# DB

, DB

DB , DB

.

.

- 
- (/)
- ( RDB / RDD)

.

- application.properties
- Repository ,JAVA package
- Configuration AConfigDB / BConfigDB

3 .

, DB

.



# 1.application.conf

## One DB

```
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.use_sql_comments=true
spring.jpa.properties.hibernate.format_sql=true
# For 1 Database
#spring.jpa.hibernate.ddl-auto=create
#spring.datasource.url=jdbc:mysql://localhost:3306/db_example
#spring.datasource.username=psmon
#spring.datasource.password=db1234
```

ddl-auto, . ddl-auto

. hiberate , .

## DB

```
# for Multiple Database

## Primary
app.datasource.primary.url=jdbc:postgresql://localhost:5432/db_example
app.datasource.primary.username=postgres
app.datasource.primary.password=db1234
app.datasource.primary.driver-class-name=org.postgresql.Driver
primary.hibernate.hbm2ddl.auto = create
primary.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect


## Second
app.datasource.second.url=jdbc:mysql://localhost:3306/db_example2
app.datasource.second.username=psmon
app.datasource.second.password=db1234
app.datasource.second.driver-class-name=com.mysql.jdbc.Driver
second.hibernate.hbm2ddl.auto = create
second.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
```

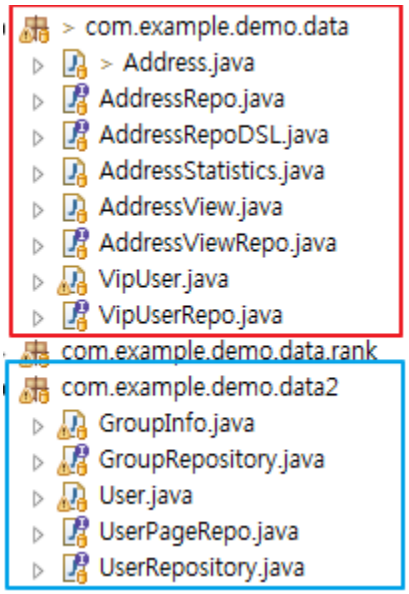
, db primary / second .

, .

primary/second DB ( Postgres / Mysql ) .

JPA DB , db driver-class-name .

## 2. Entity Package




DB , Entity/Repository JAVA package

. data / data2

data = primary , data2 = second .

entity repository , 2DB 4

.

## 3. DataBaseConfigXX

```
package com.example.demo.config;


import java.util.HashMap;
import java.util.Map;
import java.util.Properties;

import javax.persistence.EntityManagerFactory;
import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.boot.autoconfigure.jdbc.DataSourceBuilder;
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.boot.orm.jpa.EntityManagerFactoryBuilder;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Primary;
import org.springframework.core.env.Environment;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import org.springframework.orm.jpa.JpaTransactionManager;
import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;
import org.springframework.transaction.PlatformTransactionManager;
import org.springframework.transaction.annotation.EnableTransactionManagement;


primary db  db(    )

db    second   db .

 Class  DB ,  Db    .


 DB  Class

- DataBaseConfigPrimary
- DataBaseConfigSecond

## Primary DB
```

```
@Configuration
@EnableTransactionManagement
@EnableJpaRepositories(
  basePackages = {"com.example.demo.data"})
public class DataBaseConfigPrimary {

          @Primary
          @Bean(name = "dataSource")
          @ConfigurationProperties(prefix = "app.datasource.primary")
          public DataSource dataSource() {
            return DataSourceBuilder.create().build();
          }

          @Primary
          @Bean(name = "entityManagerFactory")
          public LocalContainerEntityManagerFactoryBean entityManagerFactory(
                         EntityManagerFactoryBuilder builder,
                         @Qualifier("dataSource") DataSource dataSource,
                         Environment env) {
                Map<String, Object> properties = new HashMap<String, Object>();
                properties.put("hibernate.hbm2ddl.auto", env.getRequiredProperty("primary.hibernate.hbm2ddl.
auto"));
                properties.put("hibernate.dialect", env.getRequiredProperty("primary.hibernate.dialect"));

            return builder
              .dataSource(dataSource)
              .packages("com.example.demo.data")
              .persistenceUnit("primary")
              .properties(properties)
              .build();
          }

          @Primary
          @Bean(name = "transactionManager")
          public PlatformTransactionManager transactionManager(
            @Qualifier("entityManagerFactory") EntityManagerFactory entityManagerFactory) {
            return new JpaTransactionManager(entityManagerFactory);
          }

}
```

:

- com.example.demo.data :
- prefix = "app.datasource.primary" : application.conf  prefix
- "hibernate.hbm2ddl.auto", "create"

## Second DB

```
@Configuration
@EnableTransactionManagement
@EnableJpaRepositories(
  entityManagerFactoryRef = "secondEntityManagerFactory",
  transactionManagerRef = "secondTransactionManager",
  basePackages = {"com.example.demo.data2"})
public class DataBaseConfigSecond {

          @Bean(name = "secondDataSource")
          @ConfigurationProperties(prefix = "app.datasource.second")
          public DataSource secondDataSource() {
            return DataSourceBuilder.create().build();
          }

          @Bean(name = "secondEntityManagerFactory")
          public LocalContainerEntityManagerFactoryBean secondEntityManagerFactory(
                         EntityManagerFactoryBuilder builder,
                         @Qualifier("secondDataSource") DataSource secondDataSource,
                         Environment env) {
               Map<String, Object> properties = new HashMap<String, Object>();
               properties.put("hibernate.hbm2ddl.auto", env.getRequiredProperty("second.hibernate.hbm2ddl.
auto"));
               properties.put("hibernate.dialect", env.getRequiredProperty("second.hibernate.dialect"));


            return builder
              .dataSource(secondDataSource)
              .packages("com.example.demo.data2")
              .persistenceUnit("second")
              .properties(properties)
              .build();
          }

          @Bean(name = "secondTransactionManager")
          public PlatformTransactionManager secondTransactionManager(
            @Qualifier("secondEntityManagerFactory") EntityManagerFactory secondEntityManagerFactory) {
            return new JpaTransactionManager(secondEntityManagerFactory);
          }

}
```

:

- com.example.demo.data2 :
- prefix = "app.datasource.second" : application.conf  prefix
- "hibernate.hbm2ddl.auto", "create" : ddl
- secondEntityManagerFactory / secondTransactionManager / secondDataSource : db

'

.

```
Initialized JPA EntityManagerFactory for persistence unit 'primary'
Initialized JPA EntityManagerFactory for persistence unit 'second'
```

.

http://git.webnori.com/projects/WEBF/repos/spring_jpa/browse