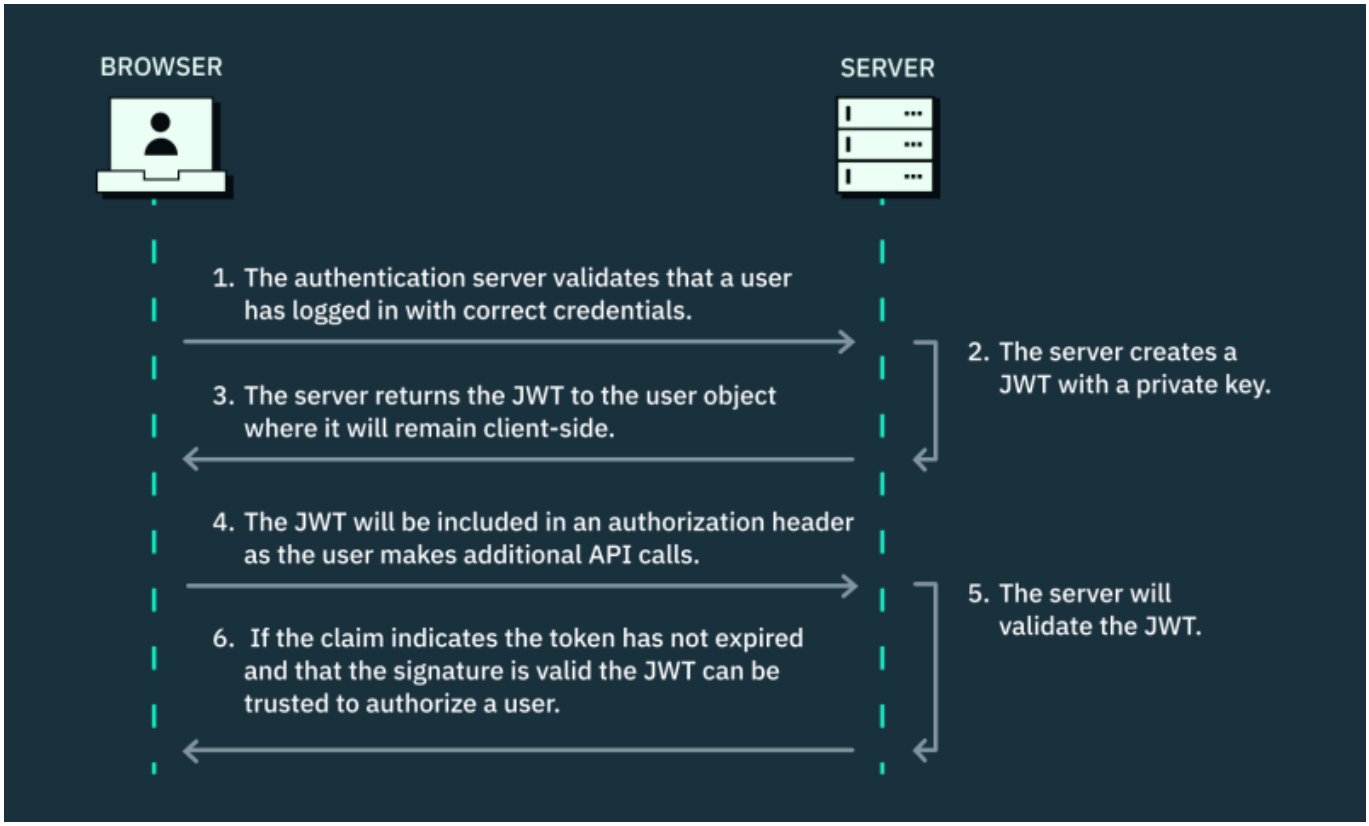
LU06a - Webtoken & Login

Übersicht

Unser bisheriger REST-API-Endpunkt `/books` ist öffentlich und kann entsprechend von allen aufgerufen werden. Anstatt bei jedem API-Aufruf den Benutzernamen und das Passwort mitzuschicken, wird in der Regel beim Login ein Token erstellt und zurückgeschickt. Der Benutzer kann dieses Token für alle weiteren Aufrufe verwenden bis die festgelegte Gültigkeitsdauer abgelaufen ist.

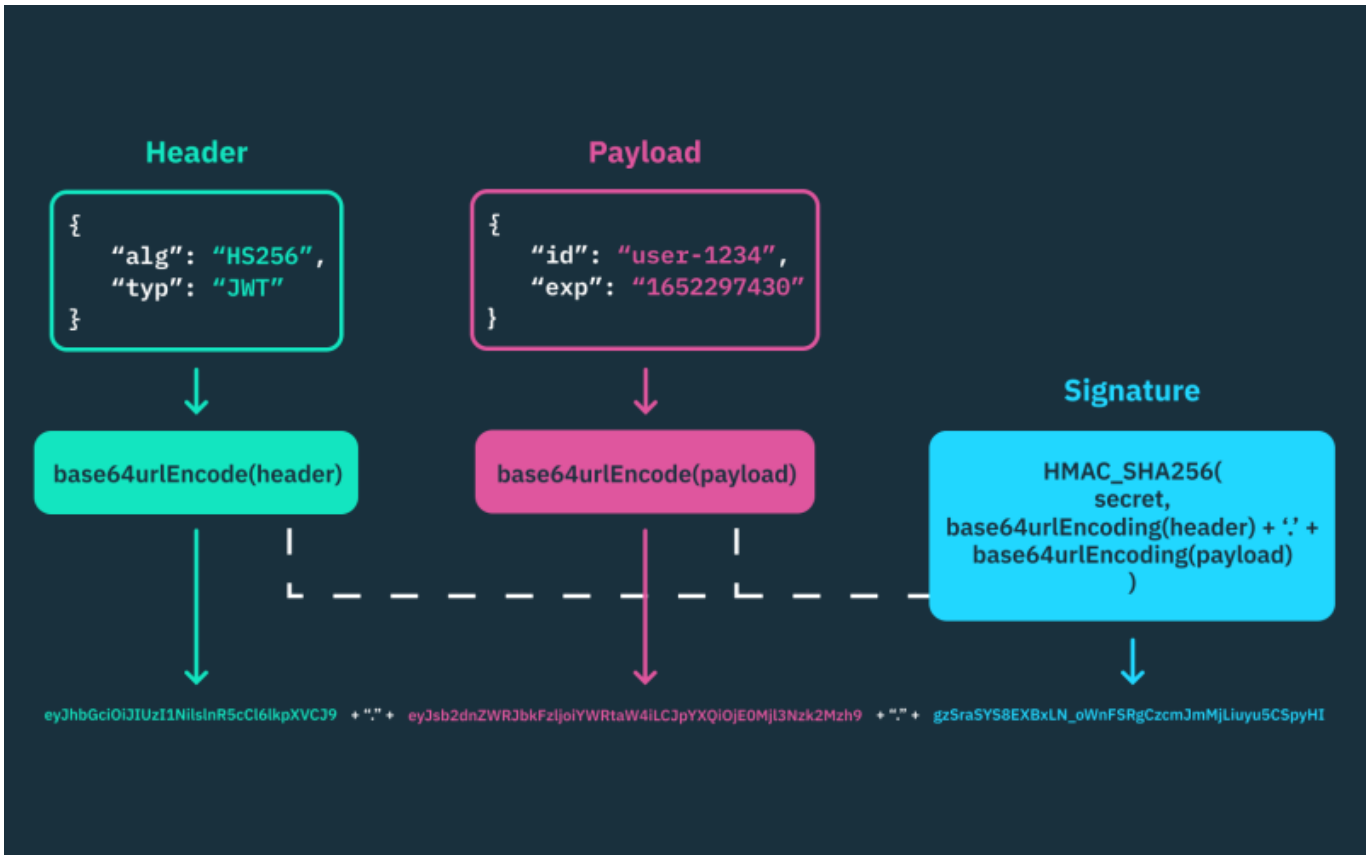
Die ganze Erklärung finden Sie auf: https://stytch.com/blog/what-is-a-json-web-token/



JSON Web Token (JWT)

Ein JWT besteht aus dem Hash des Headers, dem Hash des Payloads und dem Hash der signierten vorherigen Teilen.

Mit der Signatur wird sichergestellt, dass sich kein Benutzer selber ein JWT erstellen und mitschicken kann.



Login

Anforderung 6: Ein Benutzer in der Datenbank soll sich via `/auth/login` anmelden können.

Sie erhalten die JWT-Funktionalitäten mit dem Commit `07b12a2`. Implementieren Sie eine POST-API, welche als JSON-Body die E-Mailadresse („email“) und das Klartextpasswort („password“) akzeptiert.

Die Implementation könnte grob so aussehen:

```
var json = ctx.bodyValidator(Map.class)
    .check(m -> m.containsKey("email"), "email is required")
    .check(m -> m.containsKey("password"), "password is required")
    .get();

String inputEmail = (String) json.get("email");
String inputPassword = (String) json.get("password");

... // TODO: find user in database with given email

byte[] storedSalt = Base64.getDecoder().decode(user.getPasswordSalt());
byte[] storedHash = Base64.getDecoder().decode(user.getPasswordHash());

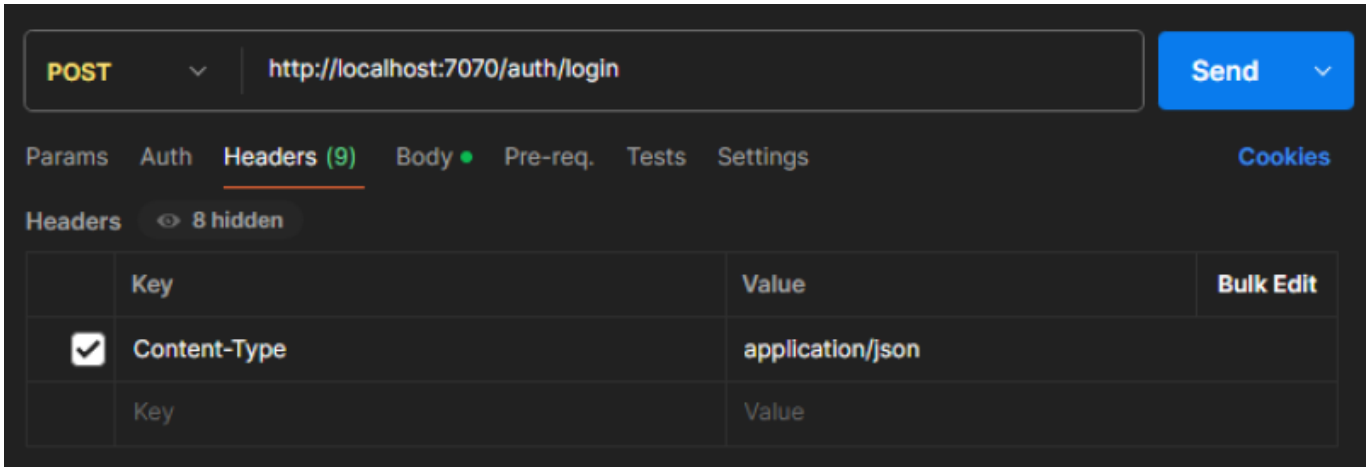
if (PasswordHandler.verifyPassword(inputPassword, storedHash, storedSalt)) {
    String jwt = JwtHandler.createJwt(inputEmail, user.getId());
    ctx.json(Map.of("token", jwt));
    return;
```

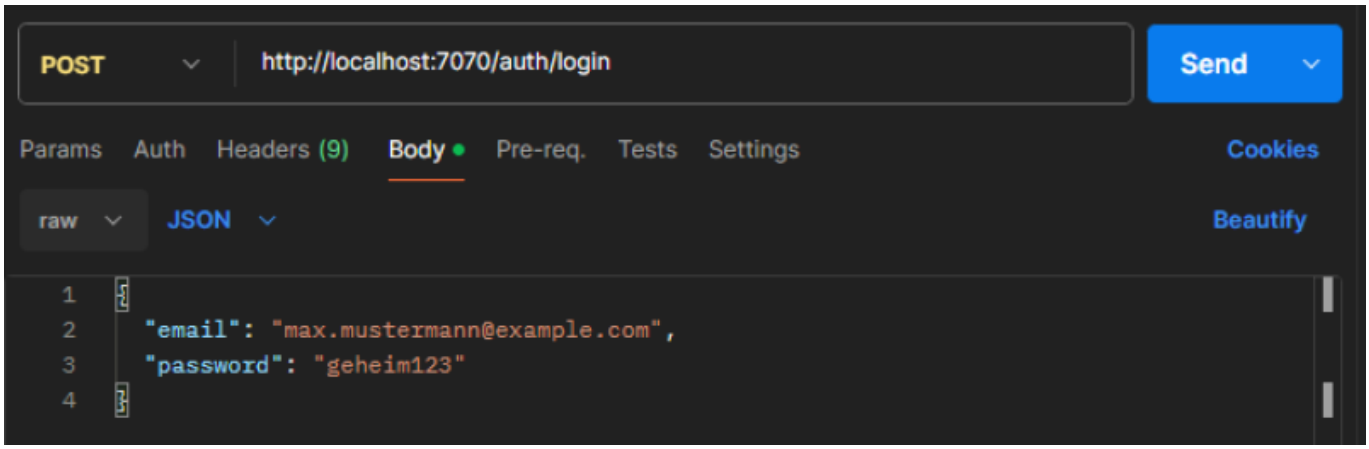
```
}

...

// TODO: Use the same error message if the user is not found and if the
password is wrong
ctx.status(401).json(Map.of("error", "Invalid email or password"));
```

Mit Postman können Sie Ihre API testen.





Analysieren Sie das zurückgelieferte JWT auf https://www.jwt.io/

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

Permanent link:
https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu06/a?rev=1758578883

Last update: **2025/09/23 00:08**