Web-Applikationen

Datenbanken

↘ Manuel Richardt, Dominik Rupprecht

↘ 04.07.23

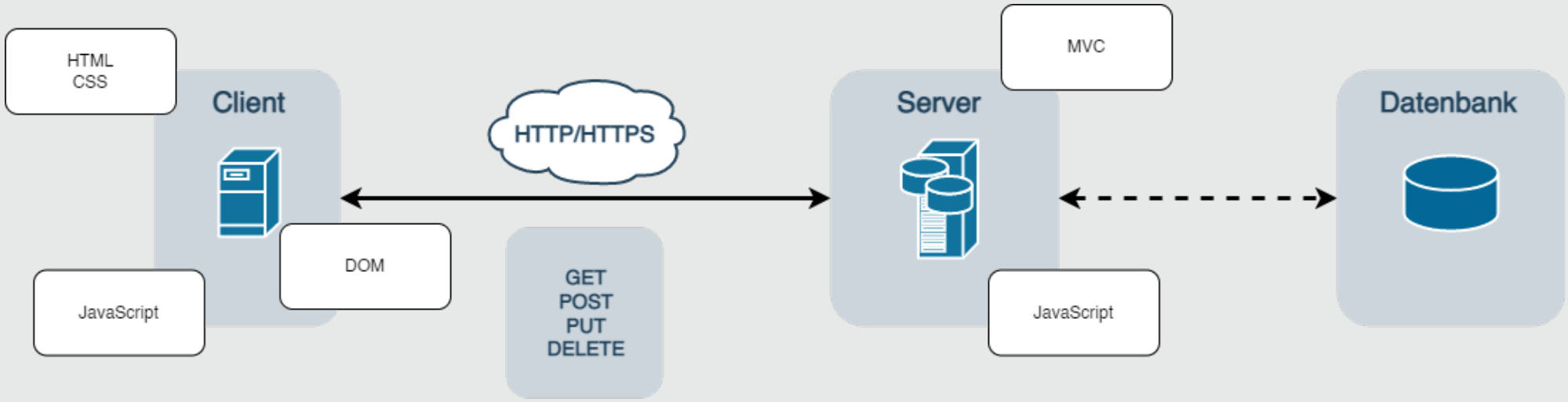
Hochschule Fulda
University of Applied Sciences

Fragen

AI

↘Rückfragen zur SU/Übung?

JavaScript im Server und im Client



Datenbanken

AI

↘ SQL (Relationale Datenbanken)
- ↘ Tabellenbasierte Schemas
- ↘ Skalieren Vertikal
- ↘ Oracle, SQL Server, MySQL, ...

↘ Vereinfachte Erläuterung:
- ↘ Datenbanken sind Sammlung von Tabellen
- ↘ Tabelle bildet “Model“ ab
- ↘ „Spalten“ haben Bezeichner und Datentyp

↘ Normalisierung von Tabellen (Aufteilung) wird im entsprechenden Modul gelehrt

Tabellen

AI

Schema

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

Beispiel

```
CREATE TABLE Persons (
    ID int PRIMARY KEY,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

Einfache SQL Statements

AI

↘ SELECT * FROM Persons

↘ SELECT * FROM Persons WHERE ID=1

↘ SELECT LastName FROM Persons WHERE ID=1

↘ INSET INTO Persons (ID, LastName, Firstname, Address, City)
Richardt, Manuel, Leipzigerstr. 123, Fulda)

↘ UPDATE Persons SET LastName=Rupprecht WHERE ID=2

```
CREATE TABLE Persons (
    ID int PRIMARY KEY,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

Object Relational Mapping

AI

↘ORM steht für objekt-relationales Mapping, bei dem Objekte verwendet werden, um die Programmiersprache mit den Datenbanksystemen zu verbinden, wobei die Möglichkeit besteht, mit SQL und objektorientierten Programmierkonzepten zu arbeiten. ORM kann auf jeder Art von Datenbankmanagementsystem implementiert werden, bei dem die Abbildung von Objekten auf Tabellen im virtuellen System möglich ist.

Sprache im Vergleich

AI

Klassisches SQL

↘SELECT * FROM users

↘CREATE TABLE users (
username varchar(255),
birthday (varchar (255)
);

ORM

↘const users = await User.findAll();

↘const Users = sequelize.define('Users', {
username: DataTypes.STRING,
birthday: DataTypes.DATE,
});

Installation und Konfiguration über Kommandozeile

AI

- Installieren von Sequelize und SQLite3
  - `npm install sequelize sqlite3`
- Installieren von Sequelize-CLI (Kommandozeilen-Tool)
  - `npm install --save-dev sequelize-cli`
  - `sequelize init`
  - `npx sequelize-cli model:generate --name User --attributes firstName:string,lastName:string,email:string`
  - `sequelize-cli db:migrate`

Einbinden und nutzen

AI

↘app.js

```
var db = require('./models/index');

db.sequelize.sync({force: true}).then(() => {
  console.log("Database is ready");
});
```

↘userController.js

```
const db = require('../models/index');

async function getAllUsers(req,res,next) {
  await tmp(); //Erstellt tmp-Benutzer
  const users = await db.User.findAll();
  res.status(200).json(users);
}
```