

Woche 3: Generics und Reflection

Aufgabe 1: JSON Serializable

In dieser Aufgabe erstellt ihr Custom-Annotations, um Objekte in eine JSON-Datei zu serialisieren. Erstellt dazu die folgenden Klassen:

Annotation

Klasse	Beschreibung
JsonSerializable	Annotation für Klassen. Es sollen nur Klassen mit dieser Annotation serialisiert werden.
JsonElement	Annotation für Attribute. Die Annotation kennzeichnet Attribute, die serialisiert werden sollen. Die Annotation hat ein Attribut (key), das es ermöglicht Attribute in der JSON-Datei anders zu benennen.
Init	Annotation für Methoden. Dieses Annotation kennzeichnet Methoden, die vor der Serialisierung ausgeführt werden sollen. In dieser Aufgabe dient diese Annotation dazu, die ersten Buchstaben des Vor- und Nachnamen gross zu schreiben.

Alle diese Attribute wurden in der User-Klasse bereits angewendet. Die genauen Anforderungen stehen als Kommentar im Code.

Aufgabe 2: Eigenes Web-Framework mit Annotationen

In dieser Aufgabe erstellt ihr ein einfaches Web-Framework in Java, das Annotationen nutzt. Das Framework kann **HTML**-Seiten aus dem **resources**-Ordner laden und **JSON-Daten** mithilfe eines in der vorherigen Aufgabe erstellten JSON-Serializers zurückgeben.

Gegeben: Grundgerüst des Web-Frameworks

1. Annotationen definieren:

Erstellt die folgenden Annotationen, um Controller und Routen zu markieren:

- **@WebController** → Markiert Klassen als **Controller**, damit das Framework sie erkennt..
- **@Get(path, contentType = "application/json")**
 - Definiert eine **GET-Route** für eine Methode.
 - **path**: Der URL-Pfad, der die Methode aufruft.
 - **contentType**: Standardmässig **"application/json"**(kann aber überschrieben werden).

2. Framework Implementation

Erweitert das gegebene Grundgerüst, um folgende Funktionalitäten zu ermöglichen:

- **addController(Object controller)**

- Scannt die Klasse nach Methoden mit **@Get**-Annotation.
- Registriert alle gefundenen Routen im Framework.
- **getResponse(String path)**
 - Sucht nach der passenden Methode für den gegebenen Pfad.
 - Führt die Methode aus und gibt das **Ergebnis als String** zurück.
 - Berücksichtigt den **contentType**, um HTML oder JSON auszugeben.

3. Controller

- Fügt die benötigten Annotationen hinzu.
- Implementiert die folgenden Methoden:
 1. **showHTML()**
 - Lädt die Datei index.html aus dem resources-Ordner.
 - Gibt den Inhalt als HTML-String zurück.
 2. **getUser()**
 - Serialisiert den user mit dem vorher erstellten JSON-Serializer.
 - Gibt das Ergebnis als JSON-String zurück.

4. Testen der Endpoints

- Um die Endpoints zu testen, führt das .http-File in den Tests aus.