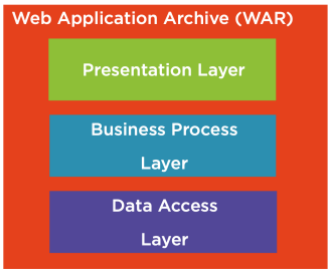


SpringBoot - Microservices – from Monolith to Microservices (Praxis)

Beschreibung
Wir werden nun hands on anlegen und eine Application von einer monolithic Arch. Zu einer MSA umschreiben.

Zeitbedarf:	2 Lektionen
Hilfsmittel:	SpringBoot Monolith to MS
Methode/Sozialform:	
Lernziele:	✓ **Monolithic Arch. Verstehen** ✓ **Microservice Arch. umschreiben** ✓ Aufgaben

Legende: Einzelarbeit, Partnerarbeit, Dokumentation, Code, Präsentation

Links:

Inhaltsverzeichnis

1 Decomposition Steps

Use the domain-driven design pattern to decompose a system!

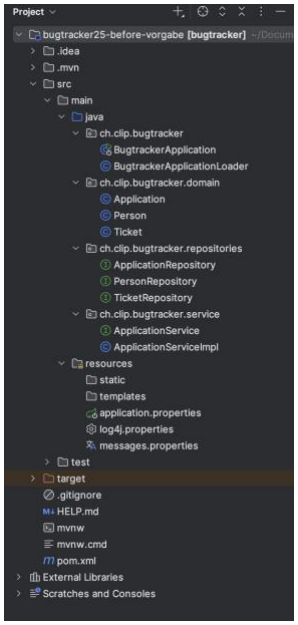
1.1 Steps

1. List of applications
   - Application Management Service List of people
   - User Management Service
2. List of tickets by application
   - Ticket Management Service
3. List of assigned tickets by person
   - Ticket Management Service

1.2 Bugtracker - Ausgangslage

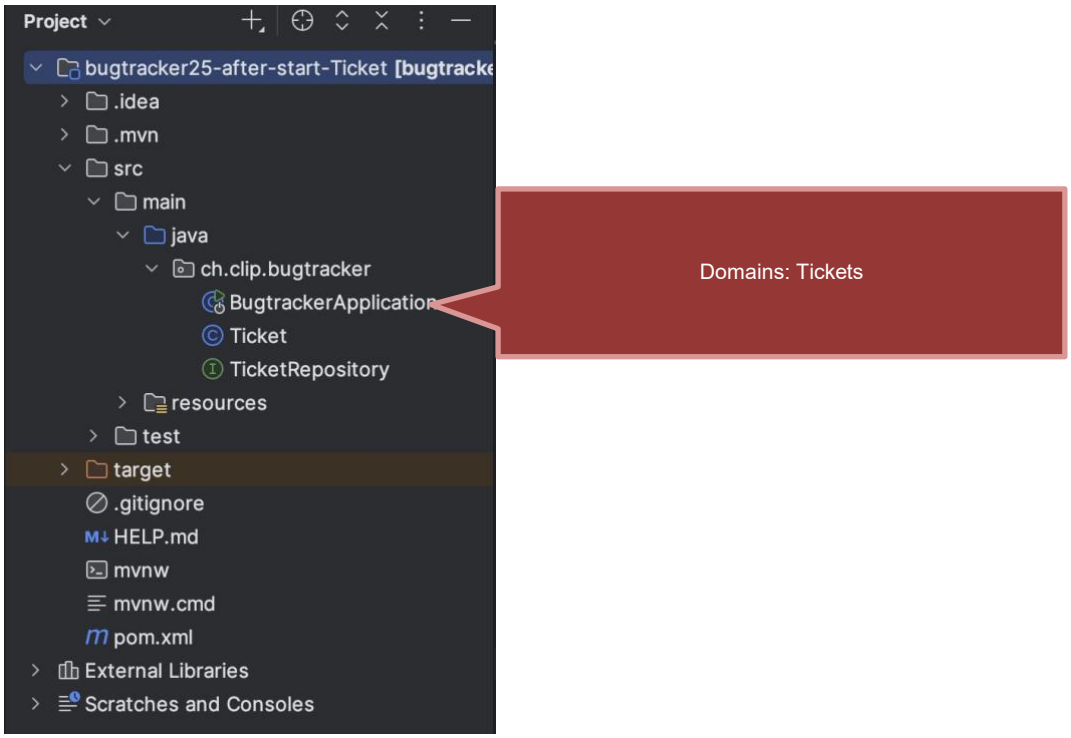
(TrackZilla)

Bugtracker-before definiert die alte monolithische Anwendung, es benutzt schon SpringBoot 3.3 und Java 17:



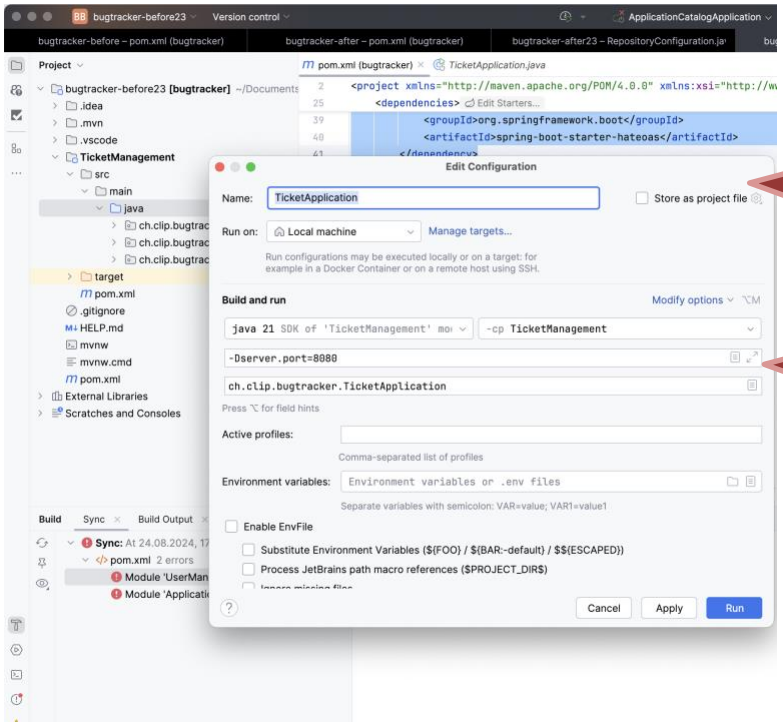
Domains: Applications, Persons, Tickets

Bugtracker-after-start ist der Start für die neue MSA – TrackZilla !



1.3 Bugtracker-before

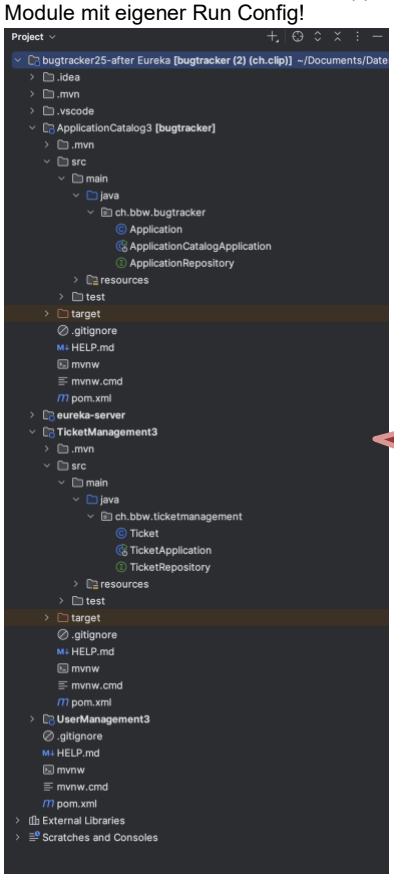
Die Ausgangslage für die MSA der TrackZilla App:



2 Aufgaben

2.1 Entwicklung

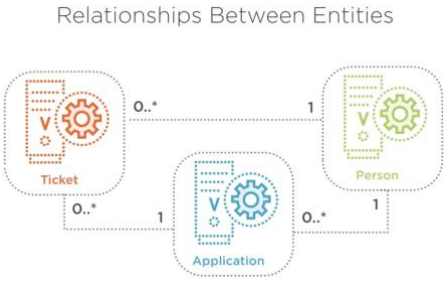
Erstellt nun für die anderen 2 MS ApplicationManagement und UserManagent analog zum TicketManagement diese MS in einem IntelliJ Module mit eigener Run Config!



Neue MS Arch. Mit IntelliJ Modulen in einem Projekt!

2.2 Complex Relationships between MS (Knacknuss)

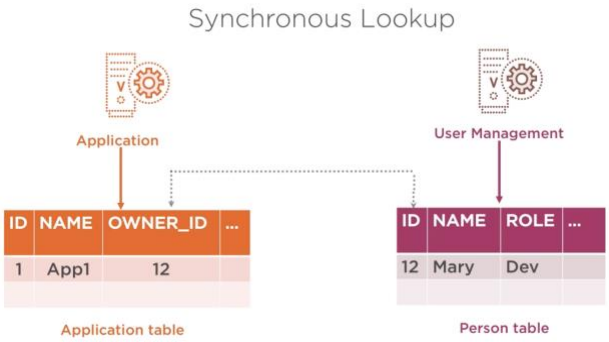
ApplicationCatalog has to call UserManagementService to . . . .



- Monolith Kommen wir nun zu einem Punkt in welchem wir eine Abfrage eines anderen Services machen müssen, in einem Monolith ist das kein Problem, wenn aber die DB's und MS getrennt sind kommt dies zu einer Herausforderung ;)

Synchronous Lookup between Application and UserManagement

Schaut euch dann den Feign Client an, das ist eine gute Lösung



We want to retrieve the Name and the Role of the User with the Owner_id!

2.2.1 Define a FeignClient

Dieses wird später erläutert!

2.3 Tests

Nehmt die PostMan Collection 11.1_MicroServices-SpringDataRest.postman_collection
Und testet ausführlich die 3 MicroServices!