

# Описание

---

Модуль реактивной работы с СУБД

## Общие настройки

---

Подключение к БД через настройки application.yml

```
jprime:
  storage:
    <уникальный код хранилища>:
      title: <описание хранилища>
      type: r2dbc
      url: <урл подключения>
      username: <логин>
      password: <пароль>
      aliases:
        <синоним 1>: <описание хранилища>
        <синоним 2>: <описание хранилища>
        <синоним 3>: <описание хранилища>
```

При запуске приложения обязательно отключить автоконфигурацию r2dbc `R2dbcAutoConfiguration`

```
@EnableScheduling
@SpringBootApplication(
    scanBasePackages = {
        "mp"
    },
    exclude = {
        R2dbcAutoConfiguration.class,
        ...
    }
)
public class Application {

}
```

## Основной принцип

---

Для построения взаимодействия с СУБД через JPrime API и метаописание используется `querydsl-r2sql`

## Хранилища

---

### Пул соединения

На каждое хранилище создается отдельный пул соединений.

При использовании алиасов можно описать дополнительные хранилища с теми же настройками, что и основное. Пул соединений в это случае будет общий

```
jprime:
  storage:
    test1:
      title: Хранилище 1
      type: r2db
      driver: <jdbc драйвер>
      url: <урл подключения>
      username: <логин>
      password: <пароль>
      aliases:
        test2: Хранилище 2
```

При необходимости разделить пулы соединений рекомендуется описать два разных хранилища, даже, если они используют одну и ту же БД

```
jprime:
  storage:
    test1:
      title: Хранилище 1
      type: r2db
      driver: <jdbc драйвер>
      url: <урл подключения>
      username: <логин>
      password: <пароль>

    test2:
      title: Хранилище 2
      type: r2db
      driver: <jdbc драйвер>
      url: <урл подключения>
      username: <логин>
      password: <пароль>
```

## Получение хранилища по коду

```
@Autowired
private R2DbRepositoryStorage dbStorage;
....
R2DbStorage dbStorage = dbStorage.getStorage(<уникальный код хранилища>);
```