LU07.L02 - Bookshelf testen mit Bruno

Musterlösung zu LU07.A02 - Bookshelf testen mit Bruno. Die Testfälle gelten gleich für die Postman-Variante (LU07.L02 - Bookshelf testen).

Testfälle

Nr	Request	Funktion	Daten	Erwarteter Status
1	`list-200`	`GET list`	–	200, Array mit 3 Büchern
2	`read-200`	`GET read/<uuid>`	`c746a291-0ef9-4b2a-8268-392b12d636bd`	200, «The Winds of Winter»
3	`read-404`	`GET read/<uuid>`	unbekannte UUID `00000000-0000-4000-8000-000000000000`	404
4	`save-201-neu`	`POST save`	gültiges Buch ohne `book_uuid`	201
5	`save-200-aendern`	`POST save`	`book_uuid` eines bestehenden Buchs, neuer Preis 16.90	200
6	`save-400-titel-fehlt`	`POST save`	Feld `title` fehlt	400
7	`save-400-preis-keine-zahl`	`POST save`	`price` = abc	400
8	`save-422-titel-zu-kurz`	`POST save`	`title` = A (1 Zeichen)	422
9	`save-422-preis-zu-hoch`	`POST save`	`price` = 300.00	422
10	`save-422-isbn-pruefziffer`	`POST save`	`isbn` = 978-3-16-148410-1 (falsche Prüfziffer)	422
11	`save-422-format`	`POST save`	`format` = Paperback	422
12	`delete-200`	`DELETE delete/<uuid>`	`3247c340-c712-402e-b400-0a23a9368c97`	200
13	`delete-404`	`DELETE delete/<uuid>`	dieselbe UUID nochmals	404

Weitere Testfälle aus der Äquivalenzklassen-Analyse

Für eine vollständige Abdeckung der Validierungsregeln kommen pro Regel ein gültiger Grenzwert und ein ungültiger Wert dazu:

Datenfeld	gültig (201)	ungültig (422)
`title`	2 und 50 Zeichen	1 und 51 Zeichen
`author`	4 und 100 Zeichen	3 und 101 Zeichen
`price`	`0.00`, `299.95`, `12.5`	`-0.01`, `300.00`, `12.345` (3 Nachkommastellen)
`isbn`	`978-3-16-148410-0`, `9783161484100`	falsche Prüfziffer, 12 Ziffern, Buchstaben
`format`	`Hardcover`, `Softcover`, `eBook`, `Audio`	`Paperback`, leer
`book_uuid`	bestehende UUID (200)	123 (formal falsch: 400)

Die Requests `save` und `delete` verändern die Daten auf dem Server. Damit `list-200` immer drei Bücher findet, stehen die Ordner in der Reihenfolge `1-list`, `2-read`, `3-save`, `4-delete`. Der Collection Runner führt die Ordner

alphabetisch aus.

Projektstruktur

```
bookshelf-bruno/
  bruno.json
  environments/
    dev.bru
  1-list/
    list-200.bru
  2-read/
    read-200.bru
    read-404.bru
  3-save/
    save-201-neu.bru
    save-200-aendern.bru
    save-400-titel-fehlt.bru
    save-400-preis-keine-zahl.bru
    save-422-titel-zu-kurz.bru
    save-422-preis-zu-hoch.bru
    save-422-isbn-pruefziffer.bru
    save-422-format.bru
  4-delete/
    delete-200.bru
    delete-404.bru
```

Beispiele

2-read/read-404.bru

```
meta {
  name: read-404
  type: http
  seq: 3
}

get {
  url: https://it.bzz.ch/book/read/00000000-0000-4000-8000-000000000000
  body: none
  auth: none
}
```

3-save/save-201-neu.bru

```
meta {
  name: save-201-neu
  type: http
  seq: 4
}

post {
  url: https://it.bzz.ch/book/save
  body: formUrlEncoded
  auth: none
}

body:form-urlencoded {
  title: Testbuch
  author: Anna Muster
  price: 19.90
  format: Softcover
  isbn: 978-3-16-148410-0
}
```

Die anderen `save`-Requests unterscheiden sich nur in den Formulardaten. Die Tests (Statuscode, Daten) kommen erst in LU08.A01 dazu, die Variable `{{baseUrl}}` in LU08.A02.

Collection Runner

Der Runner führt alle 13 Requests nacheinander aus und zeigt pro Request Statuscode und Antwortzeit. Ohne Tests bewertet er nur, ob der Request überhaupt eine Antwort erhalten hat. Ob der Statuscode der erwartete ist, muss man in dieser Aufgabe noch selbst vergleichen.

M450-LU07, M450-C3B, M450-B1I

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu07/loesungen/bookshelfbruno

Last update: **2026/11/17 17:52**