UML → JSON (Schemata) → OpenAPI (Spezifikation)

Student:
Faissal Azdufal, 1111995

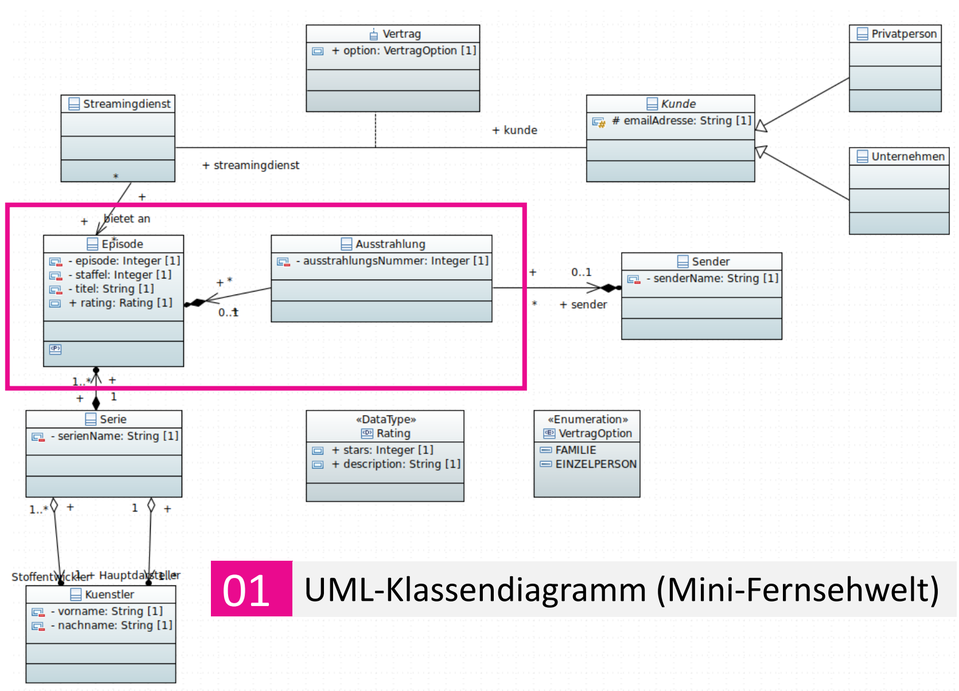
Dozent:
Sebastian Zengerle

Betrieb:
Deutsche Telekom Technik GmbH

Ausgangslage und Motivation

Das Projekt **ACCESS 4.0** transformiert das Zugangsnetz der Telekom, indem mehrere Software-Systeme miteinander kommunizieren, die ein gemeinsames **Datenmodell** nutzen.
Dieses Datenmodell und die Datenaustauschformate werden vom **Ende-zu-Ende-Design Team** bereitgestellt. Zur softwarenahen Abbildung des Datenmodells wird ein **UML-Klassendiagramm** genutzt, aus dem **JSON-Schemata** generiert werden. Die derzeitige **manuelle Generierung** dieser Schemata durch das Ende-zu-Ende-Design Team ist **zeitaufwändig** und **fehleranfällig**.

Im Rahmen meines Praxisprojekts wird der Bereitstellungsprozess **optimiert**, wodurch er nicht nur **schneller** und **effizienter** wird, sondern auch **weniger fehleranfällig** sein wird. Dies **entlastet** das Ende-zu-Ende-Design Team. Das Projekt dient als „Proof of Concept“.



01 UML-Klassendiagramm (Mini-Fernsehwelt)

Unified
Modeling
Language

Das Datenmodell wird mithilfe der visuellen Sprache **UML** modelliert. Dafür wird das Tool **Papyrus** von **Eclipse** genutzt. Papyrus übersetzt die UML-Diagramme in eine **XML**-Datei.

Zu Beginn muss somit die XML-Datei gelesen und in einem **objektorientierten Format** dargestellt werden.

Hierfür wird ein Skript in **Golang** geschrieben und als Entwicklungsumgebung wird **Visual Studio Code** genutzt.

02 Episode JSON Schema

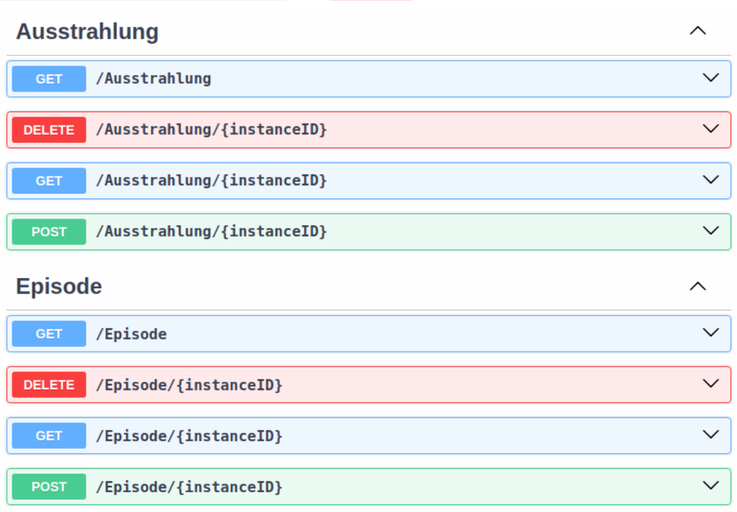
03 Ausstrahlung JSON Schema

JSON
Schemata

Das Ziel des Projekts ist die Bereitstellung der **JSON Schemata**. Deswegen müssen diese nun aus dem zuvor erstellten Format generiert werden.

Vor der Implementierung muss sich ein **Mapping** vom UML-Klassendiagramm hin zu JSON Schemata überlegt werden.

Mithilfe des entwickelten Mappings wird dann das Skript erweitert, sodass die Schemata korrekt erstellt werden.



04 OpenAPI Spezifikation (Ausstrahlung- & Episoden-Endpunkte)

OpenAPI
Spezifikation

Für die Bereitstellung der generierten JSON Schemata wird eine **OpenAPI-Spezifikation** erstellt.

Diese schafft eine generalisierte Version des UML-Diagramms und dient als Schnittstelle für die o.g. Software Systeme.

Das Skript wird nochmals erweitert, um auch die Bereitstellung der Schemata zu automatisieren.

Ergebnis

Im Projekt wurde ein **effizienter**, **zuverlässiger** und **fehlerarmer Prozess** für die Erstellung und Bereitstellung von JSON-Schemata entwickelt. Durch die Nutzung von UML-Modellierung, Skriptentwicklung und einer OpenAPI-Spezifikation konnte die **manuelle Generierung** von Schemata **eliminiert** werden. Das Ergebnis ist eine **verbesserte Arbeitsweise**, die das Ende-zu-Ende-Design Team **entlastet** und die **Effizienz** steigert.

Durch die **erfolgreiche** Umsetzung des „**Proof of Concept**“ wurde eine Lösung für das zugrunde liegende Problem geschaffen, das dieses Projekt motiviert hat. Damit das Ergebnis nun seinen Zweck erfüllen kann, muss das Skript in etwaigen **Automatisierungstools** eingebaut werden.

h_da hochschule darmstadt informatik
member of EUT+ EUROPEAN UNIVERSITY OF TECHNOLOGY
Deutsche Telekom

Praxisprojekt 2
Informatik Dual
2024

Gruppe B