To do:

* Rest:
* Eine Tabelle
  + Verben,
  + URI

- Ressourcendefinition

- Modell

Zustände der Ressourcen

* Domänenmodell Iteration:

Listenressource vs Primärressource

**Notes:**

* Ressourcen immer Substantiv im Plural
* Primärressourcen und Subressourcen sind vercshachtelt, und stehen miteinander meist in 1:n oder 1:1 beziehungen.

D.h. das die Ressourcen in einer Abhängigeitsbeziehung stehen. Meist können die Subressourcen nicht ohne die Primärressource sinnvoll bestehen.

Beispiel: Ein Nutzer hat mehrere Adressen. Wir sind hier mit zwei Ressourcen konfrontiert: Der Ressource “Nutzer” und der Ressource “Adressen”. Die Ressourcen “Adresse” kann nicht sinnvoll ohne eine Ressource Nutzer bestehen, zu der Sie zugeordnet ist, und ist daher nutzlos ohne diese. Daher ist sie die Subressource.

Soweit, so klar?

* Repräsentation einer Ressource, ist sozusagen eine Instanz von ihr. Also etwa ein Textfile, in der alle Infos zum Nutzer “Moritz” stehen, ist eine Representation der Ressource User/Moritz.

1. Was bieten wir denn eigentlich an?

Was war denn eigentlich nochmal Rest? ->Buzzwords raussuchen, und sammeln für morgen.

Notizen zu Vidoe von Fischer:

Zweite Modellierung? Wenn Zeit und Lust, warum denn auch nicht?

Beim Beispiel Order:

Okay: Der Befehel “Post” also für … geht auf die LISTENRESSOURCE! von Order, weil es ja noch keine spezifische, individuelle Order gibt. Die wird ja gerade erst erstellt.

Stichpunkte:

* Minimale Informationen via REST herausgeben.
* REST Methodik mlchte, dass Ressourcen fast immer im Plural geschrieben werden.
* Ein kleines Glossar anlegen, mit den wichtigsten Punkten von REST.
* - Listenressourcen sind ansammlungen von Ressourcen. Etwa alle Autos von James Bond.

Ressourcen Definitionen:

Benutzer, Listenressource -> Alle Nutzer (listenressource)

[http://fitundschoen.de/benutzer/](http://tolleapp.de/benutzer/all)

->Benutzer\_id -> Individueller Nutzer, Primärressource von Körperdaten.

[http://fitundschoen.de/benutzer/1234](http://tolleapp.de/benutzer/1234) -> individueller Nutzer,

Körperdaten Primärressource; Subressource von Benutzer\_id (Die Körperdaten sind an eine bestimmte Benutzer\_id gebunden.) (Größe, Geschlecht und Gewicht sind an einen Satz Körperdaten gebunden.)

-> Größe, Subressource

-> Geschlecht, Subressource

-> Gewicht, Subressource

[http://fitundschoen.de/benutzer/1234](http://tolleapp.de/benutzer/1234)/kpd/ -> alle Körperdaten dieses Nutzers

[http://fitundschoen.de/benutzer/1234](http://tolleapp.de/benutzer/1234)/kpd/groeße -> Größe dieses Nutzers

[http://fitundschoen.de/benutzer/1234](http://tolleapp.de/benutzer/1234)/kpd/geschlecht -> Geschlecht dieses Nutzers

[http://fitundschoen.de/benutzer/1234](http://tolleapp.de/benutzer/1234)/kpd/gewicht -> Gewicht dieses Nutzers

Alle Rezeptlisten:

Rezeptkategorien, Listenressource: -> Beinhaltet alle Rezeptekategorien.

[http://fitundschoen.de/](http://fitundschoen.de//all)[rezeptkategorie](http://tolleapp.de/rezepte/Kartoffelsuppe/all)[/](http://fitundschoen.de//all)

Rezeptkategorie\_id, Listenressource: -> Beinhaltet alle Rezepte eines bestimmten Typus.

[http://fitundschoen.de/rezeptkategorie/](http://tolleapp.de/rezepte/Kartoffelsuppe/all)kartoffelsuppe

Rezept\_id -> individuelles Rezept.

http://fitundschoen.de/kartoffelsuppe/42 -> kartoffelsuppen Rezept mit ID 42.

<https://www.dpunkt.de/common/leseproben/12326/4_Kapitel%208.pdf>

Fehler:

* ~~Representation~~
* ~~Erste Seite: conditional weg.~~
* ~~Geschlecht noch abändern Körpergröße.~~
* ~~Aktikivitätslevel~~
* ~~Rezeptkategorien~~