

# PYG\_Day02

## 第1节课

前日内容回顾

Dubbox

注册中心

生产者(服务提供者)

消费者(服务调用者)

项目分层架构搭建(理解)

### 1.1 本节知识点

1. .今日目标
2. .angularJS简介 (了解)
3. .angularJS四大特征 (学完当天课程后理解)
4. .angularJS入门小demo(1-3) (重点)
5. .angularJS入门小demo4 (重点)
6. .angularJS入门小demo(5-6) (重点)
7. .angularJS入门小demo(7) (重点)
8. .angularJS入门小demo(8) (重点)
9. .品牌列表-需求分析 (下面的视频全是练习,需要完成)
10. .品牌列表-1
11. .品牌列表-2
12. .品牌分页-需求分析
13. .品牌分页-后端-1
14. .品牌分页-后端-2
15. .品牌分页-前端-1
16. .品牌分页-前端-2
17. .品牌增加-后端
18. .品牌增加-前端
19. .品牌修改-后端
20. .品牌修改-前端-1
21. .品牌修改-前端-2
22. .品牌删除-后端
23. .品牌删除-前端-1
24. .品牌删除-前端-2
25. .品牌条件查询-后端
26. .品牌条件查询-前端

### 1.2 本节目标

- 目标 1: 运用 AngularJS 前端框架的常用指令（重点,掌握）
- 目标 2: 完成品牌管理的列表功能（练习要求完成）
- 目标 3: 完成品牌管理的分页列表功能（练习要求完成）
- 目标 4: 完成品牌管理的增加功能（练习要求完成）
- 目标 5: 完成品牌管理的修改功能（练习要求完成）
- 目标 6: 完成品牌管理的删除功能（练习要求完成）
- 目标 7: 完成品牌管理的条件查询功能（练习要求完成）

## 1.3课程内容

### 1.3.1 .今日目标

#### 视频信息

视频名称: 01. 今日目标  
视频时长: 01:39

#### 小节内容

- 目标 1: 运用 AngularJS 前端框架的常用指令
- 目标 2: 完成品牌管理的列表功能
- 目标 3: 完成品牌管理的分页列表功能
- 目标 4: 完成品牌管理的增加功能
- 目标 5: 完成品牌管理的修改功能
- 目标 6: 完成品牌管理的删除功能
- 目标 7: 完成品牌管理的条件查询功能

### 1.3.2 .angularJS简介

#### 视频信息

视频名称: 02.angularJS简介  
视频时长: 04:04

#### 小节内容

AngularJS 诞生于 2009 年, 由 Misko Hevery 等人创建, 后为 Google 所收购。是一款优秀的前端 JS 框架, 已经被用于 Google 的多款产品当中。  
MVC、模块化、自动化双向数据绑定、依赖注入

### 1.3.3 .angularJS四大特征

#### 视频信息

视频名称: 03.angularJS四大特征  
视频时长: 11:34

#### 小节内容

- 1)双向绑定  
页面上的数据发生变化, js 中的变量也随之改变,  
js 中的变量发生改变,页面上的数据也变化
- 2)模块化设计  
把页面的功能按照模块划分,不同的模块处理不同的事情
- 3)MVC 模式  
Model:数据,其实就是 angular 变量(\$scope.xx);  
View: 数据的呈现,Html+Directive(指令);  
Controller:操作数据,就是 function,数据的增删改查;
- 4)依赖注入  
和Spring 中依赖注入的思想类似

## 补充

高内聚低耦合  
高内聚:一个好的内聚模块应当恰好做一件事  
低耦合: 模块与模块之间应该尽量减少耦合

## 1.3.4 .angularJS入门小demo(1-3)

### 视频信息

视频名称: 04.angularJS入门小demo(1-3)  
视频时长: 10:55

### 小节内容

- 1)是{{表达式 }} 表达式可以是变量或是运算式
- 2)ng-app 指令 作用是告诉子元素一下的指令是归 angularJs 的,angularJs 会识别的
- 3)ng-model 指令用于绑定变量
- 4)ng-init 指令来对变量初始化

-----  
双向绑定

```
<body ng-app>  
  变量:<input ng-model="name">{{name}}  
</body>
```

初始化

```
<body ng-app ng-init="name='陈大海'">
```

## 1.3.5 .angularJS入门小demo4

### 视频信息

视频名称: 05.angularJS入门小demo4  
视频时长: 14:28

### 小节内容

- 1) 定义了一个叫 myApp 的模块 ,[] 表示其他模块的引用  
`var app=angular.module('myApp',[]);`
- 2) 定义控制器  
`app.controller('myController',function($scope){  
 $scope.add=function(){  
 return parseInt($scope.x)+parseInt($scope.y);  
 }  
});`
- 3)ng-controller :用于指定所使用的控制器

理解 \$scope:

\$scope:

\$scope 的使用贯穿整个 AngularJS App 应用,它与数据模型相关联,同时也是表达式执行的上 下文.有了 \$scope 就在视图和控制器之间建立了一个通道,基于作用域视图在修改数据时会 立刻更新 \$scope,同样的\$scope 发生改变时也会立刻重新渲染视图.

\$scope 是控制层和视图层交换数据的桥梁 : mvc 中的model

```
<head>
<title>入门小 Demo-3 初始化</title>
<script src="angular.min.js"></script>
<script>
var app=angular.module('myApp',[]); //定义了一个叫 myApp 的模块
//定义控制器
app.controller('myController',function($scope){
    $scope.add=function(){
        return parseInt($scope.x)+parseInt($scope.y);
    }
});
</script>
</head>
<body ng-app="myApp" ng-controller="myController">
x:<input ng-model="x" >
y:<input ng-model="y" >
运算结果: {{add()}}
</body>
```

注意:

默认一个页面只能有一个模块,定义多个模块不起作用多个模块

## 1.3.6 .angularJS入门小demo(5-6)

视频信息

视频名称: 06.angularJS入门小demo(5-6)  
视频时长: 07:10

事件指令

5) `ng-click="add()"`

数组循环

6) `ng-repeat="x in list"`

注意: 那个标签上使用 `ng-repeat` 则那个标签循环

例如下面的例子

最终用页面显示的是4个<tr> 标签

案例

```
<html>
<head>
<title>入门小 Demo-4</title>
<script src="angular.min.js"></script>
<script>
var app=angular.module('myApp',[]); //定义了一个叫 myApp 的模块
//定义控制器
app.controller('myController',function($scope){
    $scope.list= [100,192,203,434 ]; //定义数组
});

</script>
</head>
<body ng-app="myApp" ng-controller="myController">
    <table>
        <tr ng-repeat="x in list">
            <td>{{x}}</td>
        </tr>
    </table>
</body>
</html>
```

## 1.3.7 .angularJS入门小demo(7)

视频信息

视频名称: 07.angularJS入门小demo(7)  
视频时长: 04:55

小节内容

7) 循环对象数组

```
<html>
<head>
```

```

<title>入门小 Demo-4</title>
<script src="angular.min.js"></script>
<script>
var app=angular.module('myApp',[]); //定义了一个叫 myApp 的模块
//定义控制器
app.controller('myController',function($scope){
    $scope.list= [
        {name:'张三',shuxue:100,yuwen:93},
        {name:'李四',shuxue:88,yuwen:87},
        {name:'王五',shuxue:77,yuwen:56}
    ]; //定义数组
});

</script>
</head>
<body ng-app="myApp" ng-controller="myController">
<table>
    <tr>
        <td>姓名</td>
        <td>数学</td>
        <td>语文</td>
    </tr>
    <tr ng-repeat="entity in list">
        <td>{{entity.name}}</td>
        <td>{{entity.shuxue}}</td>
        <td>{{entity.yuwen}}</td>
    </tr>
</table>
</body>
</html>

```

### 1.3.8 .angularJS入门小demo(8)

#### 视频信息

视频名称：08.angularJS入门小demo(8)  
视频时长：11:43

#### 小节内容

```

内置服务
$http.get('data.json').success(
    function(response){
        $scope.list=response;
    }
);

```

```

<html>
<head>
<title>入门小 Demo-8 内置服务</title>
<meta charset="utf-8" />

```

```

<script src="angular.min.js"></script>

<script>
var app=angular.module('myApp',[]); //定义了一个叫 myApp 的模块
//定义控制器
app.controller('myController',function($scope,$http){
    $scope.findAll=function(){
        $http.get('data.json').success(
            function(response){
                $scope.list=response;
            }
        );
    }
});
</script>
</head>
<body ng-app="myApp" ng-controller="myController" ng-init="findAll()">
<table>
<tr>
<td>姓名</td>
<td>数学</td>
<td>语文</td>
</tr>
<tr ng-repeat="entity in list">
<td>{{entity.name}}</td>
<td>{{entity.shuxue}}</td>
<td>{{entity.yuwen}}</td>
</tr>
</table>
</body>
</html>

```

## 补充

## 1.3.9 .品牌列表-需求分析

### 视频信息

视频名称：09.品牌列表-需求分析  
视频时长：02:28

### 小节内容

1)要实现的效果是把第一天的json 数据,在界面上用AdminLTE+angularJS 实现

## 1.3.10 .品牌列表-1

### 视频信息

视频名称：10.品牌列表-1  
视频时长：10:01

### 小节内容

```
1) copy 金泰资源至webapp 下
2) brand.html
var app=angular.module('pinyougou',['pagination']);
app.controller('brandController',function($scope,$http){
    //查询品牌列表
    $scope.findAll=function(){
        $http.get('../brand/findAll.do').success(
            function(response){
                $scope.list=response;
            }
        );
    }
});
<body ng-app="pinyougou" ng-controller="brandController" ng-init="findAll()">
```

### 补充

## 1.3.11 .品牌列表-2

### 视频信息

视频名称：11.品牌列表-2  
视频时长：01:52

### 小节内容

```
<tr ng-repeat="entity in list">
    <td><input type="checkbox" ></td>
    <td>{{entity.id}}</td>
    <td>{{entity.name}}</td>
    <td>{{entity.firstChar}}</td>
</tr>
```

### 补充

## 1.3.12 .品牌分页-需求分析

### 视频信息



视频名称：12.品牌分页-需求分析  
视频时长：01:19

### 小节内容

数据太多的化,全部都显示显然不合理,需分页显示

## 1.3.13 .品牌分页-后端-1

### 视频信息

视频名称：13.品牌分页-后端-1  
视频时长：08:24

### 小节内容

1) 返回的数据:

总记录数  
当前页数据

2) 请求数据

页码  
每页记录数

我们对响应结果定义一个对象返回结果

```
public class PageResult implements Serializable{  
    private long total;//总记录数  
    private List rows;//当前页记录
```

## 1.3.14 .品牌分页-后端-2

### 视频信息

视频名称：14.品牌分页-后端-2  
视频时长：08:59

### 小节内容ServiceImpl

```
@Override  
public PageResult findPage(int pageNum, int pageSize) {  
    PageHelper.startPage(pageNum, pageSize);//分页  
    Page<TbBrand> page = (Page<TbBrand>) brandMapper.selectByExample(null);  
    return new PageResult(page.getTotal(), page.getResult());  
}
```

### Controller

```
@RequestMapping("/findPage")
public PageResult findPage(int page,int size){
    return brandService.findPage(page, size);
}
```

## 1.3.15 .品牌分页-前端-1

### 视频信息

视频名称: 15.品牌分页-前端-1  
视频时长: 05:54

### 小节内容

需要分页插件的支持

```
<script src="../../plugins/angularjs/pagination.js"></script>
<link rel="stylesheet" href="../../plugins/angularjs/pagination.css">
<tm-pagination conf="paginationConf"></tm-pagination>
```

备注:

paginationConf: 需要在变量中指定

```
$scope.paginationConf = {
    currentPage: 1, // 当前页
    totalItems: 10, // 总记录数
    itemsPerPage: 10, // 每页记录数
    perPageOptions: [10, 20, 30, 40, 50], // 分页选项
    onChange: function(){ // 当页码或者每页数
        $scope.reloadList();//重新加载
    }
};
```

---在需要怎加分页的地方加入如下配置即可完成分页-----

```
<tm-pagination conf="paginationConf"></tm-pagination>
```

## 1.3.16 .品牌分页-前端-2

### 视频信息

视频名称: 16.品牌分页-前端-2  
视频时长: 07:54

### 小节内容

分页控件配置currentPage:当前页 totalItems :总记录数 itemsPerPage:每页记录数  
perPageOptions :分页选项 onChange:当页码变更后自动触发的方法

```
$scope.paginationConf = {
    currentPage: 1,
    totalItems: 10,
    itemsPerPage: 10,
```

```

        perPageOptions: [10, 20, 30, 40, 50],
        onChange: function(){
            $scope.reloadList();
        }
    };

    //刷新列表
    $scope.reloadList=function(){
        $scope.findPage(
            $scope.paginationConf.currentPage,$scope.paginationConf.itemsPerPage);
    }
    //分页
    $scope.findPage=function(page,size){
        $http.get('../brand/findPage.do?page='+page+'&size='+size).success(
            function(response){
                $scope.list=response.rows;//显示当前页数据
                $scope.paginationConf.totalItems=response.total;//更新总记录数
            }
        );
    }
}

```

## 补充

页面加载时,分页插件会按照初始化参数自动请求一次数据库

## 1.3.17 .品牌增加-后端

### 视频信息

视频名称: 17. 品牌增加-后端

视频时长: 09:19

### 小节内容

```

@RequestMapping("/add")
public Result add(@RequestBody TbBrand brand){
    try {
        brandService.add(brand);
        return new Result(true, "增加成功");
    } catch (Exception e) {
        e.printStackTrace();
        return new Result(false, "增加失败");
    }
}
}

```

返回结果用一个对象分装,方便返回数据

## 补充

@RequestBody

- 1) 请求方式只能是post
- 2) 请求数据格式必须是 json 格式

## 1.3.18 .品牌增加-前端

### 视频信息

视频名称: 18.品牌增加-前端  
视频时长: 09:53

### 小节内容

```
//新增
$scope.add=function(){
    var methodName='add';//方法名
    $http.post('../brand/add.do',$scope.entity).success(
        function(response){
            if(response.success){
                $scope.reloadList();//刷新
            }else{
                alert(response.message);
            }
        }
    );
}
```

```
<table class="table table-bordered table-striped" width="800px">
    <tr>
        <td>品牌名称</td>
        <td><input class="form-control" placeholder="品牌名称" ng-model="entity.name">
    </td>
    </tr>
    <tr>
        <td>首字母</td>
        <td><input class="form-control" placeholder="首字母" ng-
model="entity.firstChar"> </td>
    </tr>
</table>
```

### 补充

## 1.3.19 .品牌修改-后端

## 视频信息

视频名称：19.品牌修改-后端  
视频时长：05:30

## 小节内容

```
@Override
public TbBrand findOne(Long id) {
    return brandMapper.selectByPrimaryKey(id);
}
@Override
public void update(TbBrand brand) {
    brandMapper.updateByPrimaryKey(brand);
}
@RequestMapping("/update")
public Result update(@RequestBody TbBrand brand){
    try {
        brandService.update(brand);
        return new Result(true, "修改成功");
    } catch (Exception e) {
        e.printStackTrace();
        return new Result(false, "修改失败");
    }
}
```

## 1.3.20 .品牌修改-前端-1

## 视频信息

视频名称：20.品牌修改-前端-1  
视频时长：05:09

## 小节内容

```
//查询实体
$scope.findOne=function(id){
    $http.get('../brand/findOne.do?id='+id).success(
        function(response){
            $scope.entity=response;
        }
    );
}
```

## html

```
<td >
    <button type="button" ng-click="findOne(entity.id)" >修改</button>
</td>
```

## 1.3.21 .品牌修改-前端-2

### 视频信息

视频名称： 21.品牌修改-前端-2  
视频时长： 03:56

### 小节内容

```
//新增
$scope.save=function(){
    var methodName='add';//方法名
    if($scope.entity.id!=null){
        methodName='update';
    }
    $http.post('../brand/'+methodName+'.do',$scope.entity).success(
        function(response){
            if(response.success){
                $scope.reloadList();//刷新
            }else{
                alert(response.message);
            }
        }
    );
}
<button class="btn btn-success" data-dismiss="modal" aria-hidden="true" ngclick="
save()">保存</button>
```

### 补充

思路：  
根据是否有ID 判断是新增还是更新

## 1.3.22 .品牌删除-后端

### 视频信息

视频名称： 22.品牌删除-后端  
视频时长： 03:43

### 小节内容

```
@Override
public void delete(Long[] ids) {
    for(Long id:ids){
        brandMapper.deleteByPrimaryKey(id);
    }
}
@RequestMapping("/delete")
public Result delete(Long [] ids){
```

```

    try {
        brandService.delete(ids);
        return new Result(true, "删除成功");
    } catch (Exception e) {
        e.printStackTrace();
        return new Result(false, "删除失败");
    }
}

```

## 补充

## 1.3.23 .品牌删除-前端-1

### 视频信息

视频名称：23.品牌删除-前端-1  
视频时长：10:44

### 小节内容

```

$event:
    源数据
$event.target = 原来的input 标签

//用户勾选的ID集合
$scope.selectIds=[];

//用户勾选复选框
$scope.updateSelection=function($event,id){
    if($event.target.checked){
        $scope.selectIds.push(id);//push向集合添加元素
    }else{
        var index= $scope.selectIds.indexOf(id);//查找值的 位置
        $scope.selectIds.splice(index,1);//参数1: 移除的位置 参数2: 移除的个数
    }
}
<input type="checkbox" ng-click="updateSelection($event, entity.id)" >

```

### 补充:

indexOf 和 splice 都是js 提供的方法 详见w3cschool  
indexOf()  
splice() 方法向/从数组中添加/删除项目，然后返回被删除的项目。

## 1.3.24 .品牌删除-前端-2

## 视频信息

视频名称: 24.品牌删除-前端-2  
视频时长: 03:47

## 小节内容

```
$scope.delete=function(){
    if(confirm('确定要删除吗? ')){
        $http.get('../brand/delete.do?ids='+$scope.selectIds).success(
            function(response){
                if(response.success){
                    $scope.reloadList();//刷新
                }else{
                    alert(response.message);
                }
            }
        );
    }
}
```

## 补充

## 1.3.25 .品牌条件查询-后端

## 视频信息

视频名称: 25.品牌条件查询-后端  
视频时长: 08:26

## 小节内容

```
PageHelper.startPage(pageNum, pageSize);//分页
// 构造分页条件
TbBrandExample example=new TbBrandExample();

Criteria criteria = example.createCriteria();
if(brand!=null){
    if(brand.getName()!=null && brand.getName().length()>0){
        criteria.andNameLike("%"+brand.getName()+"%");
    }
    if(brand.getFirstChar()!=null && brand.getFirstChar().length()>0){
        criteria.andFirstCharLike("%"+brand.getFirstChar()+"%");
    }
}

Page<TbBrand> page = (Page<TbBrand>) brandMapper.selectByExample(example);

return new PageResult(page.getTotal(), page.getResult());
```



## Controller

```
@RequestMapping("/search")
public PageResult search(@RequestBody TbBrand brand,int page,int size){
    return brandService.findPage(brand, page, size);
}
```

### 1.3.26 .品牌条件查询-前端

#### 视频信息

视频名称： 26.品牌条件查询-前端  
视频时长： 10:53

#### 小节内容

```
$scope.searchEntity={};
//条件查询
$scope.search=function(page,size){

    $http.post(' ../brand/search.do?page='+page +'&size='+size,
    $scope.searchEntity).success(
        function(response){
            $scope.list=response.rows;//显示当前页数据
            $scope.paginationConf.totalItems=response.total;//更新总记录数
        }
    );
}
```

```
$scope
$http
$evnet.target
```