

## LU15.A03 Wahl des Modells & Prompt Engineering

Probieren Sie andere LLM-Modelle aus.

Für Gemini-Modelle benötigt es eine zusätzliche Anpassung von `build.gradle`. Je nachdem, welche Dependency, dass geladen wird, werden andere Konfigurationen automatisch im Hintergrund genutzt.

```
// implementation 'org.springframework.ai:spring-ai-starter-model-openai'  
implementation 'org.springframework.ai:spring-ai-starter-model-google'
```

Falls Sie das Embedding anpassen möchten, können Sie kurzerhand eine separate DB erstellen.

```
CREATE DATABASE ragdb2;  
\c ragdb2;  
CREATE USER rag_user WITH PASSWORD 'secret';  
GRANT ALL PRIVILEGES ON DATABASE ragdb TO rag_user;  
GRANT USAGE, CREATE ON SCHEMA public TO rag_user;  
GRANT ALL PRIVILEGES ON SCHEMA public TO rag_user;
```

Und den Eintrag in der Properties-Datei anpassen.

```
spring.datasource.url=jdbc:postgresql://localhost:5432/ragdb2
```

Allenfalls muss auch `VectorTableInitializer.java` angepasst werden.

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

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu15/aufgaben/a03?rev=1765843029>

Last update: **2025/12/16 00:57**

