

# LU08b - Variablen einsetzen



Variablen in Postman erleichtern das Anpassen von Requests an unterschiedliche Bedingungen.

## Variablen für Pfade

Im Verlauf der Entwicklung wird eine Applikation zunächst in der Entwicklungsumgebung ausgeführt. Für die Requests in Postman bedeutet dies in der Regel, dass unsere API unter einem Pfad wie <http://localhost:5000/>... erreichbar ist. Später werden wir die Applikation auf unser Testsystem deployen. Der Pfad im Testsystem könnte z.B. <https://test.bzz.ch/>... sein. Ich möchte meine API-Test in beiden Umgebungen nutzen können. Welche Möglichkeiten habe ich?

- Ich passe jedesmal den Pfad in allen Requests an.
- Ich erstelle Kopien aller Requests 1x mit dem Pfad des Testsystems und 1x mit dem Pfad der Entwicklungsumgebung.
- Ich verwende eine globale Umgebungsvariable und passe jeweils nur den Wert dieser Variable an.

Um eine solche Umgebungsvariable einzusetzen, benötigen wir nur wenige Schritte:

1. Definiere die globale Umgebungsvariable.
2. Lege den Werte fest.
3. Füge die Variable in den Pfad deiner Requests ein.

Im [Postman Learning Center](#) siehst du, wie das gemacht wird.

## Variablen für Daten

Mit Variablen können wir auch Werte aus der Response auf einen unserer Requests speichern und in einem anderem Request wiederverwenden.

### Beispiele

#### Erstellen und Lesen von Daten

1. Ein Request speichert einen neuen Kunden. Als Antwort liefert die API die Kundennummer.
2. Ich speichere die Kundennummer in einer Variable.
3. Ein weiterer Request verwendet diese Variable, um den vorher erstellen Kunden zu lesen.

## Authentifikation

Bei vielen APIs brauchen wir ein Token, welches unsere Identität und Berechtigungen enthält. Ein solches Token hat immer eine begrenzte Gültigkeit, weshalb wir es nicht einfach einmalig erstellen und immer wieder nutzen können. Dieses Token erhalten wir, indem wir einen Login-Request senden.

1. Ein Request sendet die Login-Daten an die API. Als Antwort erhalte ich ein Authentifizierungstoken.
2. Ich speichere das Token in einer Variable.
3. Bei jedem weiteren Request sende ich das gespeicherte Token mit.

## Umsetzung

Ich zeige Ihnen hier ein Beispiel, wie ich das Beispiel mit der Authentifikation in einer Applikation umsetze.

### 1. Login student

Der Request sendet die Login-Daten für einen Studenten. Die Response ist eine JSON-Struktur, die das Token enthält.

#### Skript in „tests“

```
pm.test("Status test", function () {  
    pm.response.to.have.status(200); // Prüfe auf  
    HTTP-Statuscode 200  
    let jsonData = JSON.parse(responseBody); // Parse die  
    JSON-Daten in ein Javascript-Objekt  
    pm.collectionVariables.set("token", jsonData.token); // Speichere das  
    Token als Variable in meiner Collection  
});
```

### 2. Lies alle Anmeldungen zu einem Studenten

Dieser Request liest alle Anmeldungen zum angemeldeten Studenten. Dazu muss das Token aus dem ersten Request mitgesendet werden. Dieses Token wird im Header des Requests mitgesendet und nicht etwas in der URL.

#### Token in „headers“

- Key: Authorization
- Value: Bearer `token`



Marcel Suter

From:

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

Permanent link:

<https://wiki.bzz.ch/modul/m450/learningunits/lu08/variablen>Last update: **2024/03/28 14:07**