

WBA-Modellierungen

Anwendungslogik:

- Server:
 - Speichern, bearbeiten und löschen von Daten
 - Rezepte mit Lebensmittelbestand abgleichen !!
 - Filtern von Rezepten

Der Server ist als Dienstanbieter dafür zuständig die Daten dem Dienstanbieter auf Abruf bereitzustellen. Hierbei handelt es sich in diesem Fall um einen Dienstanbieter nach Restkriterien. Es liegt somit eine Client-Server Architektur vor.

Hierbei sind verschiedene Prinzipien gefordert, die Rest-Architektur erfüllen muss. Somit sollte z.B. eine Zustandslosigkeit herrschen, die eine Geschlossenheit bei jeder anfrage an den Server beschreibt. D.h. der Server muss aus den Informationen der Rest-Nachricht die Anfrage bearbeiten und somit verstehen können. Außerdem sollte jede Ressource nur durch eine URI angesprochen werden können und die Veränderungen einer Ressource sollten nur durch die Repräsentationen der Ressource, wie z.B. ein JSON-Objekt, möglich sein.

Um die Ressourcen genauer zu beschreiben, wurde eine Restspezifikation erstellt, die die einzelnen Ressourcen den URLs genau zuweist.

Fehlerzuweisung

Um die verschiedenen Fehler anzeigen zu können, werden HTTP-Statuscodes verwendet.

Code	Fehlerbeschreibung
2xx	Auftrag wurde erfolgreich ausgeführt
4xx	Anfrage konnte nicht bearbeitet werden, z.B. durch übergabe von falschen Parametern
5xx	Der Server kann nicht erreicht werden

Rest Spezifikation

Ressource	Methode	Semantik	Content-type(req)	Content-type(res)
/user	POST	Anlegen eines Users	application/json	application/json
/user/:id	PUT	Update eines Users	application/json	application/json
/user/:id	GET	Abrufen eines Users	-	application/json
/user/:id	DELETE	Löschen eines Users	-	text
/users	GET	Abrufen aller User	-	application/json
/user/:id/wochenplan	POST	Anlegen eines Wochenplans	application/json	application/json
/user/:id/wochenplan	GET	Abrufen des Wochenplans	-	application/json

/user/:id/ wochenplan	PUT	Update des Wochenplans	application/json	application/json
/user/:id/ einkaufszettel	POST	Anlegen eines Einkaufszettels	application/json	application/json
/user/:id/ einkaufszettel	GET	Abrufen des Einkaufszettels	-	application/json
/user/:id/ einkaufszettel	PUT	Update des Einkaufszettels	application/json	application/json
/user/:id/ einkaufszettel	DELETE	Löschen eines Einkaufszettels	-	text
/user/:id/ lebensmittel bestand	POST	Anlegen eines Lebensmittelbestandes	application/json	application/json
/user/:id/ lebensmittel bestand	GET	Abrufen des Lebensmittel- bestandes	-	application/json
/user/:id/ lebensmittel bestand	PUT	Update des Lebensmittel- bestandes	application/json	application/json
/user/:id/ lebensmittel bestand	DELETE	Löschen des Lebensmittel- bestandes	-	text
/rezept	POST	Hinzufügen eines Rezepts	application/json	application/json
/rezept/:id	GET	Abrufen eines Rezeptes	-	application/json
/rezept/:id	PUT	Update eines Rezepts	application/json	application/json
/rezept/:id	DELETE	Löschen eines Rezeptes	-	text
rezept/:id/bild	POST	Hinzufügen von Bildern eines Rezeptes	application/json	application/json
rezept/:id/bild	GET	Abrufen von Bildern eines Rezeptes	-	application/json
rezept/:id/bild	DELETE	Löschen eines Bildes von einem Rezept	-	text
/lebensmittel	POST	Hinzufügen von Lebensmittels	application/json	application/json
/lebensmittel/:id	GET	Update von Lebensmittels	-	application/json

Ressource: /rezepte

Verb: GET

Queryelemt	Beschreibung
? like={Lebensmittelname} + {Lebensmittelname2}+...	Hier werden die Lebensmittel angegeben, die vom Benutzer erwünscht sind.
&dislike={Lebensmitteln ame}+ {Lebensmittelname2}...	Hier werden die Lebensmittel angegeben, die vom Benutzer nicht erwünscht sind.
&kategorie={kategorie}	Hier wird die Kategorie angegeben. Es wird unterschieden zwischen "normal" oder "vegetarisch".
&schwierigkeitsgrad={sc hwierigkeitsgrad}	Hier wird der Schwierigkeitsgrad angegeben. Es gibt drei schwierigkeitsgrade : "leicht", "mittel" und "schwer".
&maxDauer={maximale Dauer}	Hier wird die maximale Dauer angegeben, die das Rezept maximal dauern darf.

Um eine korrekte Ausgabe zu erhalten, müssen alle Queryelemente aufgeführt sein. Hier ein Beispiel:

<http://localhost:8080/rezepte?>

like=Butter+Ei&dislike=Erbsen+Emmentaler&kategorie=normal&schwierigkeitsgrad=leicht&maxDauer=30

Eine weitere Queryabfrag ist hier noch möglich.

? reste={Lebensmittelname 1}+ {Lebensmittelname2}+...	Hier werden die Lebensmittelreste angegeben, die Reihenfolge ist hier wichtig, da sie mit der Menge verglichen werden.
&menge={Lebensmittelm enge1}+ {Lebensmittelmenge2}+...	Hier wird die Menge der oben aufgeführten Lebensmittel

	angegeben. Die Reihenfolge ist deshalb wichtig.
--	---

Auch hier wird wieder beide Elemente aufgeführt sein, da sonst die Abfrage nicht funktioniert. Hier ein Beispiel:
<http://localhost:8080/rezepte?reste=Butter+Ei+Kartoffeln+Schinken&menge=200+3+500+20>

Als Rückgabe wird eine Array mit den Rezepten gegeben.

- Client (PC)
 - Anzeige und bearbeiten des Wochenplans, Einkaufszettels, Rezept (Menge), Lebensmittelbestand, Benutzerprofil
 - Auswahl von Rezepten

- Client(Android)
 - Anzeige und bearbeiten des Einkaufszettels
 - Anzeige des Lebensmittelbestands
 - Anzeige des Wochenplans