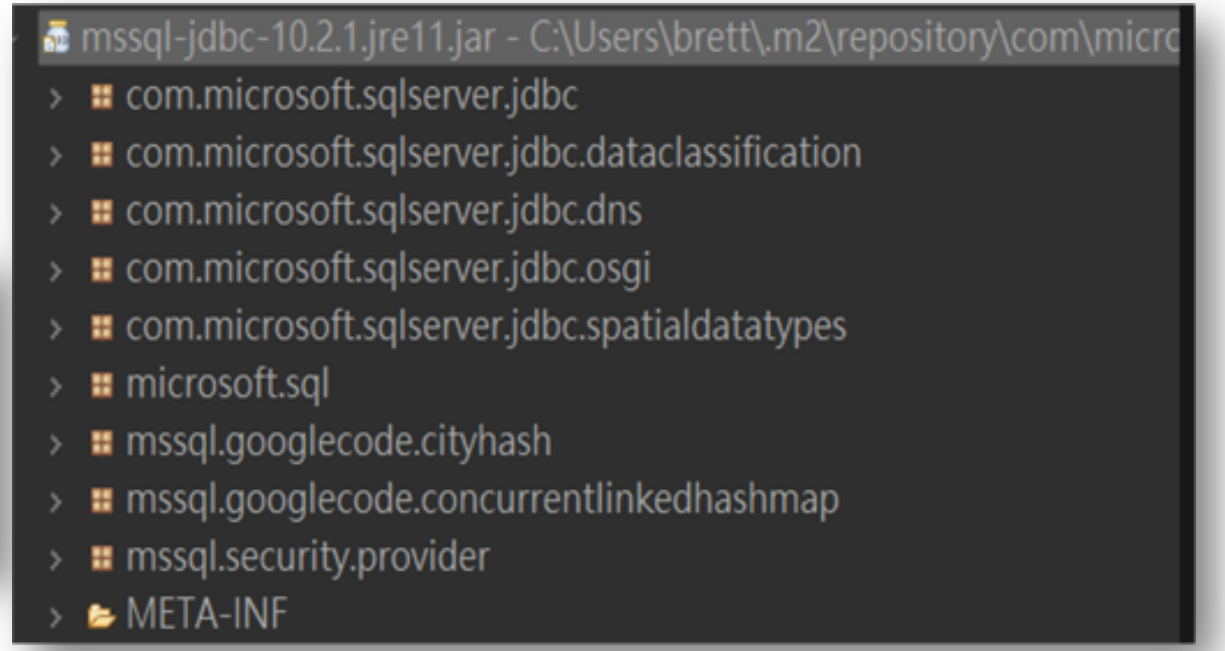
9-2 Spring Boot 配置DataSource

配置SQL Server Database API依賴模組

使用pom.xml進行SQL Server Database API配置

```
<dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>10.2.1.jre11</version>
</dependency>
```



使用application.properties配置DataSource組態

SQL Server Jdbc Database API配置

application.properties配置for SQL Server組態

DataSource為連接物件工廠，配置要件

- DriverClassName
- url(連接字串)
- Database Name
- User Name
- Password

```
#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;
trustServerCertificate=true;application name=hr
#設定登入帳號與密碼
spring.datasource.username=sa
spring.datasource.password=1111
```

使用Class 方法配置一個DataSource Spring Bean

使用@Configuration配置Spring Bean元件的組態類別

透過Method產生配置在Spring Container中的Spring Bean(元件)

建立一個DataSource個體物件

動態注入application.properties組態項目

```
package com.tibame.component;

import javax.annotation.Resource;

//資料庫有關的配置檔(class)
@Configuration
public class DBConfig {
    //Attribute 注入環境物件 可獲取到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 只有一個物

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.password"));
    System.out.println(datasource.getURL());
    return datasource;
}
```

Web啟動啟動時產生 DataSource Bean注入Spring Container

```java
//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 只有一個物

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.password"));
    System.out.println(datasource.getURL());
    return datasource;
}
```

```
Initializing Spring embedded WebApplicationContext
Root WebApplicationContext: initialization completed in 879 ms
```

```
2022-08-24 08:02:58.658   INFO 26596 --- [
2022-08-24 08:02:58.721   INFO 26596 --- [
2022-08-24 08:02:58.722   INFO 26596 --- [
建立起一個共用的DataSource物件
```

注入DataSource至MVC Controller中應用

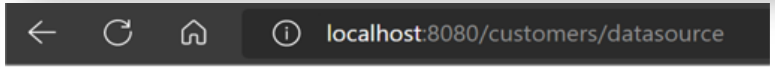
規畫一個Controller Class

使用@Autowired注入依賴物件

使用型別注入依據

```java
//客戶資料維護控制器
@Controller
@RequestMapping(path="/customers")
public class CustomersController {
    //依照型別注入依賴物件DataSource
    @Autowired
    private SQLServerDataSource datasource;
```

```java
//使用注入依賴的DataSource
@GetMapping(path="/datasource")
public String dataSourceDemo(Model model) {
    try {
        String dbName=datasource.getConnection().getCatalog();
        model.addAttribute("dbName",dbName);
        model.addAttribute("ds",datasource);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return "showdatasource";
}
```



NORTHWND

SQLServerDataSource:1

總結：9-2 Spring Boot配置DataSource

了解如何配置DataSource Configuration以及如何定義一個DataSource Spring Bean之後，接下來我們來看看如何使用Spring Data JPA設計應對資料庫資料表的存取方法。