12-2 注入JdbcTemplate進行資料查詢應用

專案配置Spring JdbcTemplate API

Pom.xml配置Spring JdbcTemplate API

配置application.properties DataSource 組態Config

```xml
<dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>10.2.1.jre11</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
    <version>2.7.2</version>
</dependency>
```

```
#SQL Server連接組態設定
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
#設定連接字串url
spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName=NORTHWND;encrypt=true;trustServerCertificat
e=true;application name=hr
#設定登入帳號與密碼
spring.datasource.username=sa
spring.datasource.password=1111
```

DataSource Bean配置

- 配置Configuration類別
- 使用Method建立一個DataSource物件
- 使用@Bean定義
- 借助Environment物件取出application.properties DataSource屬性設定

```
public class DBConfig {
    //Attribute 注入環境物件 可擷取到application.properties設定
    @Autowired
    private Environment env;
    //寫上一個方法 產生一個DataSource物件
    //產生的DataSource可以應用系統共用 生產個別使用Connection連接物件
    @Bean   //定義成一個Spring Bean 進入Spring Container
    @Scope(ConfigurableBeanFactory.SCOPE_SINGLETON)  //singletion 獨一模式
    public SQLServerDataSource dataSource() {
        System.out.println("建立起一個共用的DataSource物件");
        //建構一個DataSource物件
        SQLServerDataSource datasource=new SQLServerDataSource();
        //設定屬性 連接字串 登入帳號與密碼

        datasource.setURL(env.getProperty("spring.datasource.url"));
        datasource.setUser(env.getProperty("spring.datasource.username"
        datasource.setPassword(env.getProperty("spring.datasource.passw
        System.out.println(datasource.getURL());
        return datasource;
```

注入JdbcTemplate到控制項中

使用@Autowired注入JdbcTemplate

Spring JdbcTemplate實現DAO設計模式

```
//使用JEE JAX-RS API規範
@RestController
public class TibameCustomersService {
    @Autowired
    private JdbcTemplate jdbcTemplate;
    //注入JPA 自訂介面Repository
```

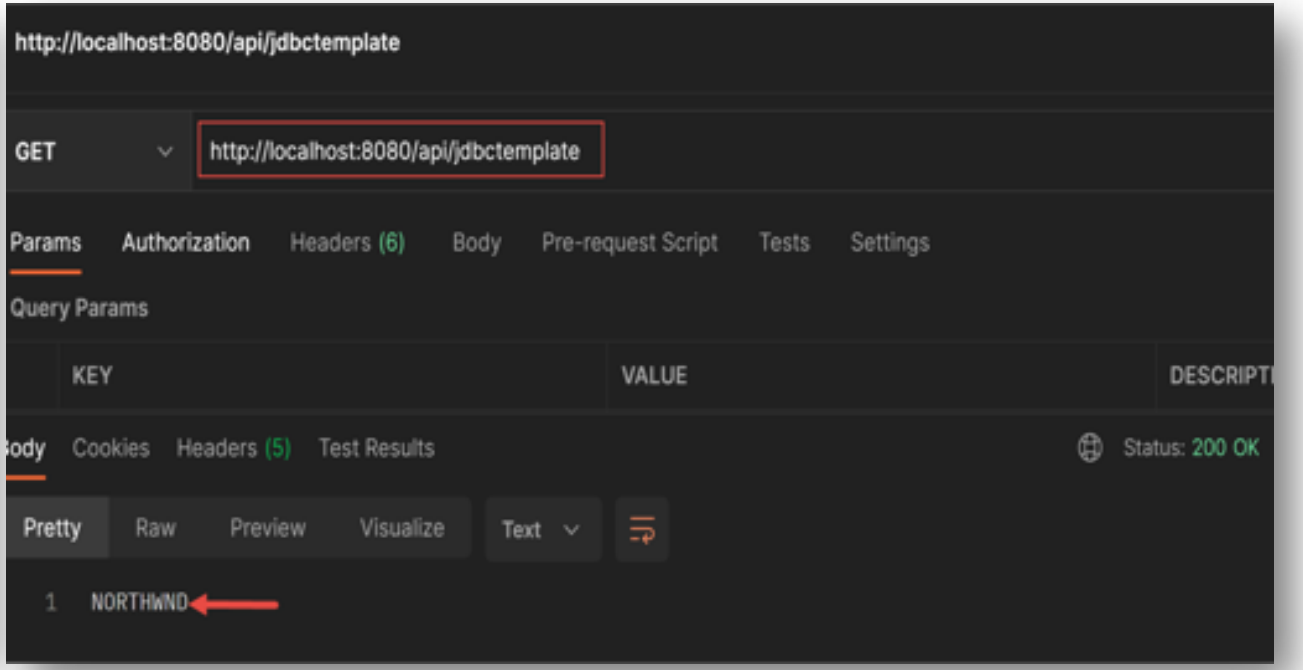
Spring Boot Web MVC

RESTful Service檢視JdbcTemplate Ioc DataSource連接狀態

檢視注入的JdbcTemplate注入控制反轉的DataSource物件

檢視連接的資料庫名稱

```java
//檢視注入的JdbcTemplate注入
@GetMapping(path="/api/jdbctemplate")
public String jdbcTemplateInject() {
    String dbName=null;;
    try {
        dbName = jdbcTemplate.getDataSource()
                .getConnection().getCatalog();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return dbName;
}
```



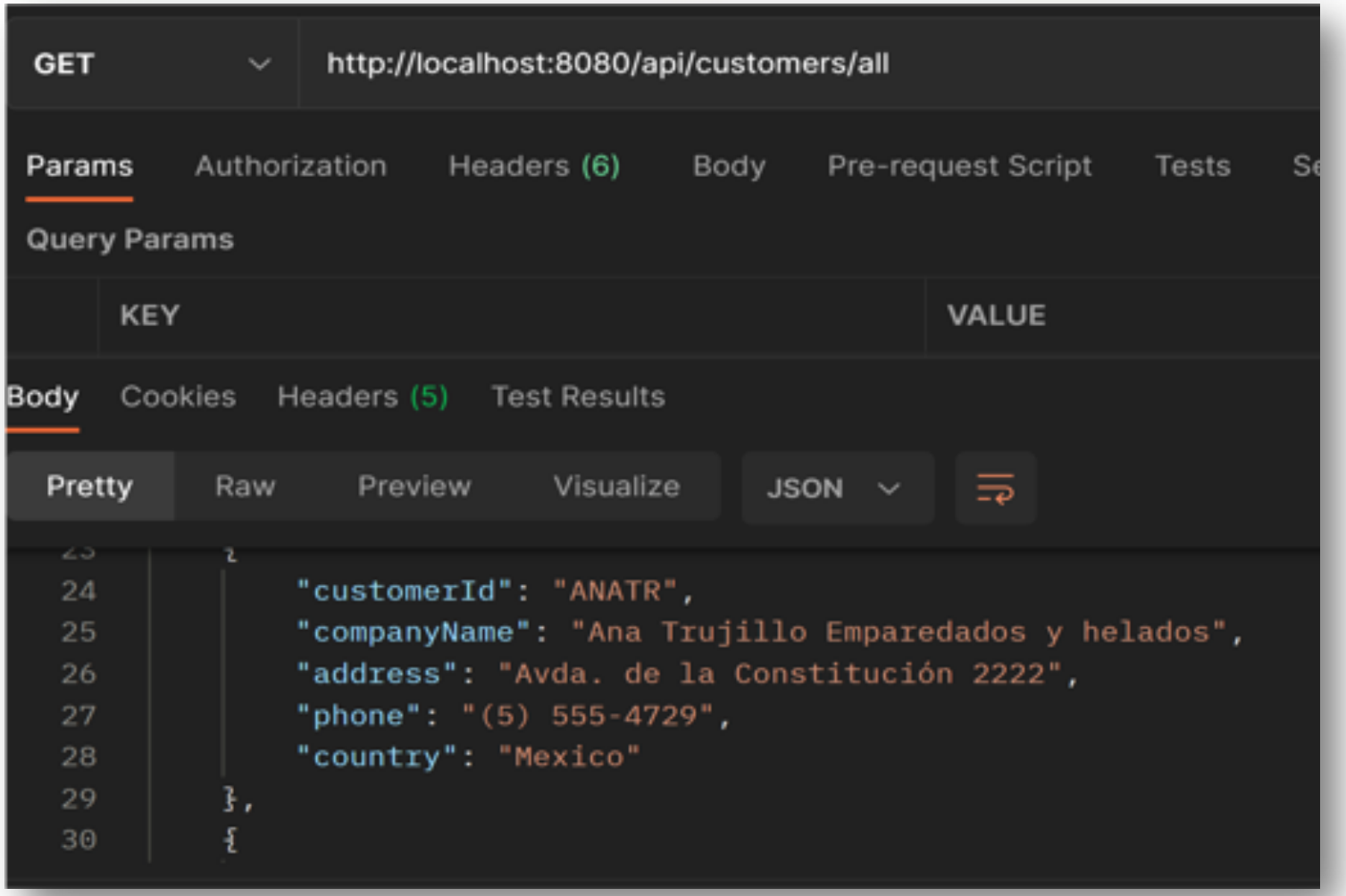
RESTful Service 查詢客戶所有資料應用

使用JdbcTemplate.query method

配合自訂JavaBean與BeanPropertyRowMapper將查詢結果反轉成List物件

```
//使用JdbcTemplate 查詢客戶資料
@GetMapping(path="/api/customers/all",produces="application/json")
public List<Customers> customersQuery() {
    String sql="Select CustomerID,CompanyName,"
            + "Address,Phone,Country From Customers";
    BeanPropertyRowMapper<Customers> mapper=
            BeanPropertyRowMapper.newInstance(Customers.class);
    List<Customers> result=jdbcTemplate.query(sql,mapper);
    return result;
}
```

RESTful Service 查詢客戶所有資料應用Demo



總結：12-2 注入JdbcTemplate進行資料查詢應用

了解注入JdbcTemplate與相關架構之後，且操作JdbcTemplate相關功能進行查詢作業，接下來我們來了解JPA如何在RESTful服務中進行資料新增作業。