

[今日课程大纲]

商品详情需求分析 完成商品详情 商品介绍功能 规格参数显示功能

[知识点详解]

一.商品详情需求分析

- 1. 在 ego-item 的 item.jsp 中完成 3 个功能
 - 1.1 显示商品详细信息
 - 1.2 显示商品的描述
 - 1.3 显示商品的规格参数
- 2. 商品详细信息通过作用域传递到页面\${item.xxx}
 - 2.1 商品对象包含了5个值
- 3. 显示商品描述功能
 - 3.1 使用 js 的 setTimeout 延迟 1 秒加载,提升商品详细信息
 - 3.2 以后项目中如果页面数据过大,可以考虑一些不是立即看见的 内容选择延迟加载.
- 4. 商品规格参数通过点击 tab 发送请求.



二. 完成商品详情

- 1. 把 ego-search 中的;TbItemChild 放入到 ego-commons 中
- 2. 在 ego-service 的 TbiltemDubboService 及实现类添加

```
/**
 * 根据主键查询
 * @param id
 * @return
 */
 TbItem selById(long id);

@Override
   public TbItem selById(long id) {
    return tbItemMapper.selectByPrimaryKey(id);
}
```

2. 在 ego-item 中新建 com.ego.item.service.TbltemService 及实现类

```
public interface TbItemService {
    /**
    * 显示商品详情
    * @param id
    * @return
    */
```

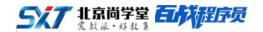


```
TbItemChild show(long id);
}
@Service
public class TbItemServiceImpl implements TbItemService
{
  @Reference
  private TbItemDubboService tbItemDubboServiceImpl;
  @Override
  public TbItemChild show(long id) {
     TbItem <u>item</u> = tbItemDubboServiceImpl.selById(id);
     TbItemChild child = new TbItemChild();
     child.setId(item.getId());
     child.setTitle(item.getTitle());
     child.setPrice(item.getPrice());
     child.setSellPoint(item.getSellPoint());
  child.setImages(item.getImage()!=null&&!item.equals
("")?item.getImage().split(","):new String[1]);
     return child;
  }
}
```



3. 在 ego-item 中 新 建
com.ego.item.controller.TbItemController

```
@Controller
public class TbItemController {
  @Resource
  private TbItemService tbItemSerivceImpl;
  /**
   * 显示商品详情
   * @param id
   * @param model
   * @return
   */
  @RequestMapping("item/{id}.html")
  public String showItemDetails(@PathVariable long
id,Model model){
     model.addAttribute("item",
tbItemSerivceImpl.show(id));
     return "item";
  }
}
```



三.添加缓存

1. 在 ego-redis 的 redis.properties 中添加 key

```
redis.item.key = item:
```

2. 在 ego-parent 添加对 ego-redis 的依赖

3. 在 ego-item 中添加对 ego-redis 的依赖

4. 在 ego-item 的 applicationContext-spring.xml 修改属性问价加载



5. 在 ego-item 的 TbltemServiceImpl 添加下面代码

```
@Resource
  private JedisDao jedisDaoImpl;
  @Value("${redis.item.key}")
  private String itemKey;
  @Override
  public TbItemChild show(long id) {
     String key = itemKey+id;
     if(jedisDaoImpl.exists(key)){
        String json = jedisDaoImpl.get(key);
        if(json!=null&&!json.equals("")){
           return JsonUtils.jsonToPojo(json,
TbItemChild.class);
     }
     TbItem item = tbItemDubboServiceImpl.selById(id);
     TbItemChild child = new TbItemChild();
     child.setId(item.getId());
     child.setTitle(item.getTitle());
     child.setPrice(item.getPrice());
     child.setSellPoint(item.getSellPoint());
```



```
child.setImages(item.getImage()!=null&&!item.equals
("")?item.getImage().split(","):new String[1]);
    //存到数据库中
    jedisDaoImpl.set(key,

JsonUtils.objectToJson(child));
    return child;
}
```

四. ego-manage 中商品删除,上架,下架同步数据

- 1. 商品上架不需要操作 redis,下架和删除都是从 redis 中删除 key
- 2. 在 ego-manage 的 TbltemServiceImpl 的 update()方法中添加

```
@Resource
private JedisDao jedisDaoImpl;
@Value("${redis.item.key}")
private String itemKey;
@Override

public int update(String ids, byte status) {
   int index = 0;
   TbItem item = new TbItem();
   String[] idsStr = ids.split(",");
```



```
for (String id : idsStr) {
        item.setId(Long.parseLong(id));
        item.setStatus(status);
        index +=
tbItemDubboServiceImpl.updItemStatus(item);
        if(status==2||status==3){
           jedisDaoImpl.del(itemKey+id);
        }
     }
     if (index == idsStr.length) {
        return 1;
     }
     return 0;
   }
```

五. 商品描述

1. 在 ego-redis 中 redis.properties 中

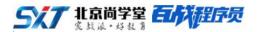
```
redis.desc.key = <u>desc</u>:
```

2. 在 ego-item 中新建 TbltemDescService

```
public interface TbItemDescService {
```



```
/**
   * 根据商品 id 显示商品描述
   * @param itemId
   * @return
   */
  String showDesc(long itemId);
}
@Service
public class TbItemDescServiceImpl implements
TbItemDescService {
  @Reference
  private TbItemDescDubboService
tbItemDescDubboServiceImpl;
  @Resource
  private JedisDao jedisDaoImpl;
  @Value("${redis.desc.key}")
  private String descKey;
  @Override
  public String showDesc(long itemId) {
     String key = descKey + itemId;
     if(jedisDaoImpl.exists(key)){
        String json = jedisDaoImpl.get(key);
```



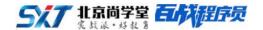
```
if(json!=null&&!json.equals("")){
    return json;
}

String result =
tbItemDescDubboServiceImpl.selByItemid(itemId).getIte
mDesc();
    jedisDaoImpl.set(key, result);
    return result;
}
```

3. 在 ego-item 中新建 TbItemDescController

```
@Controller

public class TbItemDescController {
    @Resource
    private TbItemDescService tbItemDescServiceImpl;
    /**
    * 显示商品详情
    * @param id
    * @return
    */
```



```
@RequestMapping(value="item/desc/{id}.html",produce
s="text/html;charset=utf-8")
  @ResponseBody
  public String desc(@PathVariable long id){
    return tbItemDescServiceImpl.showDesc(id);
  }
}
```

五. 为了测试方便,设置 Solr 按照 updated 降序显示

1. 在 solrhome/collection1/conf/schema.xml 添加 field

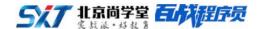
```
<field name="item_updated" type="date" indexed="true" stored="false"/>
```

2. 在 ego-search 的 init()方法中添加

```
doc.setField("item_updated", item.getUpdated());
```

3. 在 ego-search 的 <u>selByQuery</u> 中添加

```
//添加排序
params.setSort("item_updated", ORDER.desc);
```



六. 实现商品规格参数

1. 在 ego-service 新建 TbltemParamItemDubboService

```
public interface TbItemParamItemDubboSerice {
  /**
   * 根据商品 id 查询商品规格参数
   * @param itemId
   * @return
   */
  TbItemParamItem selByItemid(long itemId);
}
public class TbItemParamItemDubboServiceImpl
implements TbItemParamItemDubboSerice{
  @Resource
  private TbItemParamItemMapper tbItemParamItemMapper;
  @Override
  public TbItemParamItem selByItemid(long itemId) {
     TbItemParamItemExample example = new
TbItemParamItemExample();
  example.createCriteria().andItemIdEqualTo(itemId);
     List<TbItemParamItem> list =
```



```
tbItemParamItemMapper.selectByExampleWithBLOBs(exampl
e);
    if(list!=null&&list.size()>0)
        return list.get(0);
    return null;
}
```

2. 在 ego-service-impl 的 applicationContext-dubbo.xml

3. 在 ego-item 中新建

```
public class ParamItem {
    private String group;
    private List<ParamNode> params;
```



```
public class ParamNode {
   private String k;
   private String v;
```

4. 在 ego-item 中新建 TbItemParamItemService

```
public interface TbItemParamItemService {
  /**
   * 显示商品规格参数
   * @param itemId
   * @return
   */
  String showParam(long itemId);
}
@Service
public class TbItemParamItemServiceImpl implements
TbItemParamItemService {
  @Reference
  private TbItemParamItemDubboSerice
tbItemParamItemDubboSericeImpl;
  @Override
  public String showParam(long itemId) {
```



```
TbItemParamItem item =
tbItemParamItemDubboSericeImpl.selByItemid(itemId);
    List<ParamItem> list =
JsonUtils.jsonToList(item.getParamData(),
ParamItem.class);
    System.out.println(list);
    StringBuffer sf = new StringBuffer();
    for (ParamItem param : list) {
       sf.append("<table width='500'
style='color:gray;'>");
       for (int i = 0;i<param.getParams().size();i++)</pre>
{
         if(i==0){
           sf.append("");
           sf.append("
width='30%'>"+param.getGroup()+"");
           sf.append("
);
  sf.append(""+param.getParams().get(i).getV()+"<</pre>
```

```
/td>");
              sf.append("");
           }else{
              sf.append("");
              sf.append(" ");
              sf.append("<td</pre>
align='right'>"+param.getParams().get(i).getK()+"</td</pre>
<mark>>"</mark>);
  sf.append(""+param.getParams().get(i).getV()+"<</pre>
/td>");
              sf.append("");
           }
        }
        sf.append("");
        sf.append("<hr style='color:gray;'/>");
     }
     return sf.toString();
  }
}
```

5. 在 ego-item 中新建控制器



```
@Controller
public class TbItemParamItemController {
  @Resource
  private TbItemParamItemService
tbItemParamItemServiceImpl;
  /**
   * 显示规格参数
   * @param id
   * @return
   */
  @RequestMapping(value="item/param/{id}.html",produc
es="text/html;charset=utf-8")
  @ResponseBody
  public String param(@PathVariable long id){
     return tbItemParamItemServiceImpl.showParam(id);
  }
}
```