

Anforderungen

1. Bereitstellung Endpunkt

- Stelle für die Anlage eines Patientenstammdatensatzes den REST-Endpunkt POST [baseUrl]/Patient gemäß der [FHIR-Spezifikation](#) bereit.
- Dieser Endpunkt muss die Beispiellressource „Beispiel-FHIR-Ressource-Patient.json“ im Anhang verarbeiten können, wenn diese welche gemäß der [FHIR-Spezifikation via REST angeliefert](#) wird.

2. Aufruf proprietäre API

- Aus der FHIR Patient-Ressource sollen die Eigenschaften *Vorname*, *Nachname* und *Geburtsdatum* extrahiert werden.
- Diese Eigenschaften sollen an eine proprietäre API übergeben werden. Da das Datenmodell der proprietären API weniger umfangreich ist, müssen in der FHIR-Ressource ggfs. separat übertragene Mehrfach-Vornamen bzw. -Nachnamen jeweils verkettet und mittels Leerzeichen getrennt werden. Das Geburtsdatum muss im Format *DD.MM.YYYY* übergeben werden.
- Die proprietäre API bietet zur Anlage von Patienten die Route POST [baseUrl]/Person an. Diese Route erwartet die Übergabe der o.g. Eigenschaften als JSON-Objekt gemäß der JSON-Scheme Definition „Person-JSON-Scheme.json“ im Anhang.
- Zum Testen der API-Calls durch Deine implementierte Applikation kann die Template-Datei „Test-Patient-API_Dev.json“ im Anhang für die freie Software [Mockoon](#) genutzt werden. Damit kannst Du lokal einen Test-Server auf einem separaten Port (3001) bereitstellen, um die API der proprietären Applikation zu simulieren. Das Mockoon-Template ist so konfiguriert, dass Mockoon unabhängig vom gesendeten Request abwechselnd mit dem erfolgreichen Statuscode 201 und dem Fehler-Statuscode 500 antwortet.

3. Rückgabe Statuscode

- Der Statuscode, welcher von der proprietären Applikation (bzw. in der Entwicklungsumgebung von Mockoon) an Deine Fassade zurückgegeben wird, lautet im Erfolgsfall 201. Fehlerfälle müssen nicht genauer interpretiert werden als „Statuscode ≥ 400 “.
- Wenn der Statuscode der proprietären API einen erfolgreichen Status signalisiert, soll von der FHIR-Fassade eine erfolgreiche Operation gemäß FHIR-Spezifikation an den FHIR-Client zurückgegeben werden.
- Wenn der Statuscode der proprietären API einen Fehlerfall signalisiert, soll von der FHIR-Fassade ein Fehler gemäß FHIR-Spezifikation an den FHIR-Client zurückgegeben werden. Für die Fehlerbeschreibung kann immer ein nicht näher spezifizierter interner Serverfehler signalisiert werden.

4. Dokumentation

- Verwalte den Quelltext Deines Projektes als git-Verzeichnis mit sinnvollen Commits inklusive Commit-Messages.
- Erstelle eine kurze Dokumentation in Form einer readme.md, welche die Installation und Nutzung der Anwendung beschreibt.

- Dokumentiere den Aufbau Deiner Applikation so, dass sich ein/e fremde/r Entwickler/in effizient selbstständig einarbeiten kann.

Technische Vorgaben

- Programmiersprache: Java
- Frameworks: Du kannst gerne gängige und frei verfügbare Frameworks, wie Spring Boot für die API-Entwicklung, nutzen.

Bibliotheken: Du kannst gerne die frei verfügbare Bibliothek [HAPI FHIR](#) für die Bereitstellung der HL7 FHIR-API sowie die Implementierung des FHIR Patient-Modells nutzen.