Dokumentation zum 15.01.2018

1. Datenbanken:

- Datenbankmanagementsystem (DBMS)

- besteht aus einer oder mehreren Datenbanken

- diese wiederum bestehen aus Tabellen

- diese wiederum bestehen aus Attributen (Spalten) und Datensätzen(Zeilen)

- Relationale Datenbank

- Wenn eine Datenbank auf Daten einer anderen Datenbank zugreift

- Private und Fremde Schlüssel

- Privater Schlüssel: eindeutig, nur einmal vorhanden

- Fremder Schlüseel: Abstammung des privaten Schlüssel

- Shemenbehaftet:

- Aussehen der Datenbank wie Attribute angeordnet sind

- Die Attribute sind fix angeordnet

- Shemenlos:

- Attribute können dynamisch vergeben werden

2. Wiederholung REST Server

- Bodyparser

- JSON Daten werden in Objekte umgewandelt

- urlencoded:

- Informationen die urlencoded sind werden in ihr ursprungszeichen umgewandelt

- URL

- Fragment

- wird vom Client verarbeitet

um zB an eine bestimmte Stelle in der Seite zu springen

- Querry

- besteht aus Attributen

- Diese geben gnauere Informationen zur Ressource an

- Datenbank:

- Singelton Klasse:

- es kann nur ein Objekt von dieser Klasse erstellt werden

- iterable Objekt in javascript

- ähnlich wie Collections in Java (Map,Liste,Feld)

- Map:

- Schlüsselwertpaar: Schlüssel werden zur Authentifizierung der Einträge verwendet. Schlüssel können Strings sein

aber auch Objekte.

- Warum gibt es keine Collections in Javascript

- Felder sind dynamisch

- Maps werden mit Objekte realisiert

- Die Elemente werden dann als Attribut einfach dem Objekt hinzugefügt

3. Datenbank installieren

- MongoDB

-

REST-Server

Allgemein:

* Maschine-zu-Maschine Kommunikation
* Zustandsloses Protokoll
  + Zwischen den Nachrichten, die alle Informationen für Server bzw. Client enthalten, werden keine Zustandsinformationen gespeichert (weder Server noch Anwendung)
  + Bei einer Anfrage einer Ressource wird immer die gleiche Antwort zurückgesendet solange die Ressource nicht geändert wird
* Kodiert keine Methodeninformation in den URL
* URL gibt den Ort und Namen der Ressource an, nicht aber die Funktionalität die der Web-Dienst zu der Ressource anbietet
* Vorteile:
  + Infrastruktur ist im WWW schon vorhanden
  + Eine Ressource kann über verschieden Medientypen dargestellt werden
* Einheitliche Schnittstelle
  + Adressierbarkeit von Ressourcen
  + Repräsentationen zur Veränderung von Ressourcen
  + Selbstschreibende Nachrichten
  + Hypermedia as the Engine of Application State (HATEOAS)
* Mehrschichtige Systeme
  + Dadurch reicht es dem Anwender nur eine Schnittstelle anzubieten
  + Dahinterliegende Ebenen bleiben verborgen somit wird die Architektur einfacher
* Code on Demand
  + Optional
  + Client Code kann zu rlokalen Ausführung übertragen werden
  + z.B. JavaScript-Code bei HTML-Repräsentation

Die Bezeichnung „Representational State Transfer“ soll den Übergang vom aktuellen Zustand zum nächsten Zustand (state) einer Applikation verbildlichen. Dieser Zustandsübergang erfolgt durch den Transfer der Daten, die den nächsten Zustand repräsentieren.

*Aus <*[*https://de.wikipedia.org/wiki/Representational\_State\_Transfer*](https://de.wikipedia.org/wiki/Representational_State_Transfer)*>*

* Operationen

|  |  |  |
| --- | --- | --- |
| **Operation** | **Oracle description** | **German translation** |
| **DELETE** | Delete an existing resource. | Löscht die angegebene Ressource. |
| **GET** | Retrieve a representation of a resource. | Fordert die angegebene Ressource vom Server an. GET weist keine Nebeneffekte auf. Der Zustand am Server wird nicht verändert, weshalb GET als *sicher* bezeichnet wird. |
| **POST** | Create a new resource to an existing URL | Fügt eine neue (Sub-)Ressource unterhalb der angegebenen Ressource ein. Da die neue Ressource noch keinen URI besitzt, adressiert der URI die übergeordnete Ressource. Als Ergebnis wird der neue Ressourcenlink dem Client zurückgegeben. POST kann im weiteren Sinne auch dazu verwendet werden, Operationen abzubilden, die von keiner anderen Methode abgedeckt werden. |
| **PUT** | Create a new resource to a new URL, or modify an existing resource to an existing URL. | Die angegebene Ressource wird angelegt. Wenn die Ressource bereits existiert, wird sie geändert. |

* HATEOAS
  + Client navigiert über eine REST-Schnittstelle ausschließlich über URLs die vom Server bereitgestellt werden
  + Bereitstellung über:
    - Href,src-Attributen bei HTML-Dokumenten
    - JSON- bzw. XML-Attributen bei Schnittstellen
* Authentifizierungsthema
  + Wenn auf dem Lokalen Host ein Server lauft und z.B. eine Java-Applikation auf diese Daten von Lokalen Host zugreift
  + Dieser Server hat keine Verbindung zum Internet