

# LU10a - Umgang mit Exceptions

## Validierung des JWT

In der vorgeschlagenen Klasse `JwtUtil.java` gibt es bereits eine Möglichkeit, um ein JWT zu prüfen (Siehe auch LU08.A05 Register/Login).

```
public boolean validateToken(String token) {  
    try {  
        Jwts.parserBuilder().setSigningKey(key).build().parseClaimsJws(token);  
        return true;  
    } catch (Exception e) {  
        return false;  
    }  
}
```

### Kleine Wiederholung



Setzt man Exceptions ein, sollte man diese auch loggen oder weiterschicken (gegebenenfalls sogar ans Frontend). Ein „catch“ ohne Verwendung der gefangenen Exception, wie im Beispiel, ist zu vermeiden.

Es ist anzumerken, dass Exceptions in aller Regel in Bezug auf die Performanz schlechter abschneiden als die entsprechenden Checks. Daher sollte man Exceptions nie in „Schönwetter“-Fällen benutzen, sondern höchstens in unerwarteten Situationen. Ein invalides Token, bzw. allgemein ein ungültiger API-Aufruf, ist in unserem Fall unerwartet und kann mittels Exceptions gelöst werden. Werden aber viele ungültige API-Aufrufe erwartet, sollte man eine Exception-freie Alternative in Erwägung ziehen.

Wenn zudem Informationen wie das „Subject“ aus dem JWT herausgelesen werden müssen, ergibt es keinen Sinn dies separat zu machen.

From:

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

Permanent link:

<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu10/a?rev=1762726906>

Last update: **2025/11/09 23:21**

