

# LU08a - Resultate auswerten mit Bruno



Mittels **Tests** kann Bruno die Antwort einer API automatisiert auswerten.

In jeder Request-Datei (`.bru`) kannst du im Bereich **Tests** kleine JavaScript-Funktionen schreiben, um das Resultat eines Requests auszuwerten. Bruno zeigt dir danach direkt an, ob der Request das erwartete Resultat liefert hat oder nicht.

## http-Statuscode prüfen

Im einfachsten Fall will ich nur den http-Statuscode überprüfen:

```
test("Status test", () => {
  expect(res.status).to.eql(200);
});
```

Diese Zeilen reichen schon, damit Bruno den http-Statuscode überprüft. Ein paar Erklärungen:

- `„Status test“` ist der Name meines Tests. Dieser wird im Testprotokoll angezeigt und sollte möglichst aussagekräftig sein.
- Ich prüfe hier, ob die API mit Status **200** antwortet. Je nach Testfall würde ich unterschiedliche Statuscodes erwarten.

# LU08a - Resultate auswerten



Mittels **Tests** kann Bruno die Antwort einer API automatisiert auswerten.

In jeder Request-Datei (.bru) kannst du im Bereich **Tests** kleine JavaScript-Funktionen schreiben, um das Resultat eines Requests auszuwerten. Bruno zeigt dir danach direkt an, ob der Request das erwartete Resultat liefert hat oder nicht.

## http-Statuscode prüfen

Im einfachsten Fall will ich nur den http-Statuscode überprüfen:

```
test("Status test", () => {  &nbsp;expect(res.status).to.eql(200); });
```

Diese Zeilen reichen schon, damit Bruno den http-Statuscode überprüft. Ein paar Erklärungen:

- „Status test“ ist der Name meines Tests. Dieser wird im Testprotokoll angezeigt und sollte möglichst aussagekräftig sein.
- Ich prüfe hier, ob die API mit Status **200** antwortet. Je nach Testfall würde ich unterschiedliche Statuscodes erwarten.

## LU08a - Resultate auswerten



Mittels **Tests** kann Bruno die Antwort einer API automatisiert auswerten.

In jeder Request-Datei (.bru) kannst du im Bereich **Tests** kleine JavaScript-Funktionen schreiben, um das Resultat eines Requests auszuwerten. Bruno zeigt dir danach direkt an, ob der Request das erwartete Resultat liefert hat oder nicht.

### http-Statuscode prüfen

Im einfachsten Fall will ich nur den http-Statuscode überprüfen:

```
test("Status test", () => {
  expect(res.status).to.eql(200);
});
```

Diese Zeilen reichen schon, damit Bruno den http-Statuscode überprüft. Ein paar Erklärungen:

- „Status test“ ist der Name meines Tests. Dieser wird im Testprotokoll angezeigt und sollte möglichst aussagekräftig sein.
- Ich prüfe hier, ob die API mit Status **200** antwortet. Je nach Testfall würde ich unterschiedliche Statuscodes erwarten.

### Daten in der Response prüfen

Wenn ein Request Daten von der API anfordert, wollen wir den Inhalt der Response überprüfen. Schliesslich wollen wir nicht irgendwelche Daten lesen, sondern einen bestimmten Datensatz.

```
test("Data test", () => {
  expect(res.status).to.eql(200);
  const responseJson = res.json();
  expect(responseJson.givenname).to.eql("Eva");
  expect(responseJson.age).to.eql(28);
```

{});

In diesem Beispiel prüfen wir drei Elemente der Response-Daten:

- Die Response muss erfolgreich sein (Statuscode 200).
- Der Vorname muss „Eva“ sein.
- Das Alter muss 28 sein.

Trifft irgendeine dieser Bedingungen nicht zu, zeigt Bruno den Request als fehlgeschlagen an.

## Weitere Beispiele

Im [Bruno Docs: Tests](#) findest du noch weitere Beispiele und Möglichkeiten für Tests.

---

M450-LU08



Kevin Maurizi

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/modul/m450/learningunits/lu08/testsbruno?rev=1758180248>

Last update: **2025/09/18 09:24**

