

Allgemeine Informationen

Dokumentation für eine REST-API auf Basis des JDK's mit persistenter Speicherung der Datenbank in einer JSON-Datei. Der REST-Server verwaltet eine Datenbank, die eine Liste von Büchern ist.

Diese REST-API wurde für den „Contrôle Continue 2“ in Java im Jahr 2021-2022 programmiert.

Ressourcen

Diese API verwaltet Büchern, die mit eine ISBN abgefragt werden können.

Jedes Buch enthält eine eindeutige ISBN als Schlüssel, ein Titel und der Autor des Buches. Man darf nicht zwei Büchern mit derselbe ISBN haben. Die ISBN muss positiv und eindeutig sein. Der Titel und der Autor dürfen nicht leer sein.

Beispielsweise wird ein Buch in JSON-Format gespeichert als:

```
{  
  "isbn" : 12,  
  "titel" : "Silo",  
  "autor" : "Hugh Howey"  
}
```

Beispielsweise wird eine Buchliste in JSON-Format gespeichert als:

```
[  
  { "isbn" : 11, "titel" : "L'Empire des loups", "autor" : "Jean-Christophe Grange" },  
  { "isbn" : 12, "titel" : "Silo", "autor" : "Hugh Howey" }  
]
```

URLs

URL	Bezeichnung
/buchliste	Alle Bücher der Liste/Datenbank
/buchliste/{isbn}	Der Inhalt des Buches mit eingegebene ISBN
/buchliste/{autor}	Alle Bücher des eingegebenen Autors

Methoden

URL	GET	POST	PUT	DELETE
/buchliste	x	x	-	-
/buchliste/{isbn}	x	-	x	x
/buchliste/{autor}	x	-	-	-

Alle anderen Methoden wie COPY oder OPTIONS sind nicht implementiert. Der REST-Server wird eine Fehlermeldung mit HTTP Status 405 (Method not Allowed) erzeugen. Mehr Informationen auf Seite 8.

GET /buchliste

Beschreibung

Gibt alle Bücher der Liste / Datenbank zurück. Die Bücher sind nach ISBN sortiert.

Request

Parameter type	Content type	Attribut(en)	Beschreibung des Attributs	Definition des Attributs	Optional
-	-	-	-	-	-

Response

http Status code	Erläuterung	Attribut(en)	Beschreibung des Attributs	Definition des Attributs
200 (OK)	Die Liste mit den gespeicherten Büchern	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein
		titel	Der Titel des Buches	String, darf nicht leer sein
		autor	Der Autor des Buches	String, darf nicht leer sein

Beispiel

Request	Response
GET /buchliste HTTP/1.1 Host: <Host>	HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 { "msg": [{ <Buch 1> }, { <Buch 2> }, ... { <Buch n> }]} }}

Ein Buch wird gespeichert als:

```
{  
  "isbn" : 12,  
  "titel" : "Silo",  
  "autor" : "Hugh Howey"  
}
```

GET /buchliste/{isbn}

Beschreibung

Gibt das gewünschte Buch zurück, wenn das Buch in der Datenbank schon vorhanden ist.

Request

Parameter type	Content type	Attribut(en)	Beschreibung des Attributs	Definition des Attributs	Optional
path	-	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein	Nein

Response

http Status code	Erläuterung	Attribut(en)	Beschreibung des Attributs	Definition des Attributs
200 (OK)	Das Buch mit das eingegebenen ISBN	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein
		titel	Der Titel des Buches	String, darf nicht leer sein
		autor	Der Autor des Buches	String, darf nicht leer sein
400 (Bad Request)	Die eingegebene ISBN ist kleiner als 1	msg	Fehlermeldung	-
404 (Not Found)	Das Buch ist nicht vorhanden	msg	Fehlermeldung	-

Beispiel

Request	Response
GET /buchliste/{isbn} HTTP/1.1 Host: <Host>	HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 { "msg" : { <das gewünschte Buch> } }
GET /buchliste/{ungültiges isbn} HTTP/1.1 Host: <Host>	HTTP/1.1 400 Bad Request Content-type: application/json; charset=utf-8 { "msg" : " Die ISBN muss positiv sein." }
GET /buchliste/{ isbn nicht gefunden } HTTP/1.1 Host: <Host>	HTTP/1.1 404 Not Found Content-type: application/json; charset=utf-8 { "msg" : "Kein Buch mit ISBN {isbn}." }

GET /buchliste/{autor}

Beschreibung

Gibt alle Bücher zurück, die von dem gegebenen Autor geschrieben wurden. Die Bücher sind nach ISBN sortiert.

Notiz:

Um eine gültige URL zu haben, muss man die Leerzeichen mit _ ersetzen. Beispielsweise der Autor Hugh Howey wird in Hugh_Howey umgewandelt. Ein gültiger Request wäre somit [GET /buchliste/Hugh_Howey](#).

Request

Parameter type	Content type	Attribut(en)	Beschreibung des Attributs	Definition des Attributs	Optional
path	-	Autor (Name_Vorname)	Der Autor des Buches	String, darf nicht leer sein	String, darf nicht leer sein

Response

http Status code	Erläuterung	Attribut(en)	Beschreibung des Attributs	Definition des Attributs
200 (OK)	Die Bücher des gegebenen Autors	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein
		titel	Der Titel des Buches	String, darf nicht leer sein
		autor (Name Vorname)	Der Autor des Buches	String, darf nicht leer sein

Beispiel

Request	Response
GET /buchliste/{autor} HTTP/1.1 Host: <Host>	HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 <pre>{ "msg": [{ <Buch 1 des Autors> }, { <Buch 2 des Autors> }, ... { <Buch n des Autors> }]}</pre>
GET /buchliste/{autor existiert nicht} HTTP/1.1 Host: <Host>	HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 <pre>{ "msg": [] }</pre>

POST /buchliste

Beschreibung

Erstellt ein neues Buch und speichert es in der Datenbank.

Request

Parameter type	Content type	Attribut(en)	Beschreibung des Attributs	Definition des Attributs	Optional
body	Application/json	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein	Nein
		titel	Der Titel des Buches	String, darf nicht leer sein	Nein
		autor	Der Autor des Buches	String, darf nicht leer sein	Nein

Response

http Status code	Erläuterung	Attribut(en)	Beschreibung des Attributs	Definition des Attributs
201(Created)	Das Buch wurde hergestellt	-	Die Meldung enthält die URL des neuen Buches im Header	-
400 (Bad Request)	Das Buch ist schon vorhanden oder Fehler in dem JSON-String	msg	Fehlermeldung	-

Beispiel

Request	Response
POST /buchliste HTTP/1.1 Host: <Host> Content-Type: application/json; charset=utf-8 <pre>{ "isbn" : 12, "titel" : "Silo", "autor" : "Hugh Howey" }</pre>	HTTP/1.1 201 Created Location: {url des neuen Buches}
POST /buchliste HTTP/1.1 Host: <Host> Content-Type: application/json; charset=utf-8 <pre>{ "isbn" : {isbn schon vorhanden}, "titel" : "<titel des neuen Buches>", "autor" : "<autor des neuen Buches>" }</pre>	HTTP/1.1 400 Bad Request Content-type: application/json; charset=utf-8 <pre>{ "msg" : "Das Buch mit ISBN {isbn} ist schon vorhanden." }</pre>

PUT /buchliste/{isbn}

Beschreibung

Ändert das Buch mit der eingegebene ISBN.

Request

Parameter type	Content type	Attribut(en)	Beschreibung des Attributs	Definition des Attributs	Optional
path	-	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein	Nein
body	Application/json	titel	Der Titel des Buches	String, darf nicht leer sein	Nein
		autor	Der Autor des Buches	String, darf nicht leer sein	Nein

Response

http Status code	Erläuterung	Attribut(en)	Beschreibung des Attributs	Definition des Attributs
201 (No Content)	Das Buch wurde geändert	-	-	-
400 (Bad Request)	Fehler im JSON, ungültige ISBN oder URL	msg	Fehlermeldung	-
404 (Not Found)	Das Buch ist nicht vorhanden	msg	Fehlermeldung	-

Beispiel

Request	Response
PUT /buchliste/{isbn} HTTP/1.1 Host: <Host> Content-Type: application/json; charset=utf-8 <pre>{ "titel" : "<titel des Buches>", "autor" : "<autor des Buches>" }</pre>	HTTP/1.1 204 No Content
PUT /buchliste/{isbn} HTTP/1.1 Host: <Host> Content-Type: application/json; charset=utf-8 <pre>{ "titel" : " ", "autor" : "<autor des Buches>" }</pre>	HTTP/1.1 400 Bad Request Content-type: application/json; charset=utf-8 <pre>{ "msg" : "Der Titel darf nicht leer sein." }</pre>
PUT /buchliste/{ isbn nicht gefunden } HTTP/1.1 Host: <Host> Content-Type: application/json; charset=utf-8 <pre>{ "titel" : "<titel des Buches>", "autor" : "<autor des Buches>" }</pre>	HTTP/1.1 404 Not Found Content-type: application/json; charset=utf-8 <pre>{ "msg" : "Das Buch mit ISBN {isbn} ist nicht vorhanden." }</pre>

DELETE /buchliste/{isbn}

Beschreibung

Löscht das gewünschte Buch, wenn das Buch in der Datenbank schon vorhanden ist.

Request

Parameter type	Content type	Attribut(en)	Beschreibung des Attributs	Definition des Attributs	Optional
path	-	isbn	Eindeutige Nummer des Buches	Integer, darf nicht negativ sein	Nein

Response

http Status code	Erläuterung	Attribut(en)	Beschreibung des Attributs	Definition des Attributs
204 (No Content)	Das Buch wurde gelöscht	-	-	-
400 (Bad Request)	Es fehlt die ISBN in der URL oder ungültige ISBN	msg	Fehlermeldung	-
404 (Not Found)	Das Buch ist nicht vorhanden	msg	Fehlermeldung	-

Beispiel

Request	Response
DELETE /buchliste/{isbn} HTTP/1.1 Host: <Host>	HTTP/1.1 204 No Content
DELETE /buchliste HTTP/1.1 Host: <Host>	HTTP/1.1 400 Bad Request Content-type: application/json; charset=utf-8 { "msg": "Keine gueltige URL." }
DELETE /buchliste/0 HTTP/1.1 Host: <Host>	HTTP/1.1 400 Bad Request Content-type: application/json; charset=utf-8 { "msg": "Die ISBN muss positiv sein." }
DELETE /buchliste/{isbn nicht gefunden} HTTP/1.1 Host: <Host>	HTTP/1.1 404 Not Found Content-type: application/json; charset=utf-8 { "msg": "Kein Buch mit ISBN {isbn}." }

Nicht implementierten Methoden

Beschreibung

Nur die Methoden GET, POST, PUT und DELETE werden hier behandelt. Alle anderen Methoden sind nicht erlaubt.

Beispiel

Request	Response
COPY /buchliste/{isbn} HTTP/1.1 Host: <Host>	HTTP/1.1 405 Method Not Allowed Allow: GET, POST, PUT, DELETE Content-type: application/json; charset=utf-8 { "msg" : "Die Methode ist nicht implementiert." }
OPTIONS /buchliste HTTP/1.1 Host: <Host>	HTTP/1.1 405 Method Not Allowed Allow: GET, POST, PUT, DELETE Content-type: application/json; charset=utf-8 { "msg" : "Die Methode ist nicht implementiert." }