

IBM Internship Report

Chen Jiali

3035085695

1. Organization And Department

1.1 Organization Introduction

International Business Machines Corporation (commonly referred as IBM), is a multinational technology company headquartered. The company originated in 1911 as the Computing-Tabulating-Recording Company (CTR) and was renamed IBM in 1924.

IBM markets computer hardware, middleware and software, and offers hosting and consulting services in areas ranging from mainframe computers to nanotechnology. IBM also emphasize on research organization, holding the record for most patents generated by a business(as of 2016) for 23 consecutive years.

In recent years, IBM, nicknamed Big Blue is experiencing an innovation, relocating its field from computer hardware, middleware and software to cognitive computing and cloud service. For the previous, Watson is provided while for the latter, IBM produced Blue-mix.

1.2 Department Introduction

The department EBU is mainly for IBM cloud service sales and afterwards implementation during which different sorts of engineers and project managers are gathered and work for the project.

2. Job Responsibility

My detailed job is to stay at the customer site, i.e. PATAC Shanghai, helping them develop the web application project. Within the project there are three parts, the server side for the whole, the data managing system and the client side flow control web.

I got participated in the development of the server side and the data managing system. My job are to follow the instructions and the assignments from my boss, an java architecture, to code some interfaces for the client side service as well as to help one of my colleague, a front-end developer, developing some web pages for the data managing system.

3. Detailed Job Content And Product

As mentioned before, there are two parts having my participation, the server side and the data manage system.

3.1 Server Side Work

For the server side, Java spring framework and Mybatis are used. In order to handle the demands from the customers, complex or trivial, interfaces are created in the controllers for the front-end to call. Since the customers are really demanding and their needs are daily changeable, there are plenty of interfaces to code every day. For your instance and reference, one interface is showed there.

When an interface is called, there are logically three steps.

I. Controller

```
@Controller
@RequestMapping("/entrustInfo")
public class EntrustInfoController {
```

Controller is the most external level of the server side in the Spring framework. Its function, revealed by its name, receive the requests from the client side, control and map them to the right interface.

```
// written by Chen Jia Li
@RequestMapping(value="/{format}/entrustSampleCheck/{entrustNo}")
public ModelAndView entrustSampleCheck(@PathVariable String format,
    @PathVariable String entrustNo){
    ModelAndView mv = new ModelAndView(format);
    Map<String, Object> map = entrustInfoService.entrustSampleCheck(entrustNo);
    mv.addObject(map);
    return mv;
}
```

For your easy reference, an interface, with relatively simple logic is chosen, which handles the requests to entrustSampleCheck. From the url, the server get path variable format (json or xml) and entrust number, and pass the necessary ones to the next level -- service. There are other annotations like @RequestBody for POST and @RequestParam or encapsulated data structure for GET.

II. Service

Now comes the service, second level of spring's logic. Service consists of interfaces and their implementations. Functions are declared in the interface while determinedly defined in the implementation. The declare part is trivial, thus only the implementation is showed there.

```
@Override
public Map<String, Object> entrustSampleCheck(String entrustNo) {
    // TODO Auto-generated method stub
    String res;
    List<TEntrustSample> list = tEntrustSampleMapper.selectByEntrustNo(entrustNo);
    Map<String, Object> map = new HashMap<String, Object>();
    if (null == list || 0 == list.size()){
        res = "error";
    }else{
        res = "normal";
    }
    map.put("status", res);
    return map;
}
```

Within the service, logic is explicitly defined and mapper are used for the communication with the Database.

III Mapper

In the third level, actual data selecting, adding or updating etc take place. Java is only responsible for the interfaces.

```
public interface TEntrustSampleMapper {  
    int deleteByPrimaryKey(Long sampleId);  
  
    int insert(TEntrustSample record);  
  
    int insertSelective(TEntrustSample record);  
  
    TEntrustSample selectByPrimaryKey(Long sampleId);  
}
```

While Mybatis should take charge of the actual communication. Also, “selectByEntrustNo” used above, with relatively simple logic, is provided.

```
<select id="selectByEntrustNo" resultMap="BaseResultMap" parameterType="java.lang.String" >  
    select  
    <include refid="Base_Column_List" />  
    from T_ENTRUST_SAMPLE  
    where ENTRUST_NO = #{entrustNo,jdbcType=VARCHAR}  
</select>
```

SQL clauses are encapsulated in the xml files, with resultMap defining output data structure and parameterType for the input one.

3.2 Data Managing System

The data managing system is a front-end system, providing visible interfaces for the user to add, update, delete data. Compared to the server logic, the data manage system, as a front-end one, is logically simpler. What our team used for developing it is Angular instead of jQuery.

The major work i did is adding the search function on several pages. Also for your reference. One of the searching elements is picked.

```
<label data-translate="TEST_AREA_ID"></label>  
<span>&nbsp;&nbsp;&nbsp;&nbsp;</span>  
<select ng-model="searchTestAreaId"  
    ng-options="x.testAreaId as x.testAreaName for x in searchTestareas"  
    data-ng-change="searchWorktype()"  
    class="select"  
    style="width: 120px"  
>
```

In Angular framework, elements are now labeled by “ng-model”, which acts as an id for the controller, the javascript file, to control.

```
<div ng-controller="equipClassCtrl">
```

While in the controller, search functions are defined to control the search actions.


```
//载入
var promise = httpService.getData('equipment/json/queryByIdAndClaName?pageNum=1&pageSize=10'+$scope.timea+$scope.timeb);
promise.then(function (data) {
    $scope.equipclass = data.list;
    $scope.paginationConf.totalItems = data.total;
    $scope.paginationConf.pagesLength = data.pages;
});
//分页配置
$scope.paginationConf = {
    currentPage: 1,
    itemsPerPage: 10,
    perPageOptions: [10, 15, 20, 30, 50],
    rememberPerPage: 'perPageItems',
    onChange: function () {
        var promise = httpService.getData('equipment/json/queryByIdAndClaName?pageNum=' + $scope.paginationConf.currentPage
        promise.then(function (data) {
            $scope.equipclass = data.list;
            $scope.paginationConf.totalItems = data.total;
            $scope.paginationConf.pagesLength = data.pages;
        });
    }
};
```

Requests are sent to the right server interface, get json and automatically parse it, fetch the data we need from the data structure, and change the website showing by calling \$scope, the model controlled by this controller. With assistance of Angular JS, data can be easily shown by the iteration label “ng-repeat”.

```
<tr ng-repeat="x in equip">
    <td>{{x.equipId}}</td>
    <td>{{x.equipName}}</td>
    <td>{{x.equipType}}</td>
    <td>{{x.equipQty}}</td>
    <td>{{x.equipOwner}}</td>
    <td><a href="javascript:;" data-translate="DELETE" ng-click="delete_sub($index)"></a>
    </td>
</tr>
```

3.3 Product

For most of the time, my duty ends up to sending json/xml data to the client side.



← → ↻

This XML file does not appear to have any style information associated with it. The document

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<com.github.pagehelper.PageInfo>
  <pageNum>1</pageNum>
  <pageSize>10</pageSize>
  <size>10</size>
  <startRow>1</startRow>
  <endRow>10</endRow>
  <total>12</total>
  <pages>2</pages>
</com.github.pagehelper.PageInfo>
<list class="com.github.pagehelper.Page" serialization="custom">
  <unserializable-parents/>
  <list>
    <default>
      <size>10</size>
    </default>
    <int>10</int>
    <map>
      <entry>
        <string>createTime</string>
        <null/>
      </entry>
      <entry>
        <string>createUser</string>
        <null/>
      </entry>
      <entry>
        <string>updateTime</string>
        <null/>
      </entry>
    </map>
  </list>
</list>
```

While some times, the server interface and the front-end pages should be developed as a whole.

设备组管理

添加设备组

场地:

任务单类型:

设备类型名称:

搜索

设备类型名称	试验室名称	操作
设备组测试组	Calibration & Consignation Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
sswqw	Vehicle Road Performance	<div>删除</div> <div>修改</div> <div>查看设备</div>
设备组测试组	Chassis Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
sdds	Safety Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
ewqeqwewe	Safety Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
ewqeqwewe	Calibration & Consignation Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
设备组2	Elec. Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
设备组测试组3	Chassis Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
dwww	Dyno (Vehicle Performance) Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>
asd	Emission Lab	<div>删除</div> <div>修改</div> <div>查看设备</div>

« 1 2 »

第 1 页 每页 10 /共12条

设备组管理

添加设备组

场地:

Lab

任务单类型:

Calibration & Consignation

设备类型名称:

搜索

« 1 »

第 1 页 每页 10 /共2条

4. Assessment of the Technical substances

4.1 Spring + Mybatis

First we discuss the Spring framework. The Spring Framework is an application framework and inversion of control container for the Java platform. Inversion of control (IoC), main feature of this framework, decouple the server and the client. It's the framework, instead of programmer who controls the calls of the interfaces, which greatly simplifies the logic.

Now comes the Mybatis, different from ORM frameworks, Mybatis will not map Java Object to DB table, instead, Java Interfaces are mapped to SQL statements. Compared to JDBC, coding communicating with DB is simplified. Enterprise web software development can thus be easier.

4.2 Angular JS

AngularJS is a complete JavaScript-based open-source front-end web application framework maintained by Google, a community and some corporations. The main

feature, or so-called design goal is: to decouple DOM manipulation from application logic; to decouple the client side of an application from the server side; to provide structure for the journey of building an application. With the help of Angular, models and views are automatically synchronized, which also shorten the develop time.

5. Assessment of Value of the Internship Experience

Getting the internship at IBM is such a great opportunity to learn things, no matter soft skills or hard skills. Besides the technical activities introduced above, there are workshops and group tasks, which effectively improve my communication and organization skills. For technical field, I've got a deep understanding of the Spring framework used commonly in server developing and a brief usage of the Angular JS framework.

Conclusively, these skills are definitely useful in my future and the internship experience is a great prelude of my CS career.