UNIWERSYTET IM. ADAMA MICKIEWICZA W POZNANIU
Wydział Matematyki i Informatyki

Wprowadzenie do Spring, ORM i usług sieciowych REST

Ostatnim razem

Ostatnio:

- omówiliśmy szczegółowo testy jednostkowe
- wstęp do programowania funkcyjnego

Dziś:

- wprowadzenie do Spring
  - CDI
  - adnotacje
  - dostęp do baz danych
- zahaczymy o REST

Wprowadzenie do Spring

Spring Framework:

- modułowa “kobyła” pozwalająca tworzyć aplikacje biznesowe
- zapewnie CDI (Container Dependency Injection)
- posiada moduły dostępu do danych
  - niskopoziomowo: JdbcTemplate
  - “klasycznie”: Hibernate
  - wysokopoziomowo: Spring Data JPA
- zapewnia moduły do komunikacji sieciowej (i nie tylko)
  - pokażemy tylko Spring Web (REST)

Wprowadzenie do Spring

Container Dependency Injection:

- wzorzec projektowy “Wstrzykiwanie zależności”
- cel: modułowość i skalowalność
- realizacja wzorca Dependency Inversion (“odwrócenie sterowania”), „D” w akronimie SOLID
- container, bo Spring dostarcza zarządzanie wstrzykiwaniem:
  - tworzenie obiektów
  - i to instancji konkretnych klas, często bez konieczności wskazywania o które nam chodzi
  - niekiedy potrafi nawet sam wygenerować ww. klasy
  - dla ciekawskich: stosowany jest wzorzec projektowy Proxy

Wprowadzenie do Spring

Adnotacje powiązane z CDI:

- stereotypy
  - @Component – komponent, de facto prosta klasa
  - @Service – klasa dostarczająca usługi, na ogół dostarczająca metod wysokopoziomowego dostępu do czegoś (lub przetwarzających dane z wielu komponentów, repozytoriów, etc.)
  - @Controller – kontroler usług internetowych
  - @Repository – repozytorium, dostęp do danych – uwaga często potrzeba przekształcenia tego co da repozytorium na “zjadliwy” model obiektowy – stąd mamy @Service

Wprowadzenie do Spring

Adnotacje powiązane z CDI:

- wstrzykiwanie zależności:
  - @Autowired – podstawowa adnotacja, pozwala na automatyczne wstrzyknięcie zależności w danym miejscu
  - @Required – wstrzyknięcie zależności do settera
-

ORM – po co to wszystko?

Niezgodność impedancji (ang. impedance mismatch):

- inny model danych w bazach re(we)lacyjnych niż w Javie
- brak podstawowych struktur danych (listy, mapy, ...)
- *teoretycznie* istnieją bazy obiektowe, ale… (znacie Państwo jakieś?)
- obiektowy model zawiera wiele redundancji (ale istnieją mechanizmy rozwiązujące ten problem)
- relacyjny model danych w założeniu powstał, aby optymalizować zużycie przestrzeni dyskowej (stąd popularność tzw. baz NoSQL, tj. baz nie-relacyjnych optymalizowanych raczej pod szybkość przetwarzania).

ORM – o co chodzi

ORM (Object-Relational Mapping, mapowanie relacyjno-obiektowe):

- mapuje klasy obiektowe na reprezentację w bazie danych
- robi to teoretycznie automatycznie (po to, żeby nie trzeba było niskpoziomowo konwertować danych)
- teoretycznie samo generuje potrzebny kod SQL
  - może tworzyć schemat bazy danych (nie polecam)
  - tworzy zapytania, aktualizacje, wstawia rekordy, może je także usuwać (i wszystko to często nieefektywnie)
- teoretycznie wystarcza do zamodelowania struktury obiektowej

ORM – podstawowe pojęcia

Pojęcia powiązane z ORM:

- DAO (Data Access Object):
  - klasa dostępowa do danych implementująca wywołania konkretnych zapytań (i zwracająca dane)
  - w świecie Spring repozytorium (repository)
- Encja (ang. Entity) – klasa implementująca mapowanie obiektu na konkretną tabelę w bazie
- DTO (Data Transfer Object):
  - klasa przechodnia mapująca encje do sensownego modelu obiektowego (i z powrotem)
  - możemy ich też używać do transferu danych przez sieć

Spring Data JPA

Literatura:

Spring Data JPA documentation

Accessing Data with JPA Tutorial

Raczej nie chcemy (jako projekt akademicki OK, ale...):
Accessing JPA Data with REST Tutorial

Co to takiego:

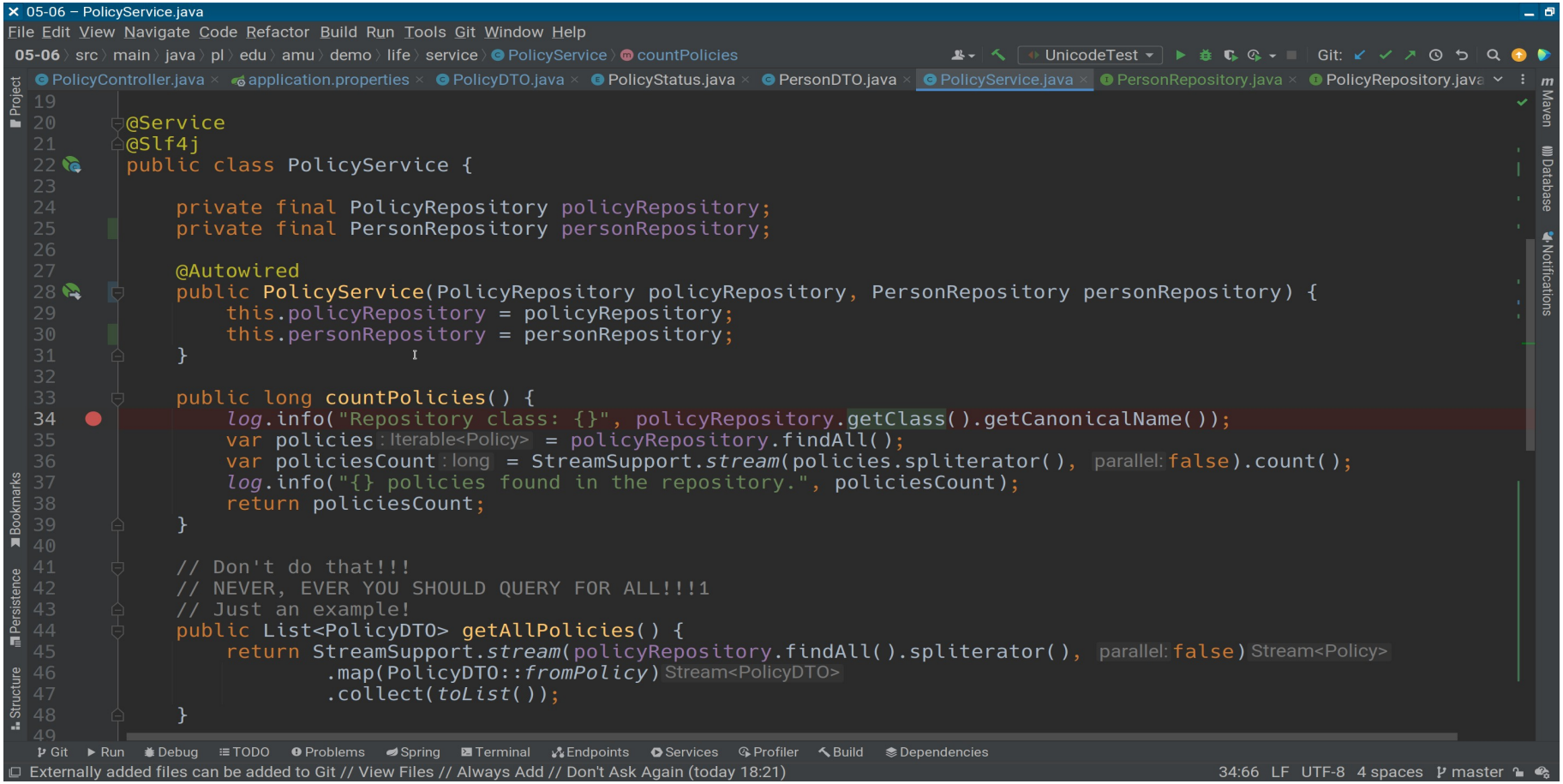
- JPA: Java Persistence API – standard dostępu do danych
- Spring Data JPA – moduł dostępu do danych Springa
- Spring ORM – to co faktycznie implementuje mapowanie
- Hibernate – to co (niestety) siedzi pod spodem Spring ORM

Wprowadzenie do Spring

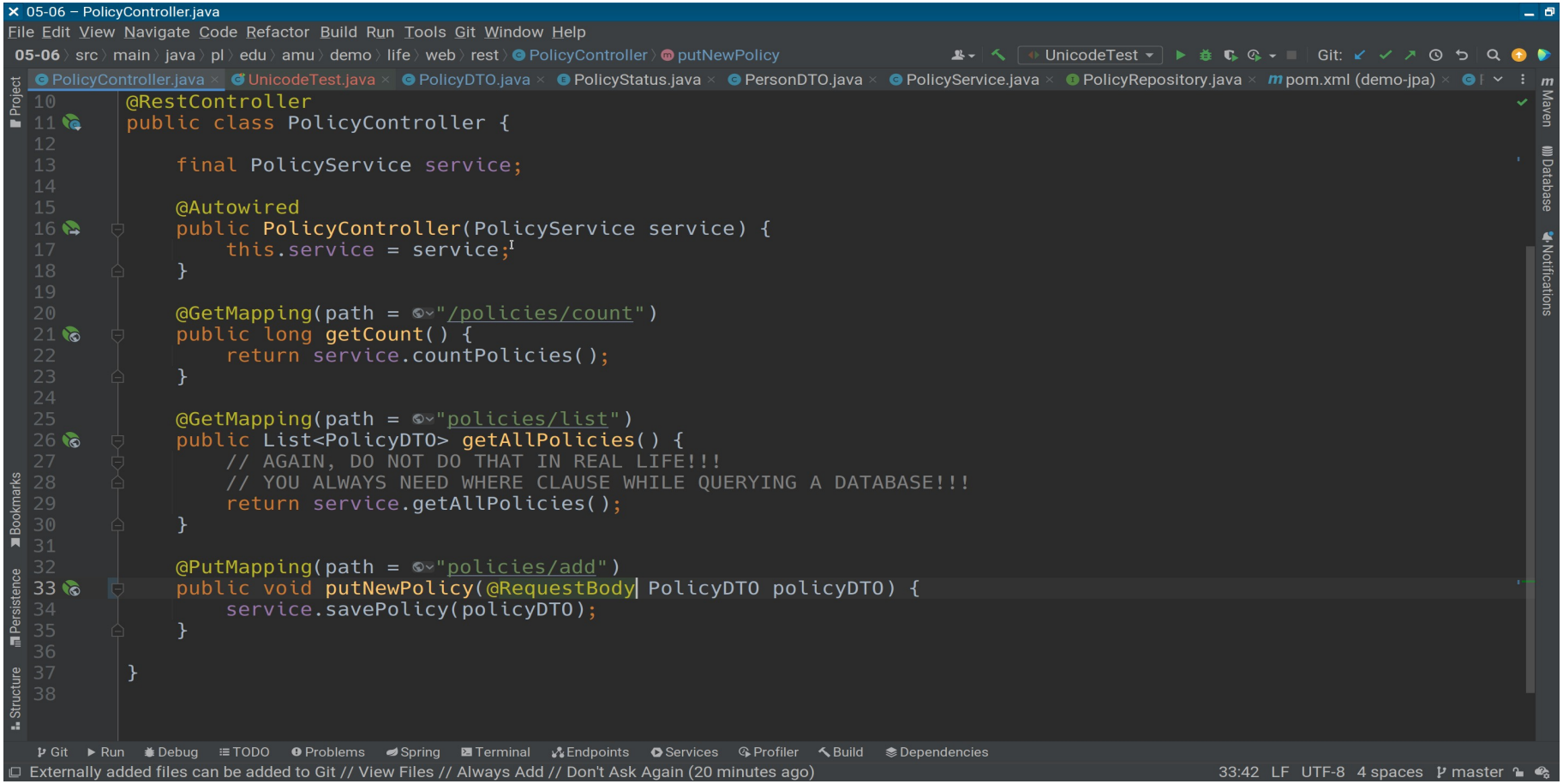
Adnotacje JPA:

- wstrzykiwanie zależności:
  - @Autowired – podstawowa adnotacja, pozwala na automatyczne wstrzyknięcie zależności w danym miejscu
  - @Required – wstrzyknięcie zależności do settera
-

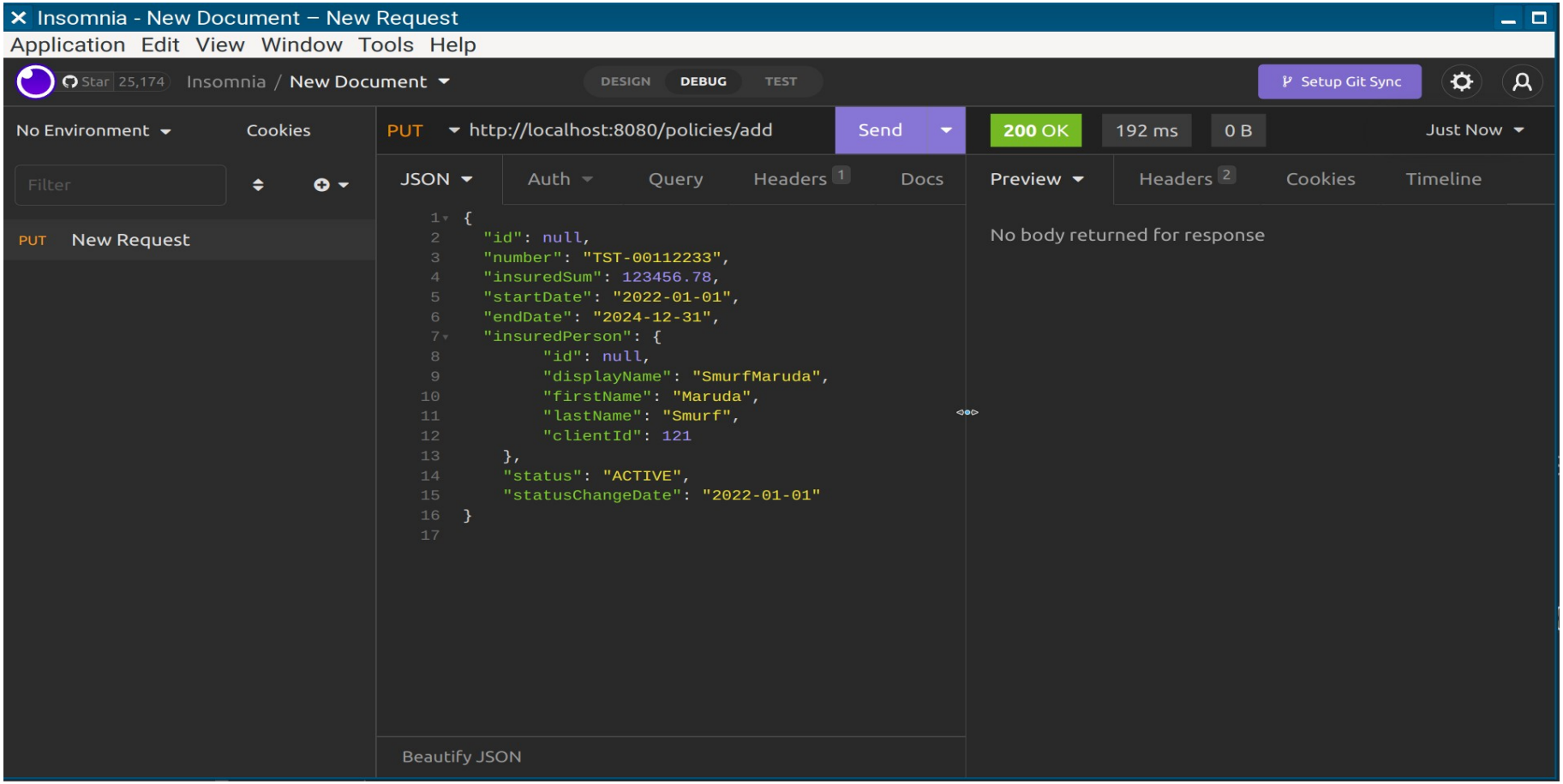
Przykład – Spring Data JPA



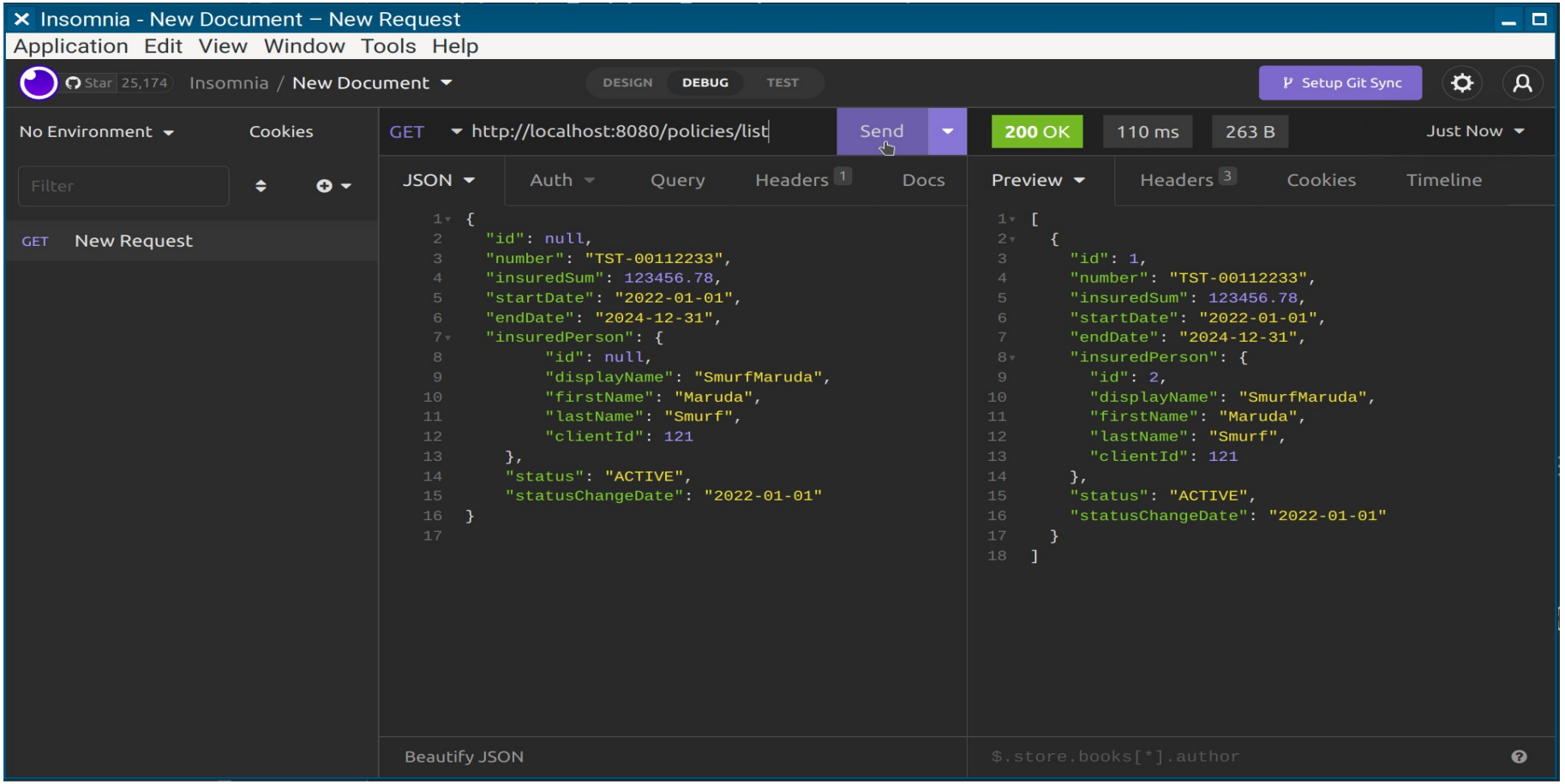
Przykład – Spring Data JPA



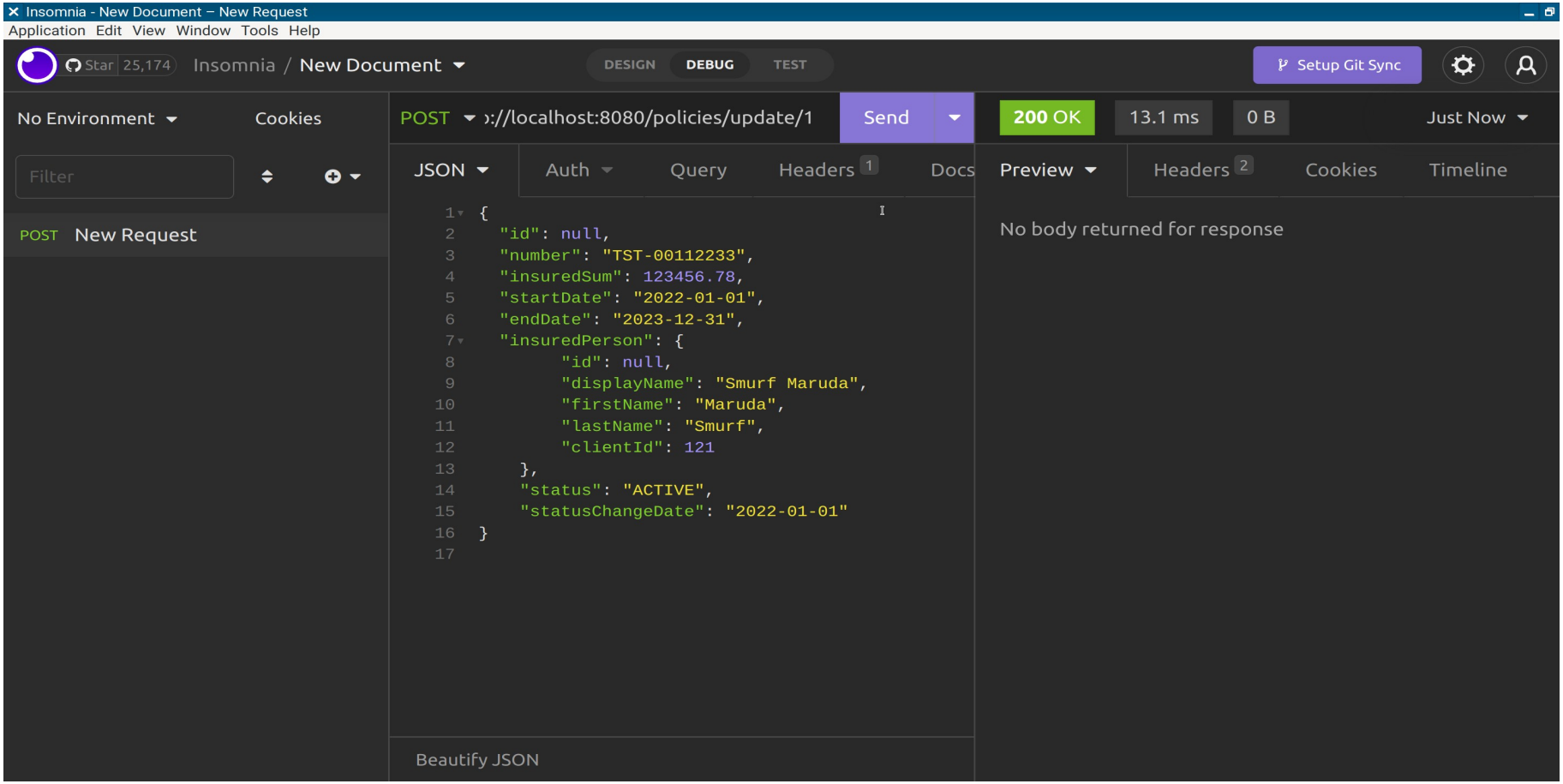
Przykład – Spring Data JPA



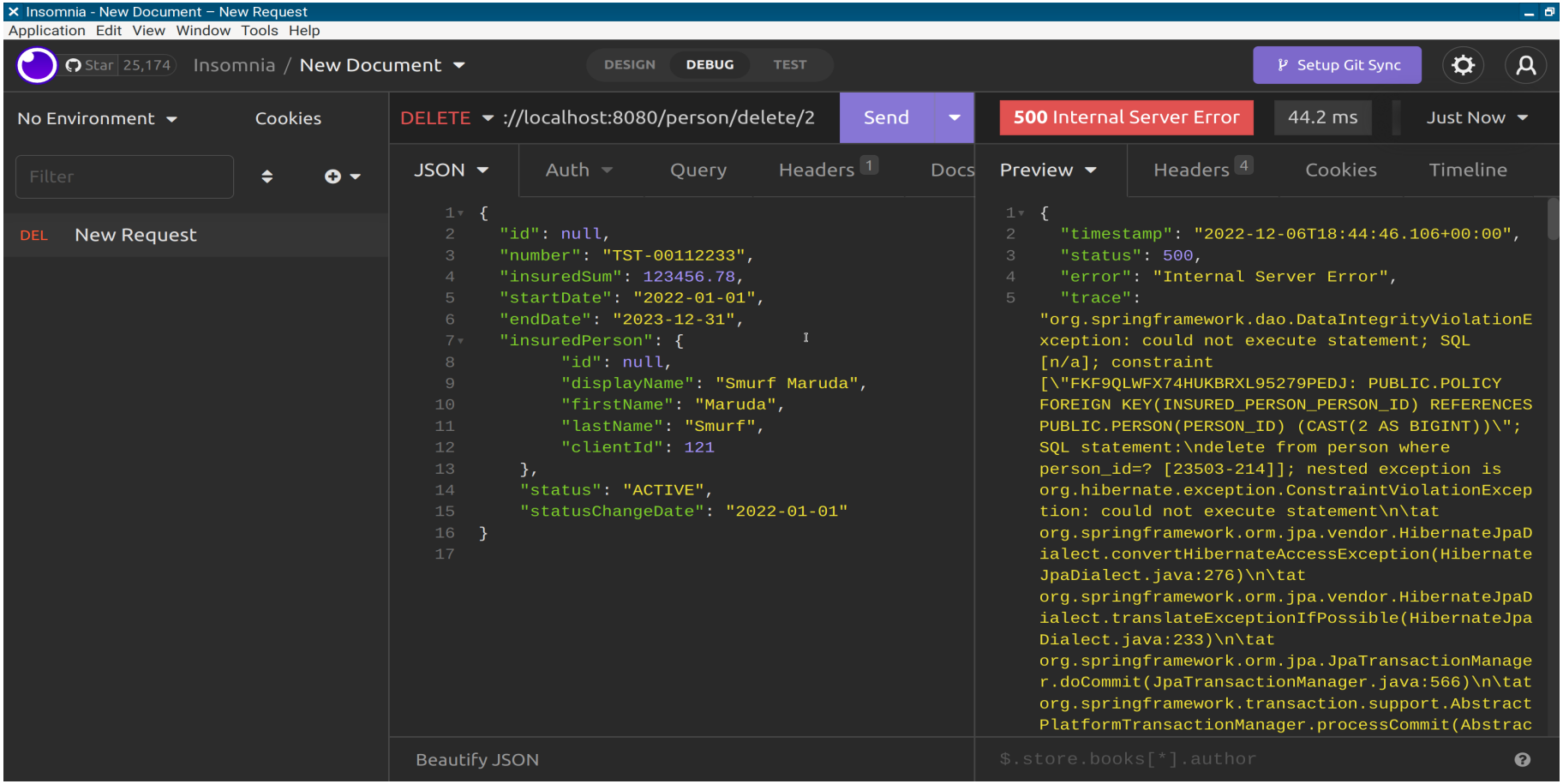
Przykład – Spring Data JPA



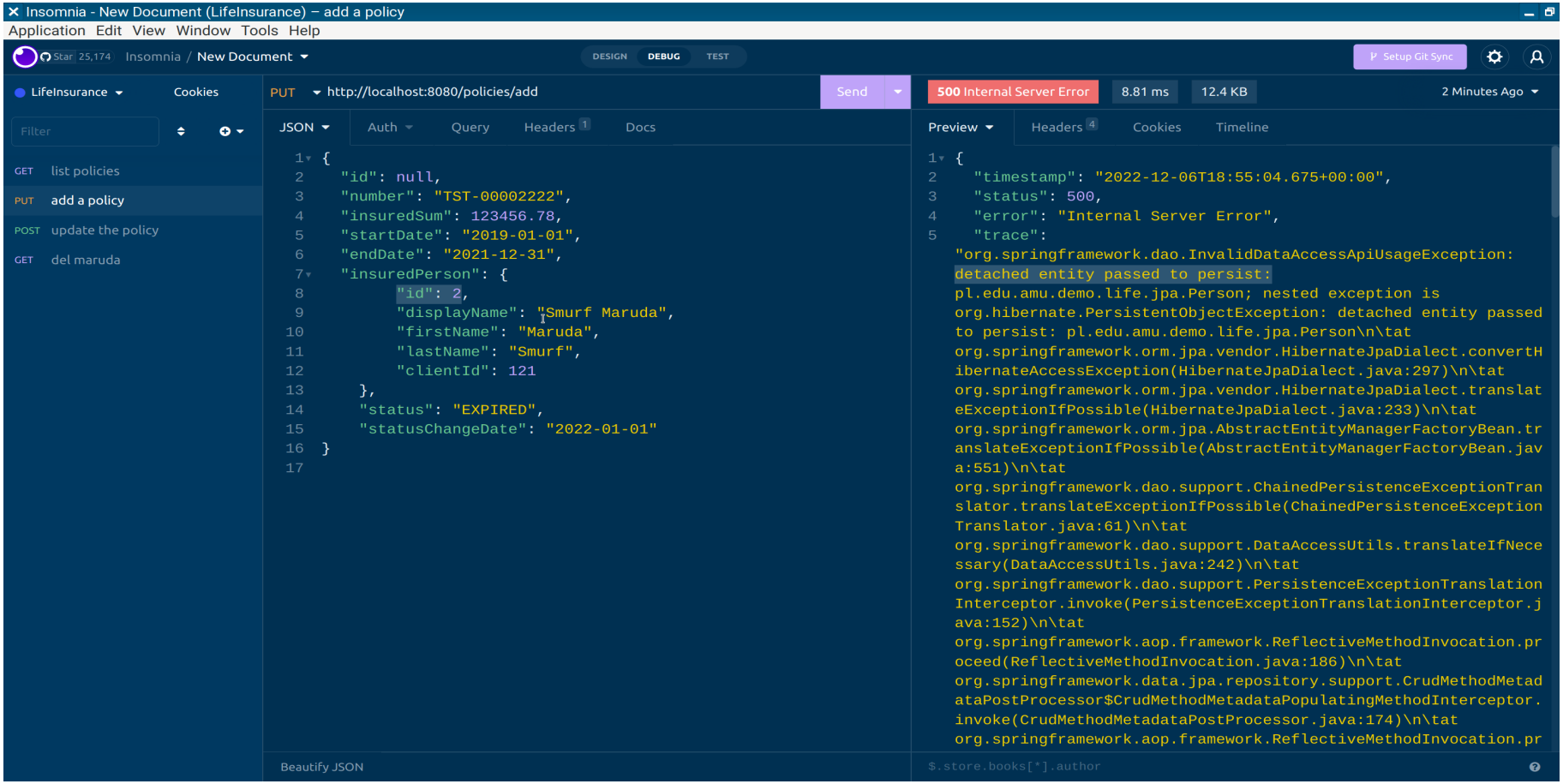
Przykład – Spring Data JPA



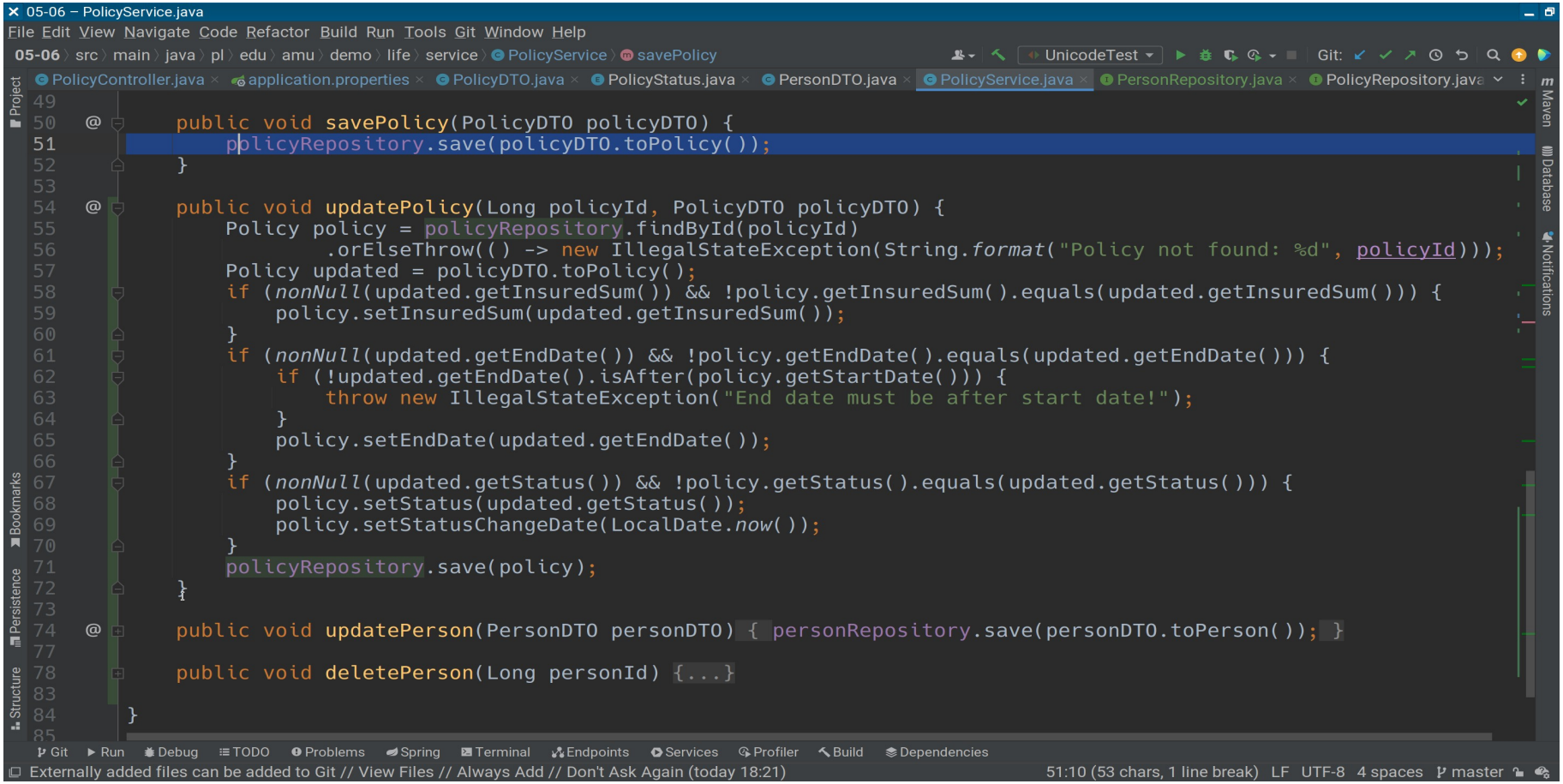
Przykład – Spring Data JPA



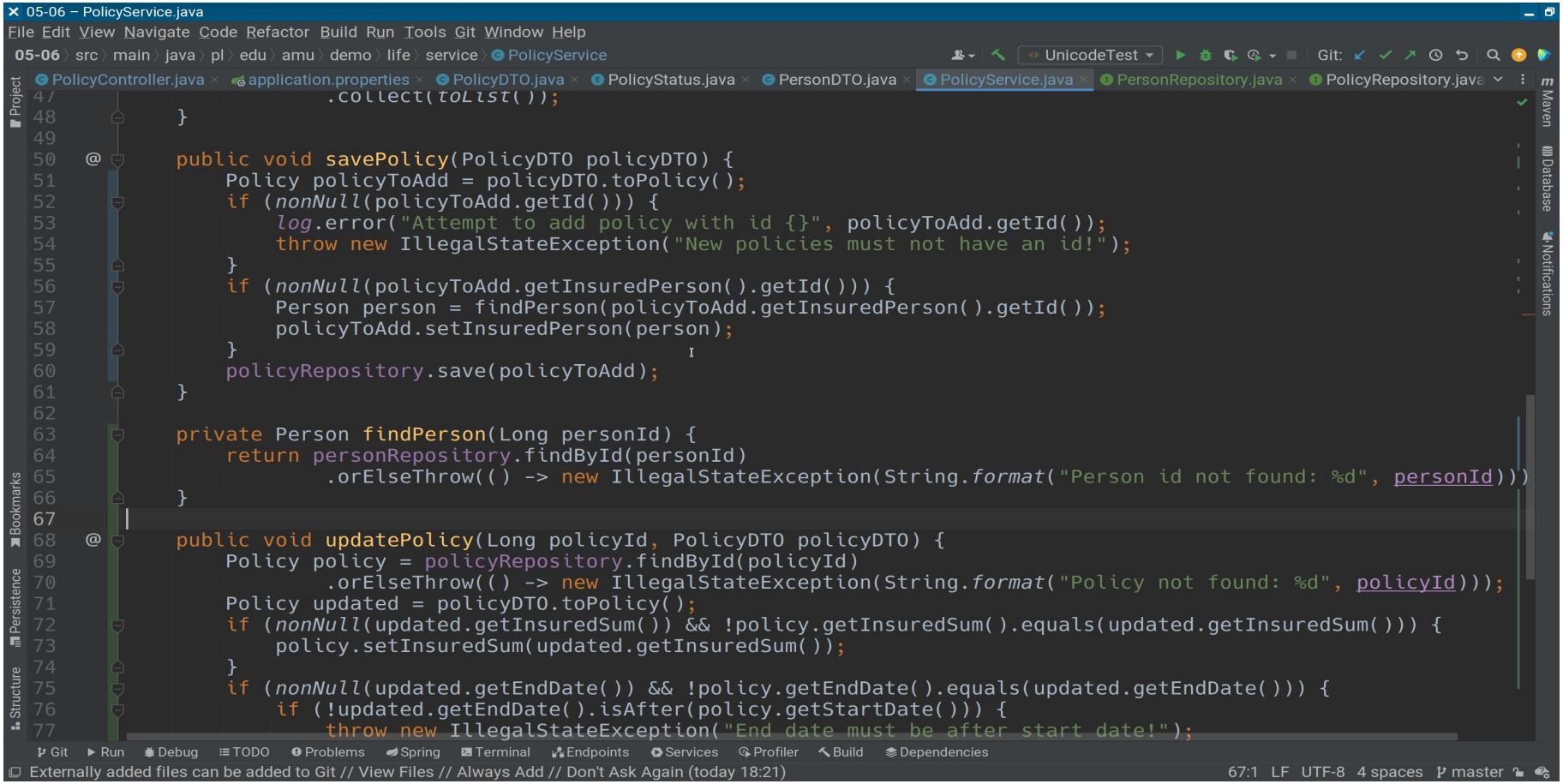
Przykład – Spring Data JPA



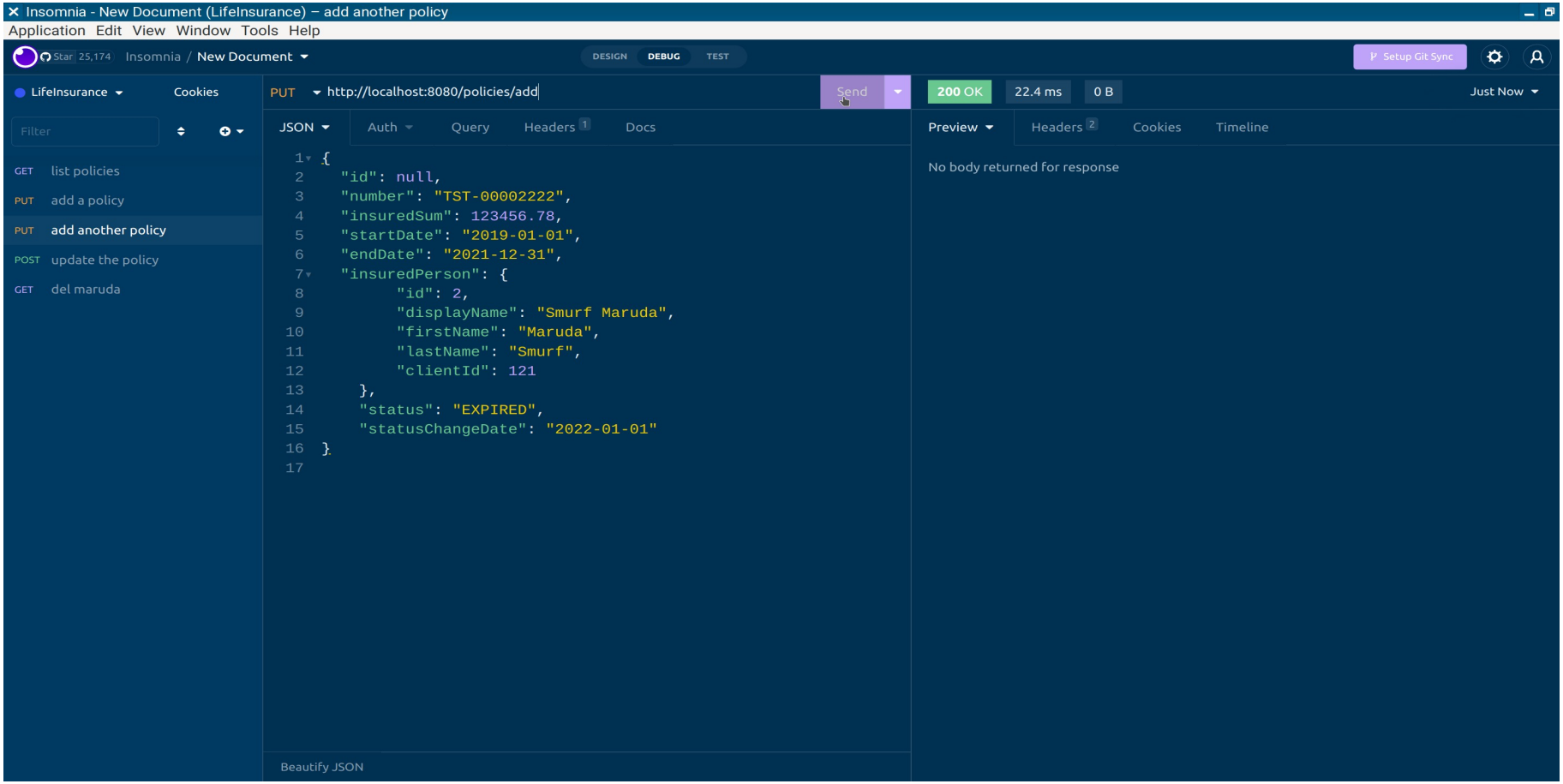
Przykład – Spring Data JPA



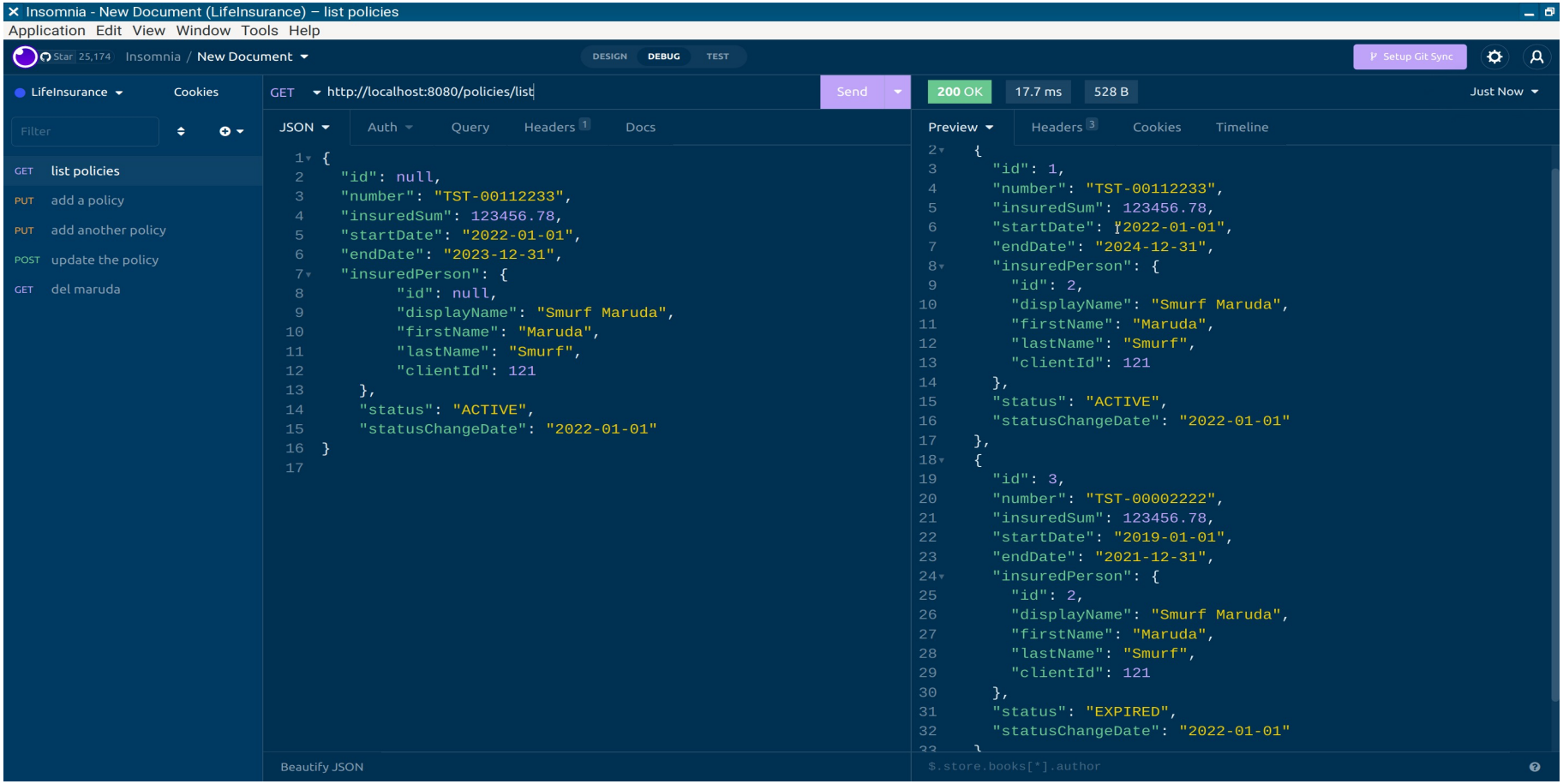
Przykład – Spring Data JPA



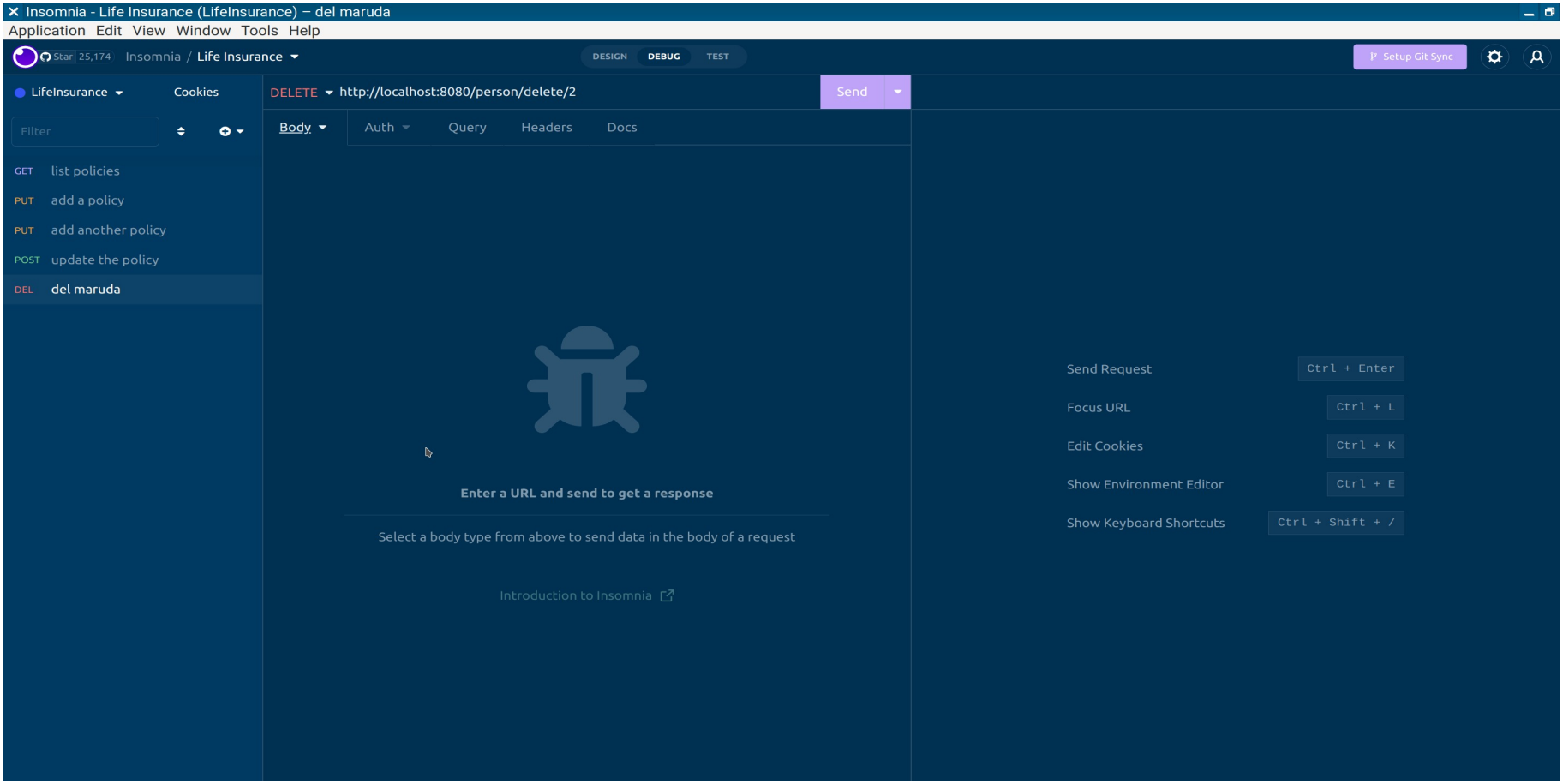
Przykład – Spring Data JPA



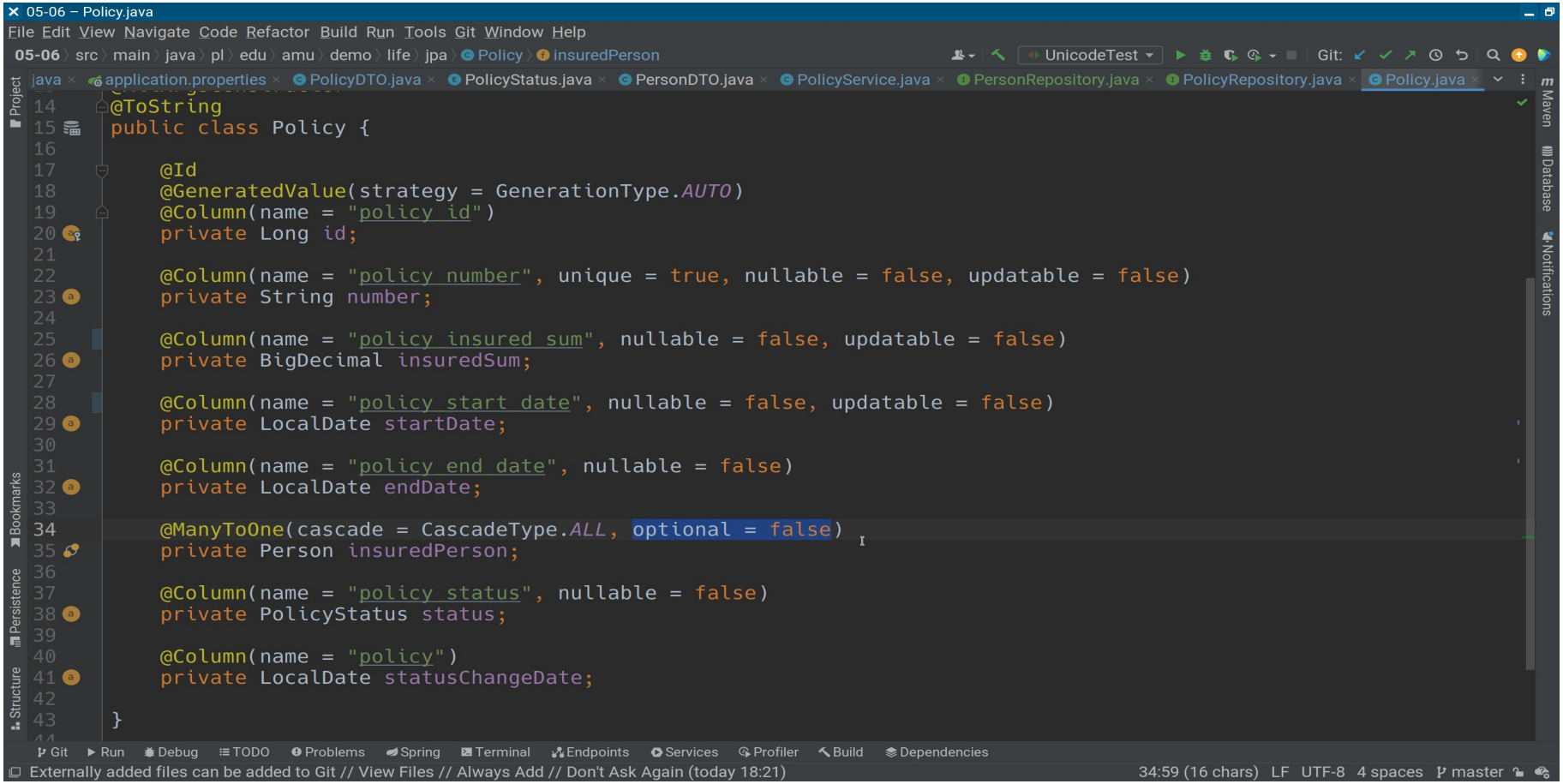
Przykład – Spring Data JPA



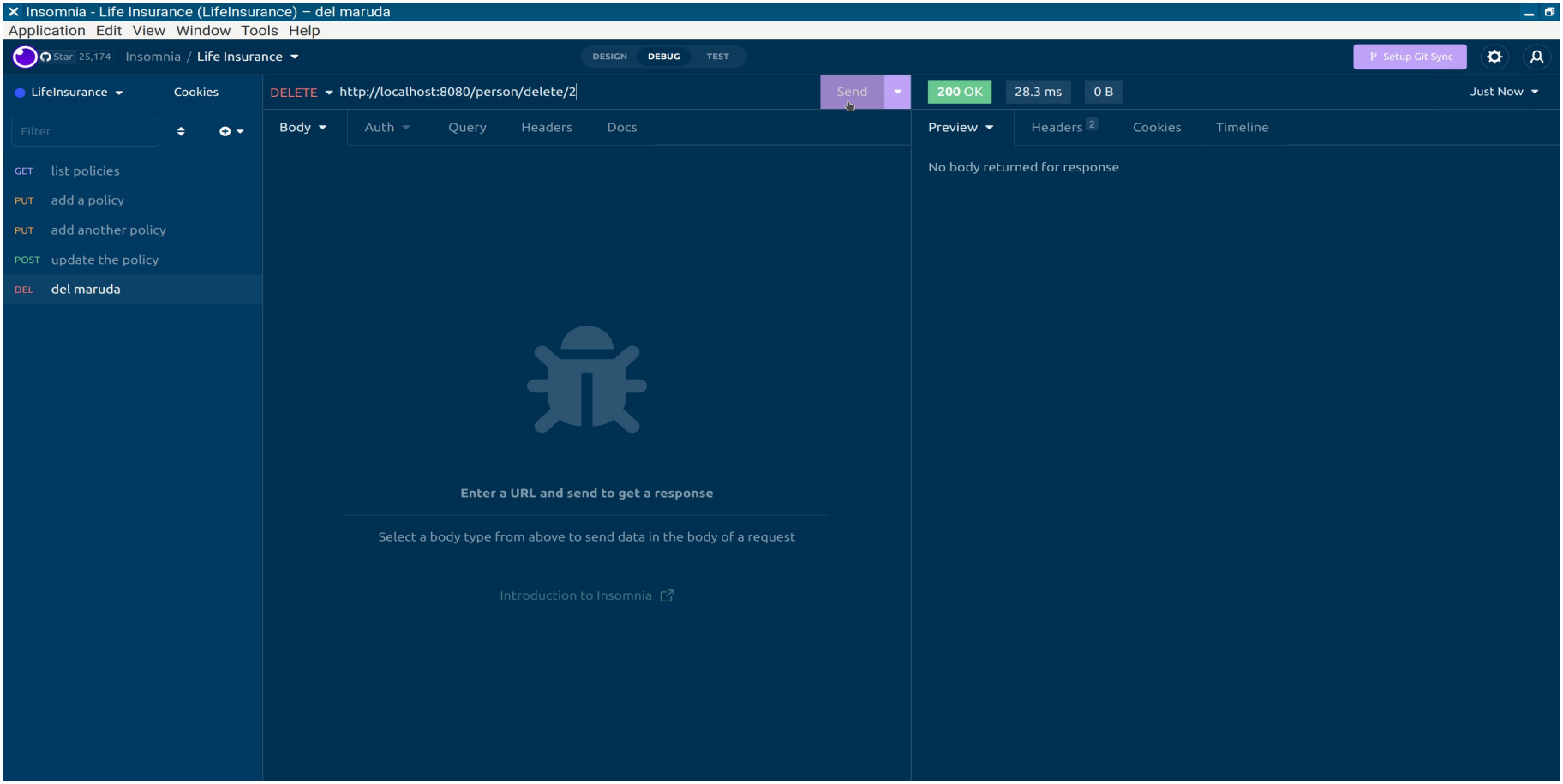
Przykład – Spring Data JPA



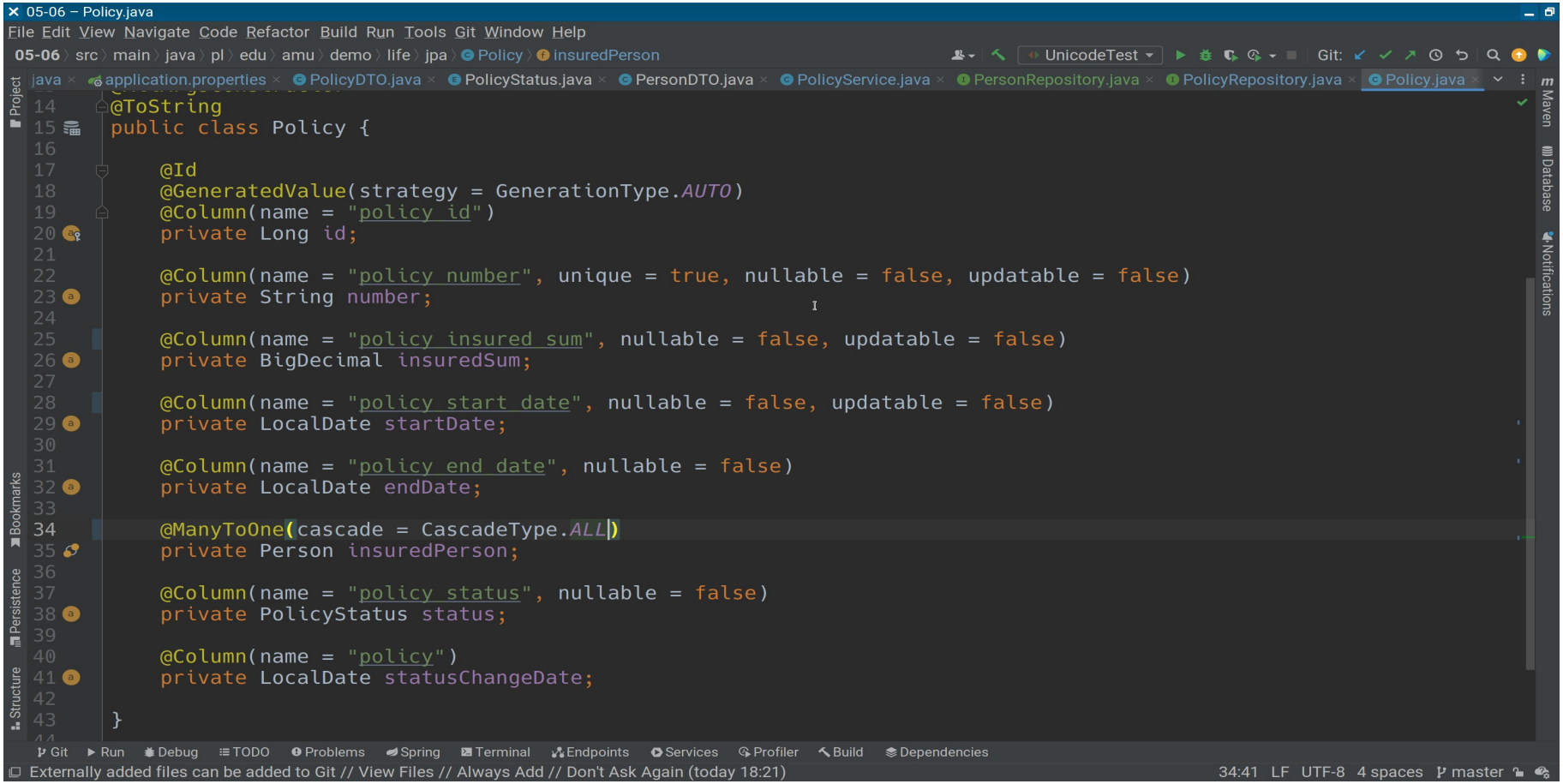
Przykład – Spring Data JPA



Przykład – Spring Data JPA



Przykład – Spring Data JPA



Przykład – Spring Data JPA

https://start.spring.io 120% Search

spring initializr

Project
- Gradle - Groovy
- Gradle - Kotlin
- Maven

Language
- Java
- Kotlin
- Groovy

Spring Boot
- 3.0.1 (SNAPSHOT)
- 3.0.0
- 2.7.7 (SNAPSHOT)
- 2.7.6

Project Metadata

Group	edu.amu.pl
Artifact	demo56
Name	demojakarta
Description	Demo project for Spring Boot
Package name	edu.amu.pl.demo
Packaging	Jar / War

Dependencies ADD DEPENDENCIES... CTRL + B

Lombok DEVELOPER TOOLS
Java annotation library which helps to reduce boilerplate code.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database SQL
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

GENERATE CTRL + ⏎ EXPLORE CTRL + SPACE SHARE...

Dodatek: Zaawansowane funkcje ORM (TODO: zrobię i zaktualizuję)

To wszystko na dziś.