

10-2 JSON文件規格



JSON(JavaScript Object Notation)

由道格拉斯·克羅克福特構想和設計的一種輕量級資料交換格式(相較於XML)

JSON可以進行巢狀資料格式設定，如同XML一般，但較為輕量。

為JavaScript Subset，所以在配合前端AJAX呼喚回呼時，整合性高。

```
[{
  "id": "0001",
  "name": "張泰山",
  "address": {
    "country": "中華民國",
    "city": "台北市",
    "road": "仁愛路",
    "number": "1"
  },
  "phone": [
    {
      "type": "homephone",
      "number": "02-12345678"
    },
    {
      "type": "mobile",
      "number": "0911-123456"
    }
  ]
}]
```

```
▼ [ 1 item
  ▼ 0 : {
    id : 0001
    name : 張泰山
    ► address : { 4 props }
    ► phone : [ 2 items ]
  }
]
```

規劃一個Company具有多個Employees
物件依賴架構

透過List集合物件進行參考

```
import java.util.ArrayList;
import java.util.List;

//公司行號類別JavaBean
public class Company implements java.io.Serializable {
    private String companyName;
    private String address;
    private String phone;
    private List<Employees> employees=new ArrayList<>();
```

```
//Employees JavaBean
public class Employees implements java.io.Serializable{
    public Employees() {
        this.id=(int) (Math.random()*10000)+1+"";
    }
    private String id;
    private String name;
```

設計RESTful Controller

提供一個Action進行公司物件與員工資料獲取

```
//公司行號類別JavaBean
public class Company implements java.io.Serializable {
    private String companyName;
    private String address;
    private String phone;
    private List<Employees> employees=new ArrayList<>();
```

關聯性類別設定

```
//關聯資料服務
public static Company companyData()
{
    //建構公司物件
    Company com=new Company();
    com.setCompanyName("凱旋企業");
    com.setAddress("台北市和平東路");
    com.setPhone("02-12345678");
    //建構員工物件
    Employees emp1=new Employees();
    emp1.setName("張無忌");
    Employees emp2=new Employees();
    emp2.setName("張三封");
    com.getEmployees().add(emp1);
    com.getEmployees().add(emp2);
    return com;
}
```

```
package com.tibame.entity;
//Employees JavaBean
public class Employees implements java.io.Serializable{
    public Employees() {
        this.id=(int) (Math.random()*10000)+1+"";
    }
    private String id;
    private String name;
```

```
//公司與員工互動關係Service
@RestController
public class CompanyService {

    //公司與多個員工物件查詢
    @GetMapping(path="/api/company", produces="application/json")
    @ResponseBody
    public Company getCompany() {
        return companyData();
    }
}
```

決定回應序列化MIME Type

http://localhost:8080/api/company

GET http://localhost:8080/api/company

Params Authorization Headers (6) Body Pre-request S

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "companyName": "凱旋企業",
3   "address": "台北市和平東路",
4   "phone": "02-12345678",
5   "employees": [
6     {
7       "id": "8058",
8       "name": "張無忌"
9     },
10    {
11      "id": "2227",
12      "name": "張三封"
13    }
14  ]
15 }
```



總結：10-2 JSON文件規格

了解JSON風格之後，與JavaBean序列化JSON關聯機制，接下來我們實作一個整合JPA資料存取的RESTful Service。

