JDBC demo

# INFO

Git-link: <https://github.com/25Rb/DemoApps>

# Inleiding

Deze applicatie maakt je wegwijs in de persistentie laag van een Java Spring applicatie.

# Leerdoelen

* Begrijpen hoe je beans annotatie gewijs definieerd.
* Begrijpen hoe je een bean instantie ophaalt.
* Een DAO aanmaken met database connectie
* Een service aanmaken waar we met een DAO data uit de database gaat capteren.
* Queries uitvoeren de database en de resultaten uitprinten.

# Opbouw

We starten vanuit de applicatie klasse waar we de **ApplicationContext** gaan aanmaken. Deze gaat de beans instantieren en bijhouden. Onze Service die we ge-annoteerd hebben met **@Service** is door de context aangemaakt en staat tot onze beschikking. Via de ***getBean(Class<T>)*** kunnen we de service ophalen en gebruiken. Deze service heeft een **DAO** object die de query results gaat afhandelen en deze doorgeeft aan de service.

De service heeft de DAO gekregen door het **Spring Framework**. De DAO heeft een **@Repository** annotatie en werd hierdoor ook als bean geinstantieerd. Door in de service de **@Autowired** annotatie te gebruiken werd de DAO door het framework in de service geinjecteerd.

Na deze stappen kunnen we de beschikbare opdrachten van service aanspreken. De resultaten hiervan zie je in de console verschijnen.

**Opmerking:** De volgorde die het framework afhandeld zijn te zien in de console. Hier zie je waar en wanneer een object geinstanteerd en geinjecteerd werd.

Hibernate demo

De Hibernate demo is een kleine uitbreiding op de JDBC demo. Hier gaat de applicatie adhv van Hibernate JPA gegevens uit de database halen.

**Opmerking:** De JDBC code staat telkens in commentaar onder de Hibernate code zo kan je de JDBC en de Hibernate code vergelijken.