9. 調整 DTO

前面的例子，是將 request dto 和 response dto 調成一樣，然而多數時候其實不同，這邊只是先定義成相同而已。dto 所代表的是一種資料的傳輸格式，也是前面所提到的前後端資料的傳接溝通，想像一下想做的 API，是提供一個查資料的管道，那 Request 的欄位可能是下面這樣：

```
{
    "Manufacturer": "Audi",
    "Type": "A8"
}
```

使用 Manufacturer 和 Type 對資料庫進行查詢，因此順理成章地開了這兩個欄位。那 Reponse 呢？Response 應該提供查詢後的完整資料欄位，也就是下面的形式：

```
{
    "Manufacturer": "Audi",
    "Type": "A8",
    "Min_Price": "100",
    "Price": "1000"
}
```

若資料庫不存在這筆資料，該怎麼處理？也許有人會出像下面的訊息：

```
{
    "Manufacturer": "Audi",
    "Type": "A8",
    "Min_Price": null,
    "Price": null
}
```

然而還有更好的做法，單純的吐出 null 其實不能夠真正代表資料不存在，也許欄位本身就是空的，因此可以多定義一個訊息，讓回傳的資訊變得更加明顯：

```
{
    "Message": "success",
    "Datas": [
        {
            "Manufacturer": "Audi",
            "Type": "A8",
            "Min_Price": null,
            "Price": null
        }
    ]
}

{
    "Message": "data not found",
    "Datas": null
}
```

多出一個 message 欄位，去代表 API 的執行結果，會讓訊息變得更加明瞭。這邊是個簡單的範例，重點在於 dto 可以相當多樣，也可定義多層 JSON 資料格式，而 Request 和 Response 也可以不同，它們所代表的只是資料的傳輸格式。再來實作程式碼的部分：

```
|--com.example.demospringboot
   |--DemospringbootApplication.java
|--com.example.demospringboot.configuration
   |--SwaggerConfig.java
   |--RestConfiguration.java
|--com.example.demospringboot.controller
   |--CarController.java
   |--TestController.java
   |--ProductController.java
|--com.example.demospringboot.dto
   |--CarRequest.java
   |--CarResponse.java
   |--CarResponse2.java // 新增的檔案
   |--Data.java // 新增的檔案
|--com.example.demospringboot.entity
   |--CarEntity.java
   |--CarPKEntity.java
|--com.example.demospringboot.model
   |--Product.java
|--com.example.demospringboot.repository
   |--CarRepository.java
|--com.example.demospringboot.service
   |--CarService.java
   |--ProductService.java
|--com.example.demospringboot.service.impl
   |--CarServiceImpl.java
   |--ProductServiceImpl.java
```

```
public class CarResponse2 {
    @JsonProperty("Message")
    private String message;

    @JsonProperty("Datas")
    private List<Data> datas;

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public List<Data> getDatas() {
        return datas;
    }

    public void setDatas(List<Data> datas) {
        this.datas = datas;
    }
}

public class Data {
    @JsonProperty("Manufacturer")
    private String manufacturer;

    @JsonProperty("Type")
    private String type;

    @JsonProperty("Min_Price")
    private BigDecimal minPrice;

    @JsonProperty("Price")
    private BigDecimal price;

    public String getManufacturer() {
        return manufacturer;
    }

    public void setManufacturer(String manufacturer) {
        this.manufacturer = manufacturer;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
```

```
        this.type = type;
    }

    public BigDecimal getMinPrice() {
        return minPrice;
    }

    public void setMinPrice(BigDecimal minPrice) {
        this.minPrice = minPrice;
    }

    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }
}
```

另一個更動的檔案是 `CarServiceImpl.java` ：

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

    public CarResponse2 queryCar2(CarRequest carRequest) {
        String manufacturer = carRequest.getManufacturer();
        String type = carRequest.getType();
        List<Car> list = carRepository.findByManufacturerAndType(manufacturer, type);
        Car carEntity = list.get(0);

        Data responseInnerData = new Data();
        responseInnerData.setManufacturer(carEntity.getManufacturer());
        responseInnerData.setType(carEntity.getType());
        responseInnerData.setPrice(carEntity.getPrice());
        responseInnerData.setMinPrice(carEntity.getMinPrice());

        List<Data> datas = new ArrayList<>();
        datas.add(responseInnerData);

        CarResponse2 carResponse = new CarResponse2();
        carResponse.setDatas(datas);
        carResponse.setMessage("success");

        return carResponse;
    }
}
```

將資料查出來後，再將資料放到 response 物件裡，進行回傳。最後再把 `CarController.java` 以及 `CarService.java` 的回傳型別改正即可。