主从表

拷贝生成代码

java部分

```
拷贝所有控制器到youlexuanmanagerweb
拷贝所有服务层接口到youlexuan-sellergoods-interface
拷贝所有服务层实现类到youlexuan-sellergoods-service
```

js部分

```
拷贝所有js控制器到youlexuanmanagerweb\src\main\webapp\js\controller
拷贝所有js服务层到youlexuanmanagerweb\src\main\webapp\js\service
```

基本增删改查（规格管理）

修改youlexuanmanagerweb\src\main\webapp\admin\specification.html

引入依赖

```
<script src="../plugins/angularjs/angular.min.js"></script>
    <!-- 分页组件开始 -->
    <script src="../plugins/angularjs/pagination.js"></script>
    <link rel="stylesheet" href="../plugins/angularjs/pagination.css">
    <script type="text/javascript" src="../js/base_pagination.js"></script>
    <!-- 分页组件结束 -->
    <script type="text/javascript" src="../js/controller/baseController.js">
</script>
    <script type="text/javascript" src="../js/service/specificationService.js">
</script>
    <script type="text/javascript"
src="../js/controller/specificationController.js"></script>
```

设置angular作用域

```
<body class="hold-transition skin-red sidebar-mini" ng-app="youlexuan" ng-
controller="specificationController">
```

设置新增、删除按钮事件

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

设置列表和编辑按钮

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

设置分页

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

设置表单

```
 <table class="table table-bordered table-striped" width="800px">
     <tr>
         <td>规格名称</td>
         <td><input class="form-control" ng-model="entity.specName" placeholder="规格名称">         </td>
     </tr>
</table>
```

保存按钮

```
<button class="btn btn-success" data-dismiss="modal" aria-hidden="true" ng-click="save()">保存</button>
```

子表处理（规格选项）

新增

js控制器

修改youlexuanmanagerweb\src\main\webapp\js\controller\specificationController.js

```
    //动态添加规格选项
    $scope.addOption = function () {
        if ($scope.entity.specOptionList == undefined)
            $scope.entity.specOptionList = [];
        $scope.entity.specOptionList.push({});
    }
    //动态移除规格选项
    $scope.delOption = function (index) {
        $scope.entity.specOptionList.splice(index, 1);
    }
```

html

修改youlexuanmanagerweb\src\main\webapp\admin\specification.html

```html
<tr ng-repeat="opt in entity.specOptionList">
    <td>
        <input class="form-control" placeholder="规格选项" ng-model="opt.optionName">
    </td>
    <td>
        <input class="form-control" placeholder="排序" ng-model="opt.orders">
    </td>
    <td>
        <button type="button" class="btn btn-default" title="删除" ng-click="delOption($index)"><i class="fa fa-trash-o"></i> 删除
        </button>
    </td>
</tr>
```

修改实体类

youlexuanpojo\src\main\java\com\xxx\proj\pojo\TbSpecification.java

```java
public class TbSpecification implements Serializable {
    private Long id;

    private String specName;
    //新增的成员变量和对应的getter，setter
    private List<TbSpecificationOption> specOptionList;
```

主表mapper的insert增加返回主键

修改youlexuandao\src\main\resources\com\xxx\proj\mapper\TbSpecificationMapper.xml

```xml
 <insert id="insert" parameterType="com.xxx.proj.pojo.TbSpecification" useGeneratedKeys="true" keyProperty="id" >
    insert into tb_specification (id, spec_name)
    values (#{id,jdbcType=BIGINT}, #{specName,jdbcType=VARCHAR})
  </insert>
```

修改服务层实现类

修改youlexuansellergoodsservice\src\main\java\com\xxx\proj\service\impl\SpecificationServiceImpl.java

```java
    @Autowired
    private TbSpecificationOptionMapper specificationOptionMapper;

    /**
     * 增加
     */
    @Override
    public void add(TbSpecification specification) {
        //添加主表数据
        //TODO 通过在主表mapper中添加useGeneratedKeys="true" keyProperty="id" ，返
回主键到specification.id
        specificationMapper.insert(specification);
        Long id = specification.getId();
        //循环添加从表数据
        for (TbSpecificationOption option : specification.getSpecOptionList()) {
            option.setSpecId(id);//设置外键
            specificationOptionMapper.insert(option);
        }
    }
```

读取和修改

修改服务层实现类

```
public void update(TbSpecification specification) {

    //删除所有旧的option
    TbSpecificationOptionExample example = new TbSpecificationOptionExample();
    TbSpecificationOptionExample.Criteria criteria = example.createCriteria();
    criteria.andSpecIdEqualTo(specification.getId());
    specificationOptionMapper.deleteByExample(example);

    //更新头表
    specificationMapper.updateByPrimaryKey(specification);
    //更新从表，添加新的option
    Long id = specification.getId();
    //循环添加从表数据
    for (TbSpecificationOption option : specification.getSpecOptionList()) {
        option.setSpecId(id);//设置外键
        specificationOptionMapper.insert(option);
    }
}

 public TbSpecification findOne(Long id) {
        //读取规格信息
        TbSpecification specification =
specificationMapper.selectByPrimaryKey(id);
        //读取规格选项信息
        TbSpecificationOptionExample example = new
TbSpecificationOptionExample();
        TbSpecificationOptionExample.Criteria criteria =
example.createCriteria();
        criteria.andSpecIdEqualTo(specification.getId());
        List<TbSpecificationOption> optionList =
specificationOptionMapper.selectByExample(example);
        specification.setSpecOptionList(optionList);

        return specification;
    }
```

删除

修改服务层实现类

```
public void delete(Long[] ids) {
    //循环主表id
    for (Long id : ids) {
        //删除从表数据
        TbSpecificationOptionExample example = new
TbSpecificationOptionExample();
        TbSpecificationOptionExample.Criteria criteria =
example.createCriteria();
        criteria.andSpecIdEqualTo(id);
        specificationOptionMapper.deleteByExample(example);
        //删除主表数据
        specificationMapper.deleteByPrimaryKey(id);
    }
}
```