

优乐选实现CRUD

web.xml

修改youlexuanmanagerweb\src\main\webapp\WEB-INF\web.xml

```
<servlet-mapping>
    <servlet-name>springmvc</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

分页查询

DTO

youlexuan-pojo模块中增加一个通用分页数据传输对象。

```
package com.xxx.proj.dto;

import java.io.Serializable;
import java.util.List;

public class PageResult implements Serializable {
    private long total;
    private List rows;

    public PageResult() {
    }

    public PageResult(long total, List rows) {
        this.total = total;
        this.rows = rows;
    }

    public long getTotal() {
        return total;
    }

    public void setTotal(long total) {
        this.total = total;
    }

    public List getRows() {
        return rows;
    }

    public void setRows(List rows) {
        this.rows = rows;
    }
}
```

```
}
```

持久层

youlexuan-dao无需做任何修改

服务层

youlexuan-sellergoods-interface增加接口

```
package com.xxx.proj.service;

import com.xxx.proj.dto.PageResult;
import com.xxx.proj.pojo.TbBrand;

import java.util.List;

public interface TbBrandService {
    List<TbBrand> findAll();

    PageResult findPage(int pageNum, int pageSize);
}
```

youlexuan-sellergoods-service增加实现类

```
package com.xxx.proj.service;

import com.alibaba.dubbo.config.annotation.Service;
import com.github.pagehelper.Page;
import com.github.pagehelper.PageHelper;
import com.xxx.proj.dto.PageResult;
import com.xxx.proj.mapper.TbBrandMapper;
import com.xxx.proj.pojo.TbBrand;
import org.springframework.beans.factory.annotation.Autowired;

import java.util.List;

@Service
public class TbBrandServiceImpl implements TbBrandService {
    @Autowired
    private TbBrandMapper mapper;

    @Override
    public List<TbBrand> findAll() {
        return mapper.selectByExample(null);
    }

    @Override
    public PageResult findPage(int pageNum, int pageSize) {
        //这里使用PageHelper进行分页
        PageHelper.startPage(pageNum, pageSize);
        //TODO:注意: 这里的Page类-import com.github.pagehelper.Page;
    }
}
```

```

        Page<TbBrand> page = (Page<TbBrand>) mapper.selectByExample(null);
        return new PageResult(page.getTotal(), page.getResult());
    }
}

```

控制层

youlexuan-manager-web增加控制器

```

package com.xxx.proj.controller;

import com.alibaba.dubbo.config.annotation.Reference;
import com.xxx.proj.dto.PageResult;
import com.xxx.proj.pojo.TbBrand;
import com.xxx.proj.service.TbBrandService;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/brand")
public class TbBrandController {
    @Reference
    private TbBrandService service;

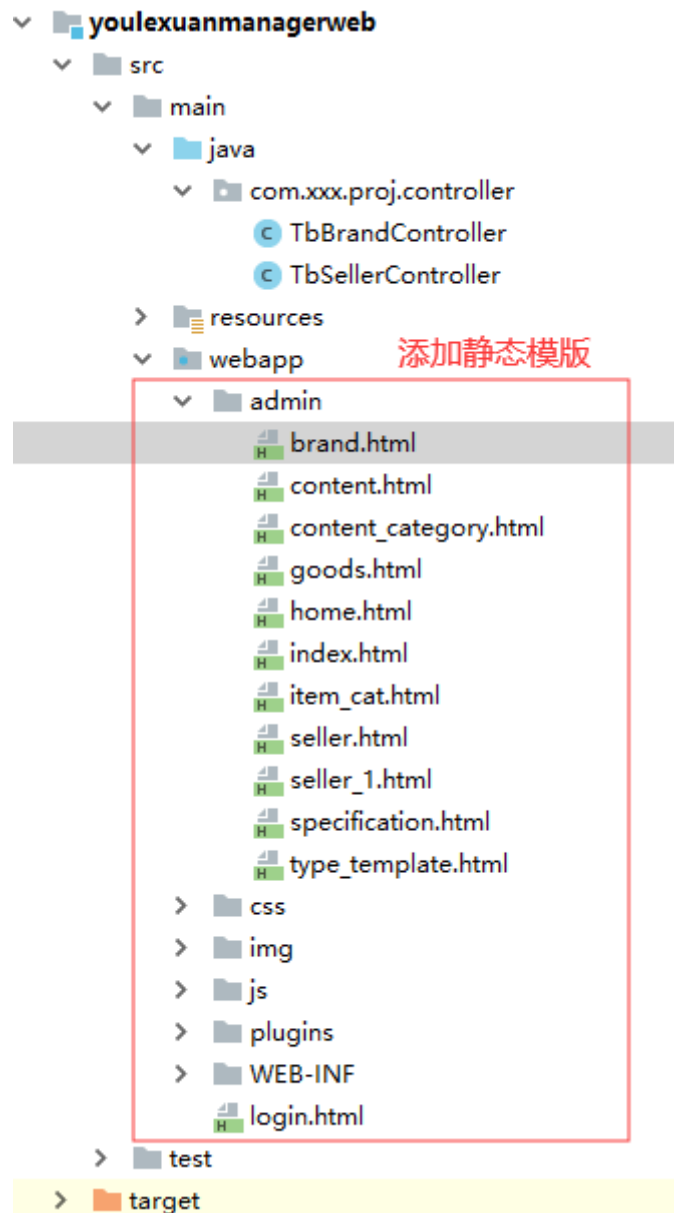
    @RequestMapping("/findAll")
    public List<TbBrand> findAll() {
        return service.findAll();
    }

    @RequestMapping("/findPage")
    public PageResult findPage(int pageNum, int pageSize) {
        return service.findPage(pageNum, pageSize);
    }
}

```

表现层

拷贝资源\静态原型\运营商管理后台



引入angularJS类库和分页类库

```
<script src="../../plugins/angularjs/angular.min.js"></script>
<link rel="stylesheet" href="../../plugins/angularjs/pagination.css">
<script src="../../plugins/angularjs/pagination.js"></script>
```

编写js代码，处理分页查询

```
<script>
var myApp = angular.module("youlexuan", ["pagination"]);
myApp.controller("brandController", function ($scope, $http) {
    //分页处理
    $scope.paginationConf = {
        currentPage: 1,
        itemsPerPage: 10,
        perPageOptions: [10, 20, 30],
        onChange: function () {
            $scope.reloadList();
        }
    };
    $scope.reloadList = function () {
```

```

        $http.get("../brand/findPage.do?pageNum=" +
$scope.paginationConf.currentPage + "&pageSize=" +
$scope.paginationConf.itemsPerPage).success(function (response) {
            $scope.list = response.rows;
            $scope.paginationConf.totalItems = response.total;
        })
    }
    $scope.findAll = function () {
        $http.get("../brand/findAll.do").success(function (response) {
            $scope.list = response;
        })
    }
}
})

</script>

```

指定有效域

```

<body ng-app="youlexuan" ng-controller="brandController" ng-init="reloadList()"
class="hold-transition skin-red sidebar-mini">

```

页面遍历数据并添加分页控件

```

<!-- 数据表格 -->
<div class="table-box" >

    <!--工具栏-->
    <div class="pull-left">
        <div class="form-group form-inline">
            <div class="btn-group">
                <button type="button" class="btn btn-default" title="新建"
data-toggle="modal"
                    data-target="#editModal"><i class="fa fa-file-o">
</i> 新建
                </button>
                <button type="button" class="btn btn-default" title="删除"><i
class="fa fa-trash-o"></i> 删除</button>
                <button type="button" class="btn btn-default" title="刷新"
onclick="window.location.reload();"><i
                    class="fa fa-refresh"></i> 刷新
                </button>
            </div>
        </div>
    </div>
    <div class="box-tools pull-right">
        <div class="has-feedback">

        </div>
    </div>
    <!--工具栏-->

    <!--数据列表-->
    <table id="dataList" class="table table-bordered table-striped table-
hover dataTable">
        <thead>
        <tr>

```

```

        <th class="" style="padding-right:0px">
            <input id="selall" type="checkbox" class="checkbox square-
blue">

        </th>
        <th class="sorting_asc">品牌ID</th>
        <th class="sorting">品牌名称</th>
        <th class="sorting">品牌首字母</th>
        <th class="text-center">操作</th>
    </tr>
</thead>
<tbody>
<tr ng-repeat="entity in list">
    <td><input type="checkbox"></td>
    <td>{{entity.id}}</td>
    <td>{{entity.name}}</td>
    <td>{{entity.firstChar}}</td>
    <td class="text-center">
        <button type="button" class="btn bg-olive btn-xs" data-
toggle="modal" data-target="#editModal">修改
        </button>
    </td>
</tr>

</tbody>
</table>
<!--数据列表/-->
<tm-pagination conf="paginationConf"></tm-pagination>

</div>

```

新增

DTO

youlexuan-pojo模块中增加一个通用消息数据传输对象。

```

package com.xxx.proj.dto;

import java.io.Serializable;

public class Result implements Serializable {
    private boolean success;
    private String message;

    public Result() {
    }

    public Result(boolean success, String message) {
        this.success = success;
        this.message = message;
    }

    public boolean isSuccess() {
        return success;
    }
}

```

```

    public void setSuccess(boolean success) {
        this.success = success;
    }

    public String getMessage() {
        return message;
    }

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

```

持久层

youlexuan-dao无需做任何修改

服务层

youlexuan-sellergoods-interface的TbBrandService增加接口

```
int add(TbBrand brand);
```

youlexuan-sellergoods-service的TbBrandServiceImpl增加实现方法

```

@Override
public int add(TbBrand brand) {
    return mapper.insert(brand);
}

```

控制层

youlexuan-manager-web的TbBrandController控制器增加方法

```

@RequestMapping("/add")
public Result add(@RequestBody TbBrand tbBrand) {
    try {
        service.add(tbBrand);
        return new Result(true, "新增品牌成功");
    } catch (Exception ex) {
        ex.printStackTrace();
        return new Result(false, "新增品牌失败");
    }
}

```

表现层

js部分处理保存数据提交

```

$scope.save = function () {
    $http.post("../brand/save.do", $scope.entity).success(function (response)
{
    if (response.success) {
        $scope.reloadList();
    } else {
        alert(response.message);
    }
    })
}

```

新增按钮增加旧数据清空

```

<button type="button" class="btn btn-default" title="新建" data-toggle="modal"
ng-click="entity={}" data-target="#editModal"><i class="fa fa-file-o"></i> 新建
</button>

```

dialog包含model绑定和提交方法调用

```

<div class="modal fade" id="editModal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">x</button>
                <h3 id="myModalLabel">品牌编辑</h3>
            </div>
            <div class="modal-body">
                <table class="table table-bordered table-striped" width="800px">
                    <tr>
                        <td>品牌名称</td>
                        <td><input ng-model="entity.name" class="form-control"
placeholder="品牌名称"></td>
                    </tr>
                    <tr>
                        <td>首字母</td>
                        <td><input ng-model="entity.firstChar" class="form-
control" placeholder="首字母"></td>
                    </tr>
                </table>
            </div>
            <div class="modal-footer">
                <button class="btn btn-success" data-dismiss="modal" aria-
hidden="true" ng-click="save()">保存</button>
                <button class="btn btn-default" data-dismiss="modal" aria-
hidden="true">关闭</button>
            </div>
        </div>
    </div>
</div>

```

修改

持久层

youlexuan-dao无需做任何修改

服务层

youlexuan-sellergoods-interface的TbBrandService增加接口

```
TbBrand findById(Long id);

int modify(TbBrand brand);
```

youlexuan-sellergoods-service的TbBrandServiceImpl增加实现方法

```
@Override
public TbBrand findById(Long id) {
    return mapper.selectByPrimaryKey(id);
}

@Override
public int modify(TbBrand brand) {
    return mapper.updateByPrimaryKey(brand);
}
```

控制层

youlexuan-manager-web的TbBrandController控制器增加方法

```
@RequestMapping("/findById")
public TbBrand findById(Long id) {
    return service.findById(id);
}

@RequestMapping("/modify")
public Result modify(@RequestBody TbBrand tbBrand) {
    try {
        service.modify(tbBrand);
        return new Result(true, "修改品牌成功");
    } catch (Exception ex) {
        ex.printStackTrace();
        return new Result(false, "修改品牌失败");
    }
}
```

表现层

js部分新增findById，并修改save方法

```
//增加根据主键获取完整对象
$scope.findById = function (id) {
    $http.get("../brand/findById.do?id=" + id).success(function (response) {
        $scope.entity = response;
    })
}
```

```

$scope.save = function () {
    var url;
    if ($scope.entity.id == null) {
        url = "../brand/add.do";
    }
    else {
        url = "../brand/modify.do";
    }
    $http.post(url, $scope.entity).success(function (response) {
        if (response.success) {
            $scope.reloadList();
        } else {
            alert(response.message);
        }
    })
}

```

html部分只修改了修改按钮的点击事件处理

```

<tbody>
    <tr ng-repeat="entity in list">
        <td><input type="checkbox"></td>
        <td>{{entity.id}}</td>
        <td>{{entity.name}}</td>
        <td>{{entity.firstChar}}</td>
        <td class="text-center">
            <button type="button" class="btn bg-olive btn-xs" data-
toggle="modal" data-target="#editModal" ng-click="findById(entity.id)">修改
            </button>
        </td>
    </tr>
</tbody>

```

删除

持久层

youlexuan-dao无需做任何修改

服务层

youlexuan-sellergoods-interface的TbBrandService增加接口

```
void remove(Long[] ids);
```

youlexuan-sellergoods-service的TbBrandServiceImpl增加实现方法

```

@Override
public void remove(Long[] ids) {
    for (Long id : ids) {
        mapper.deleteByPrimaryKey(id);
    }
}

```

控制层

youlexuan-manager-web的TbBrandController控制器增加方法

```
@RequestMapping("/remove")
public Result remove(Long[] ids) {
    try {
        service.remove(ids);
        return new Result(true, "删除品牌成功");
    } catch (Exception ex) {
        ex.printStackTrace();
        return new Result(false, "删除品牌失败");
    }
}
```

表现层

表现层处理列表多选和提交删除

```
//定义要删除的id数组
$scope.selectedIds = [];
//TODO 使用内置变量$event来获取checkbox的选中状态$event.target.checked
$scope.updateIds = function ($event, id) {
    if ($event.target.checked) {
        $scope.selectedIds.push(id);
    } else {
        //获取id所在位置
        var idx = $scope.selectedIds.indexOf(id);
        //移除该位置上的元素
        $scope.selectedIds.splice(idx, 1);
    }
}
//删除
$scope.remove = function () {
    $http.get("../brand/remove.do?ids=" + $scope.selectedIds).success(function (response) {
        if (response.success) {
            $scope.reloadList();
            $scope.selectedIds = [];
        } else {
            alert(response.message);
        }
    })
}
```

删除按钮事件

```
<button type="button" class="btn btn-default" title="删除" ng-click="remove()">
<i
                                class="fa fa-trash-o"></i> 删除
</button>
```

复选框按钮事件

```
<tbody>
```

```

        <tr ng-repeat="entity in list">
            <td><input type="checkbox" ng-click="updateIds($event,entity.id)"></td>
            <td>{{entity.id}}</td>
            <td>{{entity.name}}</td>
            <td>{{entity.firstChar}}</td>
            <td class="text-center">
                <button type="button" class="btn bg-olive btn-xs" data-
toggle="modal" data-target="#editModal"
                    ng-click="findById(entity.id)">修改
            </button>
        </td>
    </tr>

</tbody>

```

条件查询

持久层

youlexuan-dao无需做任何修改

服务层

youlexuan-sellergoods-interface的TbBrandService修改接口

```
PageResult findPage(TbBrand brand, int pageNum, int pageSize);
```

youlexuan-sellergoods-service的TbBrandServiceImpl修改实现方法

```

@Override
public PageResult findPage(TbBrand brand, int pageNum, int pageSize) {
    //这里使用PageHelper进行分页
    PageHelper.startPage(pageNum, pageSize);
    //条件判断
    TbBrandExample example = new TbBrandExample();
    //TODO ctrl+alt+v生成等号左边的声明部分
    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.andFirstCharEqualTo(brand.getFirstChar());
        }
    }
    Page<TbBrand> page = (Page<TbBrand>) mapper.selectByExample(example);
    return new PageResult(page.getTotal(), page.getResult());
}

```

控制层

youlexuan-manager-web的TbBrandController控制器修改方法

```
@RequestMapping("/findPage")
public PageResult findPage(@RequestBody TbBrand brand, int pageNum, int
pageSize) {
    return service.findPage(brand, pageNum, pageSize);
}
```

表现层

修改查询方法

```
$scope.searchEntity = {};
$scope.reloadList = function () {
    // $scope.paginationConf.itemsPerPage 每页条数
    // $scope.paginationConf.currentPage 当前页码
    // http://localhost:9002/brand/findPage.do?pageNum=1&pageSize=5
    // TODO 条件查询时，这里要改成POST请求，同时添加查询对象searchEntity
    $http.post("../brand/findPage.do?pageNum=" +
        $scope.paginationConf.currentPage + "&pageSize=" +
        $scope.paginationConf.itemsPerPage, $scope.searchEntity).success(function
        (response) {
            $scope.list = response.rows;
            $scope.paginationConf.totalItems = response.total;
        })
}
```

添加查询组建

```
<div class="box-tools pull-right">
    <div class="has-feedback">
        <div class="has-feedback">品牌名称:<input ng-model="searchEntity.name">
            品牌首字母:<input ng-model="searchEntity.firstChar">
            <button class="btn btn-default" ng-click="reloadList()">查询
    </button>
    </div>
</div>
</div>
```

效果

← → ↺ 🏠 ⓘ localhost:9002/admin/index.html ☆ 🔴 100 👤 ⋮

4 100 9 👤 测试用户

优乐选-运营商后台

测试用户
在线

菜单

首页

商家管理

商品管理

品牌管理

规格管理

模板管理

分类管理

商品审核

广告管理

品牌管理

新建 删除 刷新

品牌名称:

品牌首字母:

查询

<input type="checkbox"/>	品牌ID	品牌名称	品牌首字母	操作
<input type="checkbox"/>	3	三星	S	<div>修改</div>
<input type="checkbox"/>	6	360	S	<div>修改</div>
<input type="checkbox"/>	28	书旗	S	<div>修改</div>

« 1 »

第 1 页 每页 10 /共3条

🔍