LU08.L01 - Resultate auswerten mit Bruno

Musterlösung zu LU08.A01 - Resultate auswerten mit Bruno. Grundlage sind die 13 Requests aus LU07.L02. Jeder Request erhält einen Test für den Statuscode, `list` und `read` zusätzlich einen Test für die Daten.

In Bruno liest man den Body mit `res.getBody()` (oder `res.body`). Eine Funktion `res.json()` gibt es in Bruno nicht, das ist die Postman- bzw. fetch-Schreibweise.

Tests pro Request

Request	Test 1	Test 2
`list-200`	Status 200	Array mit genau 3 Büchern, jedes mit `book_uuid` und `title`
`read-200`	Status 200	`book_uuid` und `title` stimmen
`read-404`	Status 404	–
`save-201-neu`	Status 201	–
`save-200-aendern`	Status 200	–
`save-400-*`	Status 400	–
`save-422-*`	Status 422	–
`delete-200`	Status 200	–
`delete-404`	Status 404	–

1-list/list-200.bru

```
meta {
  name: list-200
  type: http
  seq: 1
}

get {
  url: {{baseUrl}}/list
  body: none
  auth: none
}

tests {
  test("Status 200", function () {
    expect(res.status).to.equal(200);
  });

  test("Liste mit 3 Büchern", function () {
    const body = res.getBody();
    expect(body).to.be.an("array").with.lengthOf(3);
```

```
    body.forEach((book) => {
      expect(book).to.have.property("book_uuid");
      expect(book).to.have.property("title");
    });
  });
}
```

2-read/read-200.bru

```
meta {
  name: read-200
  type: http
  seq: 2
}

get {
  url: {{baseUrl}}/read/c746a291-0ef9-4b2a-8268-392b12d636bd
  body: none
  auth: none
}

tests {
  test("Status 200", function () {
    expect(res.status).to.equal(200);
  });

  test("Richtiges Buch", function () {
    const book = res.getBody();
    expect(book.book_uuid).to.equal("c746a291-0ef9-4b2a-8268-392b12d636bd");
    expect(book.title).to.equal("The Winds of Winter");
  });
}
```

3-save/save-422-isbn-pruefziffer.bru

Alle anderen Requests sind gleich aufgebaut: nur der erwartete Statuscode ändert sich.

```
meta {
  name: save-422-isbn-pruefziffer
  type: http
  seq: 10
}

post {
  url: {{baseUrl}}/save
  body: formUrlEncoded
```

```
  auth: none
}

body:form-urlencoded {
  title: Testbuch
  author: Anna Muster
  price: 19.90
  format: Softcover
  isbn: 978-3-16-148410-1
}

tests {
  test("Status 422", function () {
    expect(res.status).to.equal(422);
  });
}
```

Ausführen

Im Collection Runner oder auf der Kommandozeile:

```
npm install -g @usebruno/cli
cd bookshelf-bruno
bru run --env dev
```

Erwartetes Ergebnis: `Requests 13 (13 Passed)`, `Tests 15/15`. Die Lösung wurde gegen einen nachgebauten Server mit denselben Regeln geprüft. Wird die Collection mehrmals hintereinander gegen den echten Server ausgeführt, schlägt `delete-200` fehl, weil das Buch bereits gelöscht ist und `list-200` nur noch 2 oder wegen `save-201-neu` mehr Bücher findet. Vor einem neuen Durchlauf müssen die Testdaten zurückgesetzt werden.

M450-LU08, M450-C3I, M450-B1I

Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu08/loesungen/testsbruno

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