

## LU12.A04 pgVector in SpringAI verwenden

Holen Sie sich die Tests mit einem Cherry-Pick auf f6c5bc4.

Erstellen Sie einen Datenbank-Benutzer und eine Database. Sie können dabei auch andere Namen bzw. Passwörter wählen.

```
CREATE DATABASE ragdb;  
\c ragdb;  
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;
```

Erstellen Sie ein `application.properties` unter `src/main/resources/` mit folgenden Einstellungen.

```
spring.datasource.url=jdbc:postgresql://localhost:5432/ragdb  
spring.datasource.username=rag_user  
spring.datasource.password=secret
```

Studieren Sie die Änderungen gründlich und stellen Sie sicher, dass der Test `StoreServiceTest.java` ohne Fehler ausgeführt werden kann.

```
./gradlew test
```

Ergänzen Sie allenfalls das `application.yml`, um sicherzustellen, dass beim Suchen dasselbe Embedding und dieselbe Distanz- beziehungsweise Ähnlichkeitsfunktion verwendet wird, wie beim Abspeichern.

```
vectorstore:  
  pgvector:  
    enabled: true  
    table-name: vector_store  
    indexType: "IVFFlat"  
    lists: 100  
    distance-type: "COSINE_DISTANCE"
```

An dieser Stelle sei angemerkt, dass der Test die echte Datenbank verwendet anstatt einer separaten Testdatenbank oder einem Mock. Damit nicht bei jeder Testausführung Duplikate hinzugefügt werden, wurde in der Klasse `DocumentFactory` die `id` deterministisch basierend auf dem Text erstellt. Dadurch werden Datensätze mit demselben Inhalt überschrieben anstatt neu angelegt.

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

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu12/aufgaben/a04?rev=1765841513>

Last update: 2025/12/16 00:31



