# 今日任务

1. 商品详情页分析
2. 详情页设计策略
3. 技术选型
4. 页面实现

# 详情页

## 页面分析

1. 基本信息部分



1. 详细介绍部分



1. 属性信息部分



1. 评价信息



## 分析总结

如图所示，商品页主要包括商品基本信息（基本信息、图片列表、规格、扩展属性、包装清单、售后保障等）、商品介绍、其他信息（分类、品牌、同类相关品牌）以及评论信息等等，不同网站还有细微不同点。

总结特点就是，商品详情页是电商网站中最复杂的页面，数据来源之广，当有大量商品时，如果每次动态获取如上内容进行模板拼装，数据来源之多足以造成性能无法满足要求，所以有必要在设计时考虑性能优化的问题。

1. 先说一种简单粗暴的解决方法，就是直接利用模板生成静态页，但是静态页的最大问题就是，无法迅速响应页面需求变更；另一个就是会生成大量的页面文件，且无法满足个性化、多样化页面需求。
2. 仔细分析页面数据，发现商品的基本信息可以直接展示，而详细、规格属性及评论等数据，则可以实现异步加载的策略。再进一步对数据进行细分，有些数据是不经常变化的，如基本信息，可以放在缓存中；而库存、价格等信息，是活跃的，可以使用ajax技术实时进行更新。对缓存中的失效数据，同步更新即可。

## 技术选型

1. 使用thymeleaf 模板动态拼接数据
2. Ajax技术异步加载详情、规格等信息
3. 使用redis缓存商品信息

# 结构图

详情页面：thymeleaf

ajax异步

同步

同步

Redis缓存

DB：mysql

缓存有数据

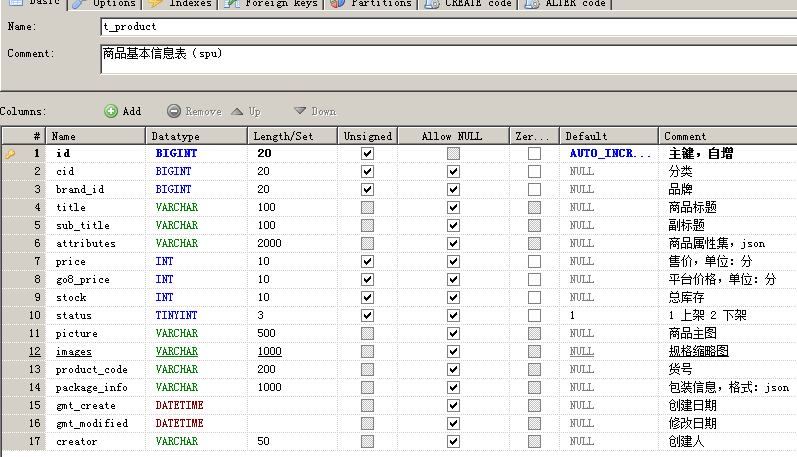
缓存无数据

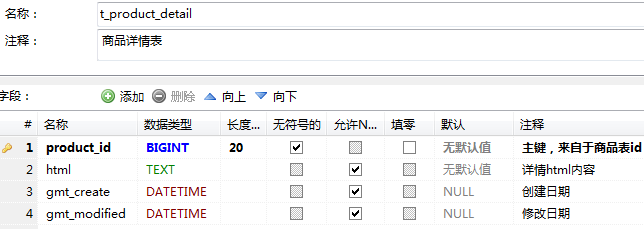
同步

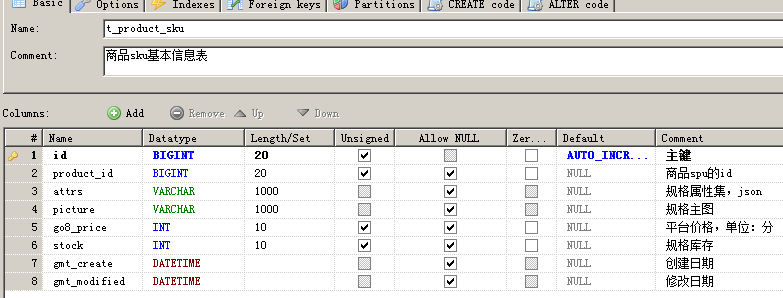
同步

# 无缓存版本

## 表结构







## 接口

|  |  |  |  |
| --- | --- | --- | --- |
| **业务类型** | **method** | **url** | **return body** |
| 查询基本信息 | get | products/basic?id | 返回一个统一的json  对象，  {  “code”: xxx,  “message”:”xxxx”,  “data”:{} or []  } |
| 查询详情 | get | products/detail?id |
| 查询规格 | get | products/attrs?id |
| 查询sku | get | products/sku?id |

# 缓存版本

预习reids。