1 Aufgabe

Entwicklung eines Backends für eine der REST-Schnittstellen „reservations“ oder „personnel“ wie in gitlab.com/biletado/apidocs in Form von OpenAPI dokumentiert.
Jedes Backend muss mindestens eine Anfrage an eine andere Schnittstelle stellen, um die eigene API vollständig bedienen zu können. Der Endpunkt für diese Anfrage soll konfigurierbar sein.
Für bestimmte Anfragen ist eine Authentifizierung nötig. Für die Authentifizierung werden Json Web Tokens (JWT) verwendet. Bestimmte Berechtigungen (Autorisierung) müssen nicht geprüft werden, aber die Gültigkeit inklusive Signatur des JWT. Die Authentifizierung kann, insofern möglich, auch über eine Middleware/Reverse-Proxy statt im Backend-Quellcode geprüft werden.

2 Bewertung

Die Abschnitte im backend-howto (permalink) werden wie folgt bewertet:

containers	15%	Das Backend kann als container nach Spezifikation der OpenContainerInitiative gebaut und gestartet werden.
script everything	5%	Skript für den Containerbau und Tests in standardisierten Umgebung ohne spezielle Vorbedingungen wie installierte Entwicklungstools.
choose your framework wisely	15%	Aktuelle Version
secure your queries	5%	
clean code	5%	
validate the JWT	10%	
tracing	nicht bewertet	
configurability	5%	
compose-environment	5%	
licensing	1% (bonus)	
version-control	1% (bonus)	
write documentation	5%	
testautomation	5%	Ein einzelner Test reicht, egal was für einer, hauptsache automatisiert.
logging	5%	
API	20%	

3 Gruppenteilnehmer

Liste aller Teilnehmer:innen dieser Gruppe

Fabian Droll, Davis Schnebelt, Johannes Vater

4 Abgabe

Für die Abgabe wird benötigt:

- dieser Fragebogen
- compose-Konfiguration
- backend-Quellcode
- container-image des backends

Auch, wenn die compose-Konfiguration, der Quellcode und/oder das Image öffentlich erreichbar sind, bitte wenn möglich trotzdem bei wenigstens einem Gruppenmitglied in Moodle hochladen, für den Fall, dass mit dem Abruf etwas schiefgeht (fehlende Berechtigung, etc.). Für den container bitte `docker save` (docs) verwenden.

5 Quellcode

compose	https://github.com/fabiand35/spring/tree/master/compose
backend	https://github.com/fabiand35/spring/tree/master/src
backend-container	https://ghrc.io/fabiand35/personal

6 Lizenz

Lizenz des Backend-Quellcodes und dessen Dateiname.

MIT: LICENSE.txt

7 Fork

Link zur Projektbasis des Backends, falls Anwendbar.

https://github.com/fabiand35/spring/tree/master/src

8 Sprache und Frameworks

Verwendetes Framework.

Sprache	Java
Sprachversion	19
Sprachversion ist aktuell	☐ Ja
Framework (FW)	Spring Boot
FW-version	2.7.0
FW-version ist aktuell	☐ (Ja)
Website zum FW	https://spring.io/projects/spring-boot
Prepared statements/ORM	Hibernate
ORM Version	5.6.9
ORM Version ist aktuell	☐ Ja
Website zum ORM	https://hibernate.org/

9 Automatisierung

GitHub Actions

10 Testautomatisierung

unittest während build

11 Authentifizierung

- ☐ JWT wird berücksichtigt
- ☐ Signatur wird überprüft

12 Konfiguration & Dokumentation

- Dokumentation existiert
- ☐ Konfigurationsparameter sinnvoll gewählt
- ☐ keine Hardcoded Zugänge zu angeschlossenen services (URLs, Passwörter, Datenbanknamen, etc.)
- ☑ Mögliche Umgebungsvariablen sind dokumentiert

13 Logging

- [ ] Logsystem des Frameworks wurde genutzt
- [x] Logs enthalten sinnvolle Details (z.B. object-ids)
- [ ] Loglevel ist konfigurierbar