theoretischen Fragen
- Theoretische Fragen mit Freitext-Antworten (kein Multiple-Choice)
  - Keine langen Texte, eher 1-2 Sätze notwendig
- Praktische Aufgaben in Form von Lücken-Texten
  - Nicht seitenweise Quellcode, eher 5-10 Zeilen pro Aufgabe
- Vorgegebener Quellcode und Sie müssen den erwarteten Output beschreiben

# Themen die abgefragt werden könnten

- HTTP Requests/Responses, Headers
- Sessions (Client- und Server-Side)
- Injections (HTML, SQL, JSON)
- WebFrameworks
  - Middlewares, Dispatcher/Router, Conf-over-Conv/Conv-over-Conf
- REST APIs
  - URL-Design, Konzept, Umsetzung in Node.js
- AJAX / Asynchrone Programmierung
  - Lambdas, Promises, Closures
- WebSockets
- Microservices

# Themen die ausgeklammert werden

- In meinem Teil kein HTML/CSS

- OpenBook Klausur

# Unterschiede zwischen StuPos

- Es gibt zwei Klausuren je nach StuPo
  - Eine Klausur nur mit Fragen von Herrn Eppler
  - Eine Klausur mit Fragen von Herrn Eppler und Herr Prof. Nemirovski
- Studenten der alten StuPo bearbeiten die „gemischte“ Klausur
- Studenten nach neuer StuPo bearbeiten die „reine“ Klausur

# Mögliche Aufgaben

- Eine REST-API soll Hunde verwalten. Dabei soll es möglich sein, die Hunde aufzulisten, Details zu einem Hund anzuzeigen, einen Hund zu löschen und einen neuen Hund anzulegen. Des Weiteren soll es möglich sein, einem Hund einen anderen Hund als Freund hinzuzufügen und alle Freunde aufzulisten. Beschreiben Sie ein REST-API Design mit den verwendeten URLs, URL-Parametern und HTTP-Methoden

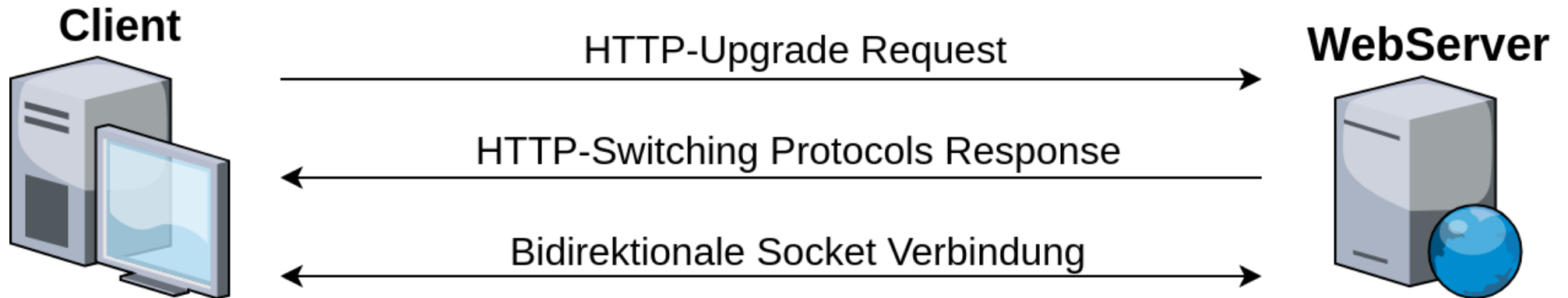
# Mögliche Lösung

- GET /hunde/  
Liefert eine Liste aller Hunde
- POST /hunde/  
Legt einen neuen Hund an
- GET /hunde/:hund\_id/  
Liefert Details eines Hundes
- DELETE /hunde/:hund\_id  
Löscht einen Hund
- GET /hunde/:hund\_id/friends  
Listet alle Freunde eines Hundes
- POST /hunde/:hund\_id/friends  
Fügt einen Freund hinzu



# Mögliche Fragen

- Skizzieren Sie einen WebSockets Handshake zwischen Client und Server



# Mögliche Fragen

- Beschreiben Sie was der jeweilige HTTP-Status Code bedeutet:
  - 404: ...
  - 200: ...
  - 500: ...

# Mögliche Fragen

- Erweitern Sie den Beispielcode so, dass ein HTTP-GET Aufruf den Text „Hallo“ mit dem Content-Type „text/plain“ zurück gibt.

```
router.get('/test', (req, res) => {  
  
});
```

```
router.get('/test', (req, res) => {  
  res.type("text/plain");  
  res.send("Hallo");  
});
```

# Ist noch etwas unklar?

- Noch weitere Fragen zur Klausur?
- Konkrete Fragen zum behandelnden Inhalt?