

5.5. (補充) 兩個 class 實作 Implement

上一章有提到 Spring Boot 可以不要是 Singleton，那我們再建立一個 CarServiceImpl2.java，並把原本的那支 ServiceImpl 的內容全部複製過來，看看 Spring Boot 會發生什麼事。

```
@Service
public class CarServiceImpl2 implements CarService{
    @Autowired
    private CarRepository carRepository;

    @Override
    public List<CarEntity> queryAllCar() {
        return carRepository.findAll();
    }

    @Override
    public CarResponse queryCar(CarRequest carRequest) throws DataNotFoundException {
        String manufacturer = carRequest.getManufacturer();
        String type = carRequest.getType();
        List<CarEntity> list = carRepository.findByManufacturerAndType(manufacturer, type);
        if (list == null || list.isEmpty()) {
            throw new DataNotFoundException();
        }

        CarEntity carEntity = list.get(0);
        CarResponse carResponse = new CarResponse();

        CarResponse.Data responseInnerData = new CarResponse.Data();

        responseInnerData.setManufacturer(carEntity.getManufacturer());
        responseInnerData.setType(carEntity.getType());
        responseInnerData.setPrice(carEntity.getPrice());
        responseInnerData.setMinPrice(carEntity.getMinPrice());
        List<CarResponse.Data> datas = new ArrayList<>();
        datas.add(responseInnerData);

        carResponse.setDatas(datas);
        carResponse.setMessage("success");

        return carResponse;
    }
}
```

這時候應該會發現 Spring Boot 專案啟動失敗，出現如下面這樣的訊息：

```
*****
APPLICATION FAILED TO START
*****
```

Description:

```
Field carService in com.example.answer.controller.CarController required a single bean, but 2 were found:
- carServiceImpl: defined in file [D:\DEVTOOLS\Projects\answer\answer\target\classes\com\example\answer\service\impl\CarServiceImpl.class]
- carServiceImpl2: defined in file [D:\DEVTOOLS\Projects\answer\answer\target\classes\com\example\answer\service\impl\CarServiceImpl2.class]
```

Action:

Consider marking one of the beans as @Primary, updating the consumer to accept multiple beans, or using @Qualifier to identify the bean that should be

編譯器會告訴你說它只需要單一個 Bean，但是在專案中找到了兩個，一個是 CarServiceImpl.java，另一個是剛剛建立的 CarServiceImpl2.java。如果要指定要用哪一個實例，可以用 @Primary、@Qualifier 等 Annotation 來指定使用哪一個 ServiceImpl，解決的方法不只一種，這邊提供比較常看到的三種。

方法一：@Primary

在 ServiceImpl Class 加上 @Primary Annotation，之後 @Autowired 取得的實例物件就會是這個 Class。

```
@Primary
@Component
public class MagicianServiceImpl {
    // do sth here...
}
```

方法二：@Qualifier

先在 ServiceImpl Class 加上 @Qualifier 以及別名。

```
@Service
@Qualifier("magician1")
public class MagicianServiceImpl {
    // do sth here...
}
```

在 Controller 引用的時候加入 @Qualifier 並指定此用哪一個 ServiceImpl。使用這個方法時，變數名稱不需要跟 @Qualifier 中的別名相同。

```
@RestController
public class MagicianController {
    @Autowired
    @Qualifier("magician1")
    private MagicianService magicianService1; // 變數名稱可跟 @Qualifier 別名不同
}
```

方法三：@Qualifier & @Component

在任何一個有 @Component 的 Annotation 上加入別名，在 Controller 取出來的時候，宣告的別名需要跟參數名稱相同。

```
@Service("magician1")
//@Qualifier
public class MagicianServiceImpl {
    // do sth here...
}

@RestController
public class MagicianController {
    @Autowired
    private MagicianService magician1; // 變數名稱需要跟宣告的 @Component 別名相同
}
```

參考

<https://matthung0807.blogspot.com/2019/12/spring-autowired-with-qualifier-inject.html>

<https://www.baeldung.com/spring-qualifier-annotation>