# 今日任务

1. 熟悉商城项目特点、创建商城项目
2. 理解首页分类加载原理及功能实现（重点）
3. cms概念
4. 首页内容分析
5. 表设计
6. 实现轮播广告位管理
7. **Thymeleaf**入门

# 商城项目

商城项目是整个项目的核心，所有后台的付出，最后都要靠这个商城来展示给用户。像天猫商城，京东商城。那么面向大流量用户群体的商城项目自然和普通的后台支撑系统相比，有明显的不同。

1. 界面设计要更专业，更漂亮
2. 利于搜索、需要进行SEO的优化
3. 页面伪静态化
4. 使用模板真静态化
5. 响应时间有严格要求
6. 经得起大流量的高并发访问
7. 其他。。。。。。。

总结起来一点就是，对技术要求更高了，尤其是搞活动时，面对超高的瞬时并发，依然能保证服务的可用性，是衡量一家公司技术水准的关键指标。阿里的双十一，一次又一次的突破了极限，刷新了我们对技术的认知。也就是只要你技术够牛逼，PC级的服务器，你也能玩出**超级计算机**的感觉。Spring之父Rod Johnson当年在写J2EE指南的时候，是极力推荐轻量级单体结构的。Spring框架当初就是以救世主的身份出现，拯救了所有深陷权威中，使用分布式EJB的java程序员的。唉！！时代真是变了，现在你出去找工作，不吹个分布式你都不好意思讲话，C:\Users\ADMINI~1\AppData\Local\Temp\SGPicFaceTpBq\1336\02F2DFE3.pngC:\Users\ADMINI~1\AppData\Local\Temp\SGPicFaceTpBq\1336\02F2F057.png。

# go8-mall

## 架构图

客户端（应用中心），展示层

商品服务

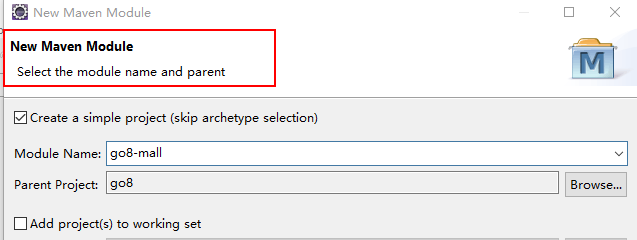
Pc-mall

App-mall

…………….

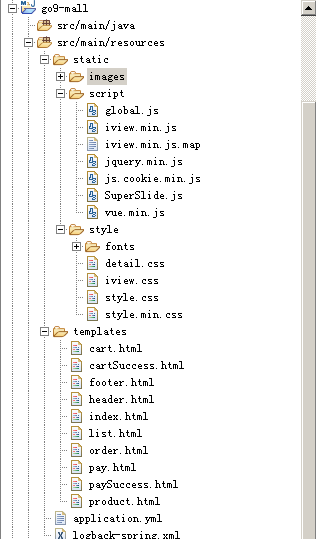
Rpc(调用)

## 创建工程



## 静态资源

见resources文件夹



## 配置项目

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.go9</groupId>  <artifactId>go9</artifactId>  <version>1.0.0-SNAPSHOT</version>  </parent>  <artifactId>go9-mall</artifactId>  <dependencies>  <dependency>  <groupId>com.go9</groupId>  <artifactId>go9-common</artifactId>  <version>1.0.0-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-thymeleaf -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <!-- 热部署模块 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <optional>true</optional> <!-- 这个需要为 true 热部署才有效 -->  </dependency>  </dependencies>  <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build>  </project> |

|  |
| --- |
| server:  port: 8888  #eureka:  # client:  # serviceUrl:  # defaultZone: http://localhost:666/eureka/  spring:  application:  name: go9-mall  thymeleaf:  cache: **false**  logging:  level:  root: info  #custom properties  #LOGIN\_URL: http://localhost:3030/login.html |

## Thymeleaf

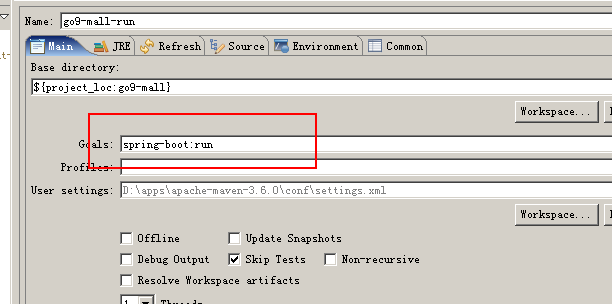
Jsp这种页面模板，已经逐渐退出历史舞台，无论是前后端的彻底分离，还是未分离，都无条件的向“静态化”靠拢，这就是技术的时代性。

简单入门。配合前端代码实现循环输出“顶级类目”到页面的功能。

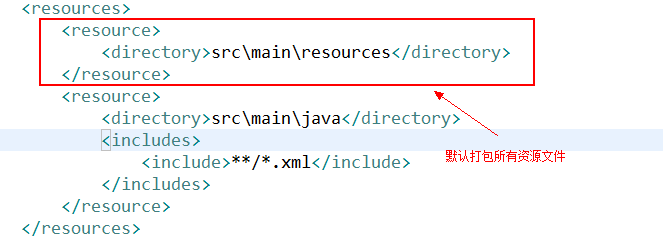
## 代码

|  |
| --- |
| **package** com.go9.mall.controller;  **import** org.springframework.stereotype.Controller;  **import** org.springframework.web.bind.annotation.RequestMapping;  @Controller  **public** **class** IndexController {  @RequestMapping("/")  **public** String index() {  **return** "index";  }  } |

## 启动项目



注意可能误操作引出的坑：如果加载不到静态资源文件，检查一下这里。



访问首页，不考虑数据，如果成功，算是各种资源整合成功。

思考问题：首页的分类数据如何获取呢？

# 分类加载



演示淘宝或者京东商城首页分类的加载过程。得到如下的加载流程：

首页显示默认顶级节点名称

加载所有分类数据

鼠标上移事件

事件

是否是第一次上移

是

是

不是

将已经加载的数据

根据顶级类，

解析相应的数据显示

根据当前鼠标指向的分类显示相应数据

***需要使用ajax跨域请求，需要服务器配合***

***不需要服务器配合，由前端js完成***

***不需要服务器配合，由前端js完成***

# 分类服务

在商品服务项目中，新发布一个供商城调用的加载分类数据的接口，这个接口要一次性完成所有分类数据查询的功能。注意下递归查询就行。

# 接口约定

|  |  |  |  |
| --- | --- | --- | --- |
| **业务类型** | **method** | **url** | **return body** |
| 加载所有分类 | get | catalogs/all?pid=0 | 返回一个统一的json  对象，  {  “code”: xxx,  “message”:”xxxx”,  “data”:{} or []  } |
| xxxxx | xxxxx | xxxxxx |

# 代码

|  |
| --- |
| /\*\*  \* 递归查询  \*/  @Override  **public** List<Catalog> getAllByPid(**long** pid) {  List<Catalog> parents = catalogmapper.selectByPid(pid);  **for**(Catalog cat : parents) {  //先判断是否父节点  **if**(cat.getYesParent()) {  List<Catalog> children = getAllByPid(cat.getId());  cat.setChildren(children);  }  }  **return** parents;  } |

# 测试

|  |
| --- |
| **package** com.go9.goods.service;  **import** java.util.List;  **import** org.junit.Test;  **import** org.junit.runner.RunWith;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.boot.test.context.SpringBootTest;  **import** org.springframework.test.context.junit4.SpringJUnit4ClassRunner;  **import** com.go9.goods.pojo.Catalog;  @RunWith(SpringJUnit4ClassRunner.**class**)  @SpringBootTest  **public** **class** CatalogServiceTest {  @Autowired  **private** CatalogService catalogService;  @Test  **public** **void** testGetAllByPid() {  List<Catalog> list = catalogService.getAllByPid(0L);  System.***out***.println(list);  }  } |

# cms

## 概念

内容管理系统（Content Management System，简称CMS） 在不同行业有着不同的应用范围，电商中CMS系统可以简单理解为用作**页面动态配置**的系统。 首页设置、 活动页面配置、 自定义新页面等都属于此类范畴。 在不同的电商体系内部的叫法可能不同。

## 作用

CMS系统与促销系统一样，对电商的运营人员相当重要，不论是节日活动还是日常售卖，都需要CMS系统的支持。 CMS系统给予促销活动前端的展示位和展示方式，吸引用户去浏览购买商品。

## 内容组件

常用组件主要有图片、 文本、 商品、 商品排行榜、 优惠券、 导航栏、 HTML代码等方式。

## 页面管理

首页，其他模块页面

## 文章管理

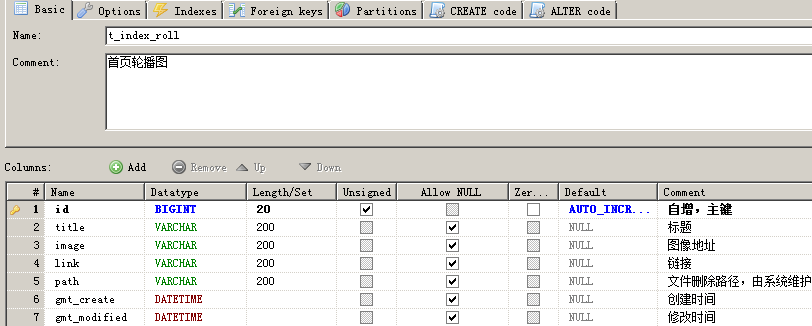
公告、新闻，文章等等

# 首页内容

分析淘宝、京东等电商网站首页，发现大部分内容都是需要动态设置的，如轮播图、广告、 导航、 公告、特卖、 团购、 倒计时、友情链接等等，运营新建页面或者配置首页时，利用模块的样式，添加相应的元素内容，定义模块顺序，最后决定最终显示效果。

我们课程以其中的大广告轮播管理为例，来讲解如何动态配置页面。

# 表设计



# 接口表

首页轮播接口

|  |  |  |  |
| --- | --- | --- | --- |
| **业务类型** | **method** | **url** | **return body** |
| 添加 | post | contents/index/roll | 返回一个统一的json  对象，  {  “code”: xxx,  “message”:”xxxx”,  “data”:{} or []  } |
| 修改 | put | contents/index/roll/{id} |
| 删除 | delete | contents/index/roll /{id} |
| 查询单体 | get | contents/index/roll/{id} |
| 分页查询 | get | contents/index/roll?page&size&[可选条件] |

# 流程图

go8-admin

content-service

mysql

go8-mall

管理动态内容

获取内容展示

# 内容服务

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE generatorConfiguration  PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"  "http://mybatis.org/dtd/mybatis-generator-config\_1\_0.dtd">  <generatorConfiguration>  <classPathEntry location="mysql-connector-java-8.0.15.jar" />  <context id="cjl-sc" targetRuntime="MyBatis3">  <commentGenerator>  <property name="suppressAllComments" value="true" />  </commentGenerator>  <jdbcConnection driverClass="com.mysql.cj.jdbc.Driver"  connectionURL="jdbc:mysql://localhost:3306/go9?characterEncoding=utf-8"  userId="root" password="111111">  </jdbcConnection>  <javaTypeResolver>  <property name="forceBigDecimals" value="false" />  </javaTypeResolver>  <javaModelGenerator targetPackage="com.go9.cms.pojo"  targetProject="src">  <property name="enableSubPackages" value="true" />  <property name="trimStrings" value="true" />  </javaModelGenerator>  <sqlMapGenerator targetPackage="com.go9.cms.mapper"  targetProject="src">  <property name="enableSubPackages" value="true" />  </sqlMapGenerator>  <javaClientGenerator type="XMLMAPPER"  targetPackage="com.go9.cms.mapper" targetProject="src">  <property name="enableSubPackages" value="true" />  </javaClientGenerator>    <table tableName="t\_index\_roll" domainObjectName="IndexRoll"  enableSelectByExample="false" enableDeleteByExample="false"  enableCountByExample="false" enableUpdateByExample="false"  selectByExampleQueryId="false">  <generatedKey column="id" sqlStatement="mysql" identity="true" />  </table>  <!-- <table tableName="t\_brand" domainObjectName="Brand"  enableSelectByExample="false" enableDeleteByExample="false"  enableCountByExample="false" enableUpdateByExample="false"  selectByExampleQueryId="false">  <generatedKey column="id" sqlStatement="mysql" identity="true" />  <columnOverride column="is\_parent" property="yesParent" javaType="java.lang.Boolean" />  </table> -->    <!-- <table tableName="t\_attribute" domainObjectName="Attribute"  enableSelectByExample="false" enableDeleteByExample="false"  enableCountByExample="false" enableUpdateByExample="false"  selectByExampleQueryId="false">  <generatedKey column="id" sqlStatement="mysql" identity="true" />  </table> -->    <!-- <table tableName="t\_product" domainObjectName="Product"  enableSelectByExample="false" enableDeleteByExample="false"  enableCountByExample="false" enableUpdateByExample="false"  selectByExampleQueryId="false">  <generatedKey column="id" sqlStatement="mysql" identity="true" />  </table>  <table tableName="t\_product\_sku" domainObjectName="Sku"  enableSelectByExample="false" enableDeleteByExample="false"  enableCountByExample="false" enableUpdateByExample="false"  selectByExampleQueryId="false">  <generatedKey column="id" sqlStatement="mysql" identity="true" />  </table>  <table tableName="t\_product\_detail" domainObjectName="Detail"  enableSelectByExample="false" enableDeleteByExample="false"  enableCountByExample="false" enableUpdateByExample="false"  selectByExampleQueryId="false">  </table> -->  </context>  </generatorConfiguration> |

启动类，调整路径：

|  |
| --- |
| package com.go9;  import org.mybatis.spring.annotation.MapperScan;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  @MapperScan({"com.go9.goods.mapper","com.go9.cms.mapper"})  public class GoodsServiceMain {  public static void main(String[] args) {  SpringApplication.run(GoodsServiceMain.class, args);  }  } |

|  |
| --- |
| **package** com.go9.cms.controller;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.web.bind.annotation.DeleteMapping;  **import** org.springframework.web.bind.annotation.GetMapping;  **import** org.springframework.web.bind.annotation.PostMapping;  **import** org.springframework.web.bind.annotation.RequestBody;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.RestController;  **import** com.go9.cms.pojo.IndexRoll;  **import** com.go9.cms.service.IndexRollService;  **import** com.go9.common.ServiceMessage;  @RestController  @RequestMapping("/admin/indexroll")  **public** **class** IndexRollController {  @Autowired  **private** IndexRollService indexRollService;    @GetMapping  **public** ServiceMessage getAll() {  **try** {  List<IndexRoll> all = indexRollService.getAll();  **return** ServiceMessage.*ok*(all);  } **catch** (Exception e) {  e.printStackTrace();  **return** ServiceMessage.*error*();  }  }    @PostMapping  **public** ServiceMessage add(@RequestBody IndexRoll indexRoll) {  **try** {  indexRollService.add(indexRoll);  **return** ServiceMessage.*ok*();  } **catch** (Exception e) {  e.printStackTrace();  **return** ServiceMessage.*error*();  }  }    @DeleteMapping  **public** ServiceMessage delete(**long** id) {  **try** {  indexRollService.delete(id);  **return** ServiceMessage.*ok*();  } **catch** (Exception e) {  e.printStackTrace();  **return** ServiceMessage.*error*();  }  }  } |

|  |
| --- |
| **package** com.go9.cms.service;  **import** java.util.Date;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Service;  **import** com.go9.cms.mapper.IndexRollMapper;  **import** com.go9.cms.pojo.IndexRoll;  @Service  **public** **class** IndexRollServiceImpl **implements** IndexRollService {  @Autowired  **private** IndexRollMapper indexRollMapper;    @Override  **public** List<IndexRoll> getAll() {  **return** indexRollMapper.selectAll();  }  @Override  **public** **void** add(IndexRoll indexRoll) {  Date now = **new** Date();  indexRoll.setGmtCreate(now);  indexRoll.setGmtModified(now);  indexRollMapper.insertSelective(indexRoll);  }  @Override  **public** **void** delete(**long** id) {  indexRollMapper.deleteByPrimaryKey(id);  }  } |

# 联合测试

启动内容服务、启动前端go8-admin工程，添加轮播广告。

