**首页、商品详情、加入购物车**

# 数据库表

|  |
| --- |
| ---商品信息表  CREATE TABLE `tb\_item` (  `id` bigint(20) NOT NULL COMMENT '商品id，同时也是商品编号',  `title` varchar(100) NOT NULL COMMENT '商品标题',  `sell\_point` varchar(500) DEFAULT NULL COMMENT '商品卖点',  `price` bigint(20) NOT NULL COMMENT '商品价格，单位为：分',  `num` int(10) NOT NULL COMMENT '库存数量',  `barcode` varchar(30) DEFAULT NULL COMMENT '商品条形码',  `image` varchar(500) DEFAULT NULL COMMENT '商品图片',  `parent\_id` bigint(10) NOT NULL COMMENT '所属类目，叶子类目',  `cid` bigint(10) NOT NULL COMMENT '所属类目，叶子类目',  `status` tinyint(4) NOT NULL DEFAULT '1' COMMENT '商品状态，1-正常，2-下架，3-删除',  `created` datetime NOT NULL COMMENT '创建时间',  `updated` datetime NOT NULL COMMENT '更新时间',  PRIMARY KEY (`id`),  KEY `cid` (`cid`),  KEY `status` (`status`),  KEY `updated` (`updated`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='商品表';  商品分类  CREATE TABLE `tb\_item\_cat` (  `id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '类目ID',  `parent\_id` bigint(20) DEFAULT NULL COMMENT '父类目ID=0时，代表的是一级的类目',  `name` varchar(50) DEFAULT NULL COMMENT '类目名称',  `img` varchar(150) DEFAULT NULL COMMENT '图片地址',  `status` int(1) DEFAULT '1' COMMENT '状态。可选值:1(正常),2(删除)',  `sort\_order` int(4) DEFAULT NULL COMMENT '排列序号，表示同级类目的展现次序，如数值相等则按名称次序排列。取值范围:大于零的整数',  `is\_parent` tinyint(1) DEFAULT '1' COMMENT '该类目是否为父类目，1为true，0为false',  `created` datetime DEFAULT NULL COMMENT '创建时间',  `updated` datetime DEFAULT NULL COMMENT '创建时间',  PRIMARY KEY (`id`),  KEY `parent\_id` (`parent\_id`,`status`) USING BTREE,  KEY `sort\_order` (`sort\_order`)  ) ENGINE=InnoDB AUTO\_INCREMENT=1183 DEFAULT CHARSET=utf8 COMMENT='商品类目';  商品描述  CREATE TABLE `tb\_item\_desc` (  `id` bigint(20) NOT NULL COMMENT '商品ID',  `itemdesc` text COMMENT '商品描述',  `created` datetime DEFAULT NULL COMMENT '创建时间',  `updated` datetime DEFAULT NULL COMMENT '更新时间',  PRIMARY KEY (`id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='商品描述表'; |

# 首页功能

## 商品服务

itmayiedu-shopp-commodity

## 引入商品服务依赖

|  |
| --- |
| **<dependencies>**  **<dependency>**  **<groupId>com.itmayiedu</groupId>**  **<artifactId>itmayiedu-shopp-commodity-api</artifactId>**  **<version>0.0.1-SNAPSHOT</version>**  **</dependency>**  **<dependency>**  **<groupId>com.itmayiedu</groupId>**  **<artifactId>itmayiedu-shopp-common</artifactId>**  **<version>0.0.1-SNAPSHOT</version>**  **</dependency>**  **<!-- springboot整合mybatis -->**  **<dependency>**  **<groupId>org.mybatis.spring.boot</groupId>**  **<artifactId>mybatis-spring-boot-starter</artifactId>**  **<version>1.3.0</version>**  **</dependency>**  **</dependencies>** |

## 引入商品服务配置文件

|  |
| --- |
| **server:**  **port: 8765**  **# context-path: /member**  **eureka:**  **client:**  **serviceUrl:**  **defaultZone: http://localhost:8761/eureka/**  **spring:**  **application:**  **name: commodity**  **redis:**  **host: 192.168.110.163**  **password: 123456**  **port: 6379**  **pool:**  **max-idle: 100**  **min-idle: 1**  **max-active: 1000**  **max-wait: -1**  **#数据库连接信息**  **datasource:**  **name: test**  **url: jdbc:mysql://127.0.0.1:3306/itmayiedu-shopp**  **username: root**  **password: root**  **# 使用druid数据源**  **type: com.alibaba.druid.pool.DruidDataSource**  **driver-class-name: com.mysql.jdbc.Driver**  **filters: stat**  **maxActive: 20**  **initialSize: 1**  **maxWait: 60000**  **minIdle: 1**  **timeBetweenEvictionRunsMillis: 60000**  **minEvictableIdleTimeMillis: 300000**  **validationQuery: select 'x'**  **testWhileIdle: true**  **testOnBorrow: false**  **testOnReturn: false**  **poolPreparedStatements: true**  **maxOpenPreparedStatements: 20** |

## 数据库访问层代码

|  |
| --- |
| **@Mapper**  **---商品分类**  **public interface ItemCatDao {**  **@Select("select id as id , parent\_id as parentId ,name as name, img as img,status as status , is\_parent as isParent, created as created ,updated as updated from tb\_item\_cat")**  **public List<ItemCatEntity> allItemCat();**  **}**  **---商品信息**  **@Mapper**  **public interface ItemDao extends BaseDao {**  **@Select("select a.id as id ,a.title as title ,a.sell\_point as sellPoint, a.price as price,a.num as num,a.barcode as barcode,a.image as image ,a.parent\_id as parentId,a.cid as cid,a.status as status,a.created as created,a.updated as updated from tb\_item as a inner join tb\_item\_cat as b on a.parent\_id=b.id where b.id =#{id} LIMIT 0,8")**  **public List<ItemEntity> getIndexItem(@Param("id") Long id);**  **@Select("select a.id as id ,a.title as title ,a.sell\_point as sellPoint, a.price as price,a.num as num,a.barcode as barcode,a.image as image ,a.parent\_id as parentId,a.cid as cid,a.status as status,a.created as created,a.updated as updated from tb\_item as a where a.id =#{id}")**  **public ItemEntity getItem(@Param("id") Long id);**  **}**  **商品详情**  **@Mapper**  **public interface ItemDescDao extends BaseDao {**  **@Select(" SELECT id AS id , itemDesc as itemDesc , created as created,updated as updated FROM tb\_item\_desc where id=#{id} ")**  **public ItemDescEntity getItemDesc(@Param("id") Long id);**  **}** |

## 业务逻辑层代码

|  |
| --- |
| **@RestController**  **public class ItemServiceImpl extends BaseApiService implements ItemService {**  **@Autowired**  **private ItemDao itemDao;**  **@Autowired**  **private ItemCatDao itemCatDao;**  **@RequestMapping("/getIndexItem")**  **public Map<String, Object> getIndexItem() {**  **// 查询所有的类型**  **List<ItemCatEntity> listItemCat = itemCatDao.allItemCat();**  **Map<String, Object> reuslt = new HashMap<String, Object>();**  **for (ItemCatEntity itemCatEntity : listItemCat) {**  **Long id = itemCatEntity.getId();**  **String name = itemCatEntity.getName();**  **List<ItemEntity> listItem = itemDao.getIndexItem(id);**  **if (!(listItem.isEmpty() && listItem.size() <= 0)) {**  **reuslt.put(name, listItem);**  **}**  **}**  **return setResutSuccessData(reuslt);**  **}**  **@Override**  **public Map<String, Object> geItem(@RequestParam("id") Long id) {**  **ItemEntity item = itemDao.getItem(id);**  **if(item==null){**  **return setResutError("没有查询到结果");**  **}**  **return setResutSuccessData(item);**  **}**  **}** |

## 页面展示

# 购物车

|  |
| --- |
| **@Autowired**  **private BaseRedisService baseRedisService;**  **public Map<String, Object> addShopping(String userId, String itemId) {**  **String value = baseRedisService.get(userId);**  **if (StringUtils.*isEmpty*(value)) {**  **JSONObject root = new JSONObject();**  **root.put(itemId, itemId);**  **baseRedisService.set(userId + "", root.toJSONString(), Constants.*WEBUSER\_SHOP\_TERMVALIDITY*);**  **} else {**  **JSONObject parseObject = new JSONObject().*parseObject*(value);**  **String redisiTemId=parseObject.getString(itemId);**  **if(!StringUtils.*isEmpty*(redisiTemId)){**  **return setResutError("该商品已经加入到购物车");**  **}**  **//添加到redis中**  **parseObject.put(itemId, itemId);**  **baseRedisService.set(userId, parseObject.toJSONString(), Constants.*WEBUSER\_SHOP\_TERMVALIDITY*);**  **}**  **return setResutSuccess("加入购物车成功!");**  **}**  **public Map<String, Object> remoShopping(String userId, String itemId) {**  **String value = baseRedisService.get(userId);**  **if (StringUtils.*isEmpty*(value)) {**  **return setResutError("该商品没有加入过购物车");**  **}**  **JSONObject parseObject = new JSONObject().*parseObject*(value);**  **parseObject.remove(itemId);**  **baseRedisService.set(userId, parseObject.toJSONString(), Constants.*WEBUSER\_SHOP\_TERMVALIDITY*);**  **return setResutSuccess("删除购物车成功!");**  **}** |

# 引入商品服务配置文件

# 购物车前端页面

|  |
| --- |
| **<script type=*"text/javascript"* src=*"js/jquery-3.2.1.min.js"*></script>**  **<script type=*"text/javascript"*>**  **$(document).ready(function (){**  **$("#bt\_sp").click(function (){**  **var itemId=$("#itemId").val();**  **$.ajax({url:"addShopping?itemId="+itemId,success:function(result){**  **alert(result.msg)**  **}});**  **})**  **});**  **</script>** |