**新巴巴运动网 项目第二天**

**传智.上官云**

# 课前回顾

1. SVN服务器 提交 检出
2. Spring+Mybatis整合 分块 （复制）
3. 整合Springmvc 第一个前端控制器
4. 整合Dubbo 分布式服务框架 log4j

# 今天内容

1. 优化Dubbo 三项 ：1）设置连接超时时间 10分钟 2）直接连接 3）设置不检查
2. 后台管理
3. 品牌管理之列表查询
4. 去品牌修改页面（查询一个品牌）
5. 上传图片（异步上传）接收图片
6. FastDFS的原理 及Java接口的工具类
7. 保存到分布式文件系统FastDFS
8. 回显图片
9. 提交修改

# 优化Dubbo

zookeeper

服务提供方

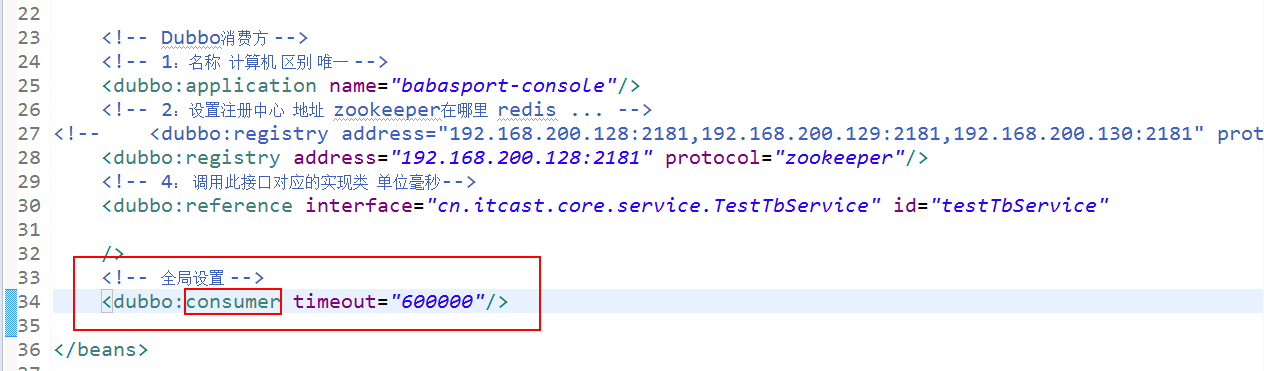
Service-product

服务消费方

服务提供方

Service-product

## 设置连接超时时间

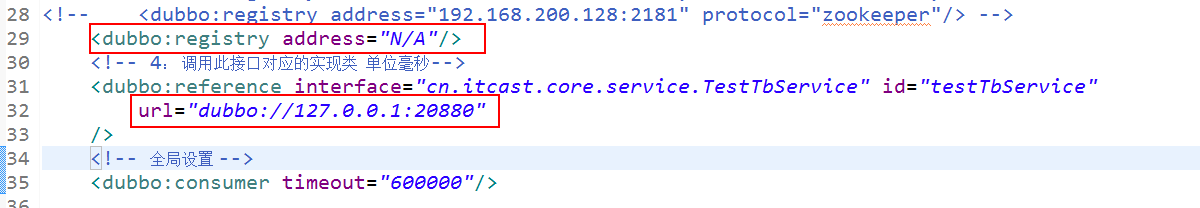


## 直接连接

服务提供方设置不注册自己地址



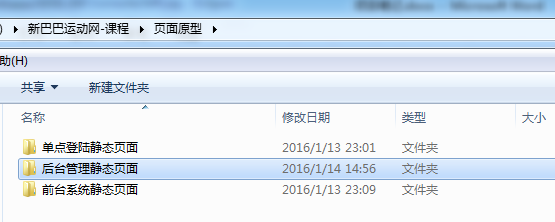
服务消费方

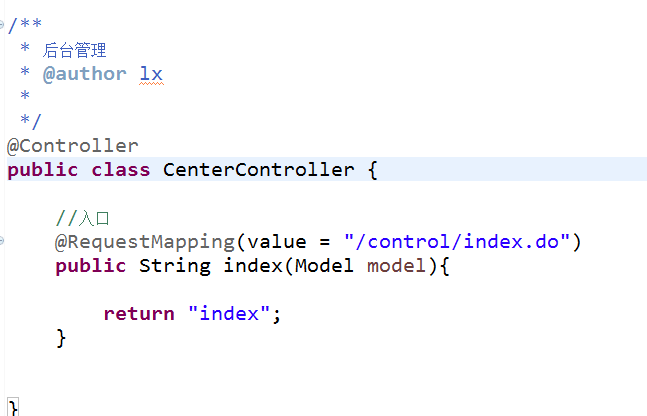


## 不检查



# 后台管理搭建







# 品牌管理

## 列表查询

准备工作

|  |
| --- |
| DROP TABLE IF EXISTS `bbs\_brand`;  CREATE TABLE `bbs\_brand` (  `id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT 'ID',  `name` varchar(40) NOT NULL COMMENT '名称',  `description` varchar(80) DEFAULT NULL COMMENT '描述',  `img\_url` varchar(80) DEFAULT NULL COMMENT '图片Url',  `web\_site` varchar(80) DEFAULT NULL COMMENT '品牌网址',  `sort` int(11) DEFAULT NULL COMMENT '排序:最大最排前',  `is\_display` tinyint(1) DEFAULT NULL COMMENT '是否可见 1:可见 0:不可见',  PRIMARY KEY (`id`)  ) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8 COMMENT='品牌'; |

**INSERT INTO bbs\_brand VALUES ('1', '依琦莲', null, null, null, '99', '1');**

**INSERT INTO bbs\_brand VALUES ('2', '凯速（KANSOON）', null, null, null, 98, '1');**

**INSERT INTO bbs\_brand VALUES ('3', '梵歌纳（vangona）', null, null, null, 97, '1');**

**INSERT INTO bbs\_brand VALUES ('4', '伊朵莲', null, null, null, 96, '1');**

**INSERT INTO bbs\_brand VALUES ('5', '菩媞', null, null, null, 95, '1');**

**INSERT INTO bbs\_brand VALUES ('6', '丹璐斯', null, null, null, 94, '1');**

**INSERT INTO bbs\_brand VALUES ('7', '喜悦瑜伽', null, null, null, 93, '1');**

**INSERT INTO bbs\_brand VALUES ('8', '皮尔（pieryoga）', null, null, null, 92, '1');**

**INSERT INTO bbs\_brand VALUES ('9', '路伊梵（LEFAN）', null, null, null, 91, '1');**

**INSERT INTO bbs\_brand VALUES ('10', '金啦啦', null, null, null, 90, '0');**

**INSERT INTO bbs\_brand VALUES ('11', '来尔瑜伽（LaiErYoGA）', null, null, null, 89, '1');**

**INSERT INTO bbs\_brand VALUES ('12', '艾米达（aimida）', null, null, null, 88, '1');**

**INSERT INTO bbs\_brand VALUES ('13', '斯泊恩', null, null, null, 87, '1');**

**INSERT INTO bbs\_brand VALUES ('14', '康愫雅KSUA', null, null, null, 86, '1');**

**INSERT INTO bbs\_brand VALUES ('15', '格宁', null, null, null, 85, '1');**

**INSERT INTO bbs\_brand VALUES ('16', 'E奈尔（Enaier）', null, null, null, 84, '1');**

**INSERT INTO bbs\_brand VALUES ('17', '哈他', null, null, null, 83, '1');**

**INSERT INTO bbs\_brand VALUES ('18', '伽美斯（Jamars）', null, null, null, 82, '1');**

**INSERT INTO bbs\_brand VALUES ('19', '瑜伽树（yogatree）', null, null, null, 81, '1');**

**INSERT INTO bbs\_brand VALUES ('20', '兰博伊人（lanboyiren）', null, null, null, 80, '1');**

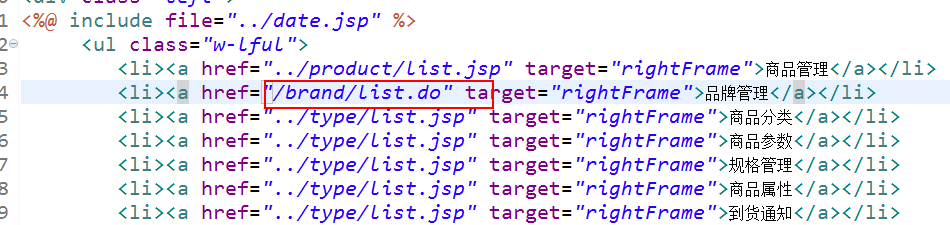
### Brand对象

|  |
| --- |
| /\*\*  \* 品牌  \* @author lx  \*  \*/  public class Brand implements Serializable{    /\*\*  \*  \*/  private static final long serialVersionUID = 1L;  //品牌ID bigint  private Long id;  //品牌名称  private String name;  //描述  private String description;  //图片URL  private String imgUrl;  //排序 越大越靠前  private Integer sort;  //是否可用 0 不可用 1 可用  private Integer isDisplay;//is\_display |

### 品牌管理

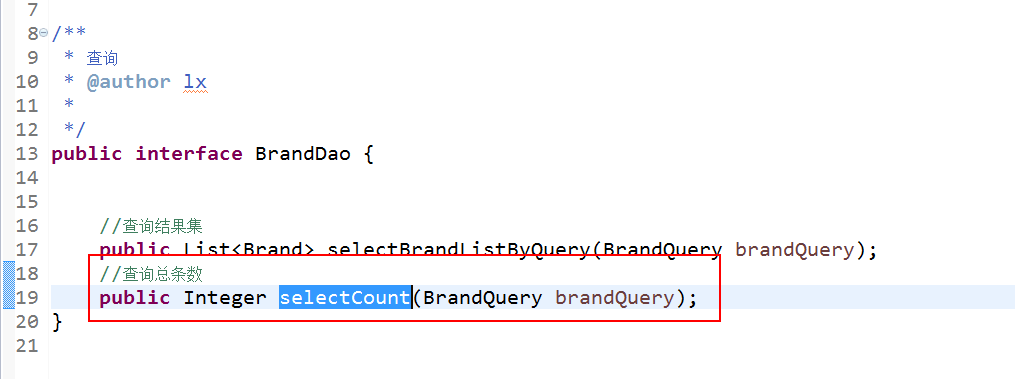
设置路径

/brand/list.do

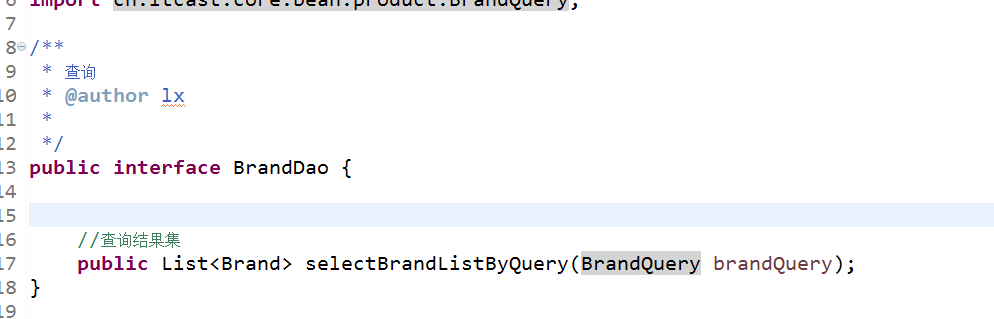


### Dao

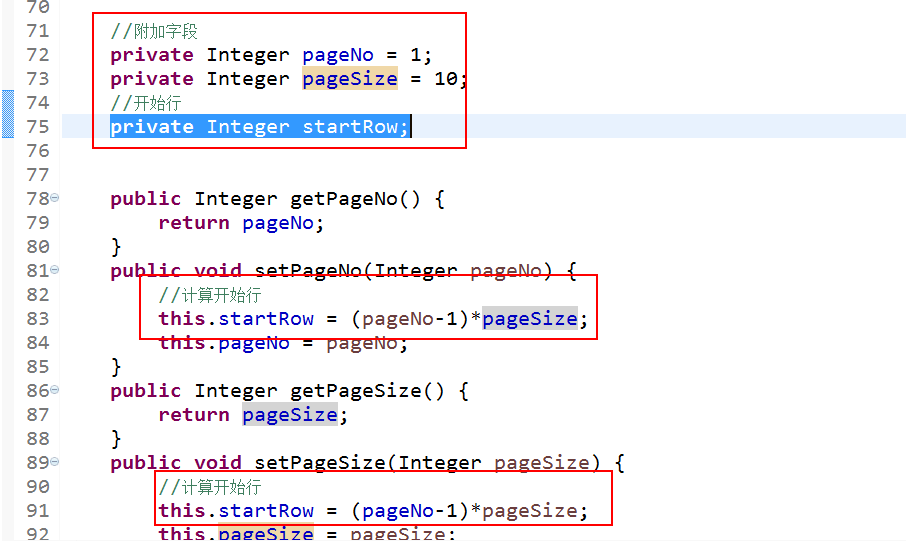
查询总条件



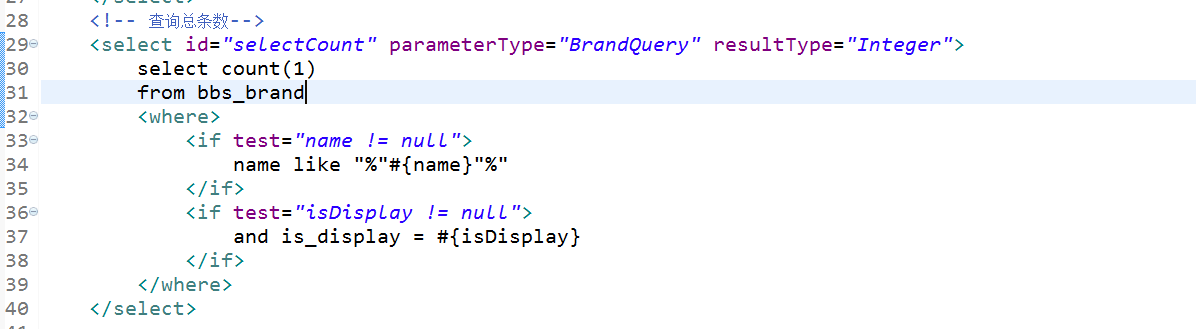
结果集

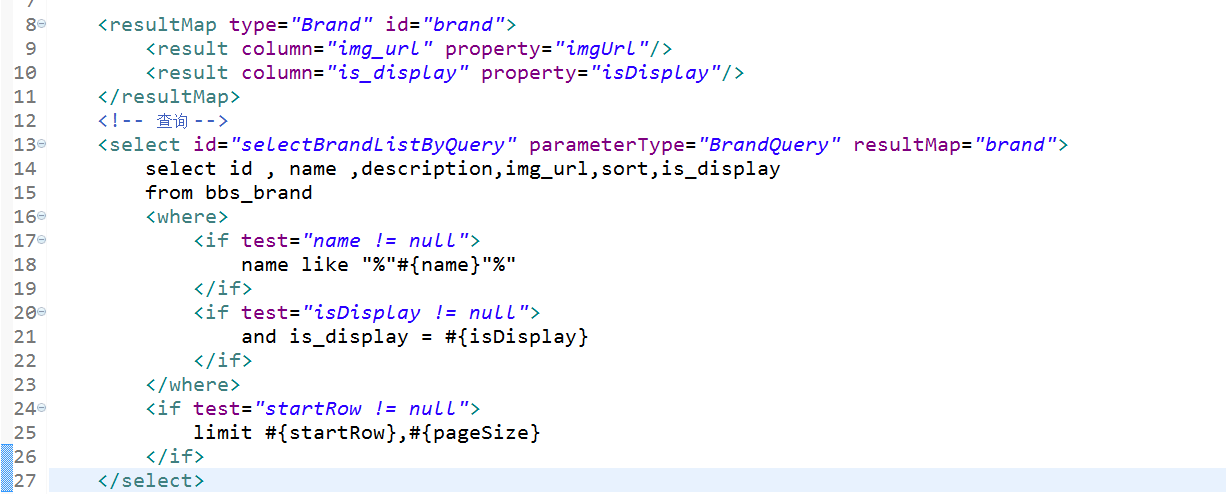


### brandQuery



### Mapper

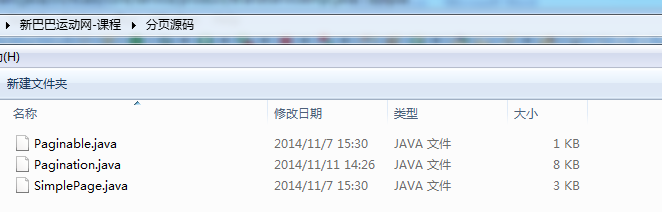




### Service

入参： 名称、是否可用、页号

返回值：分页对象（结果集、分页（当前页、每页数、总条数）

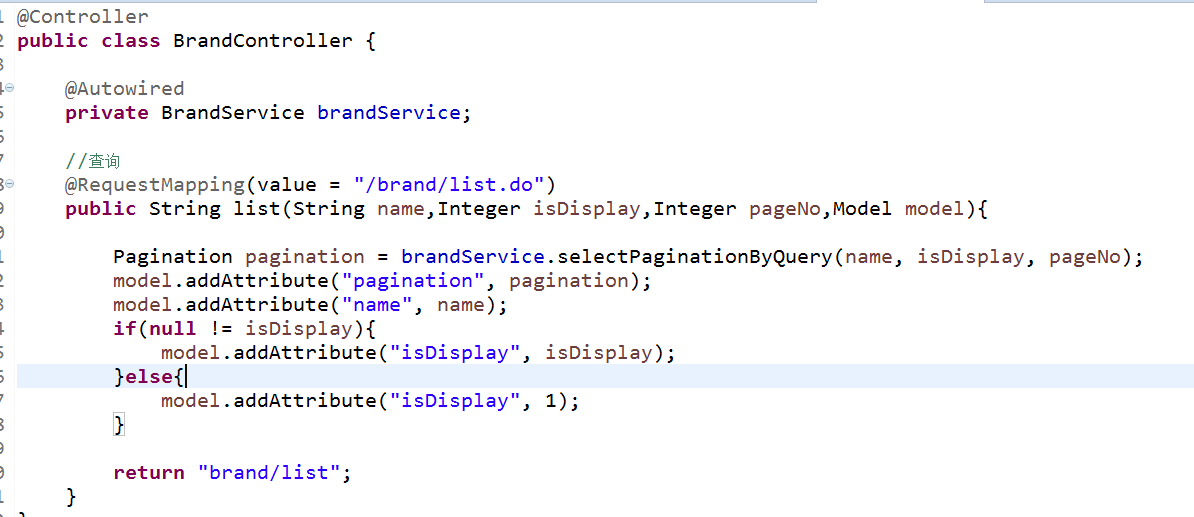


|  |
| --- |
| @Service("brandService")  @Transactional  public class BrandServiceImpl implements BrandService{  @Autowired  private BrandDao brandDao;  //查询分页对象  public Pagination selectPaginationByQuery(String name,Integer isDisplay,Integer pageNo){  BrandQuery brandQuery = new BrandQuery();  //当前页  brandQuery.setPageNo(Pagination.cpn(pageNo));  //每页数  brandQuery.setPageSize(3);    StringBuilder params = new StringBuilder();    //条件  if(null != name){  brandQuery.setName(name);  params.append("name=").append(name);  }  if(null != isDisplay){  brandQuery.setIsDisplay(isDisplay);  params.append("&isDisplay=").append(isDisplay);  }else{  brandQuery.setIsDisplay(1);  params.append("&isDisplay=").append(1);  }    Pagination pagination = new Pagination(  brandQuery.getPageNo(),  brandQuery.getPageSize(),  brandDao.selectCount(brandQuery)  );  //设置结果集  pagination.setList(brandDao.selectBrandListByQuery(brandQuery));  //分页展示  String url = "/brand/list.do";    pagination.pageView(url, params.toString());    return pagination;  } |

### controller

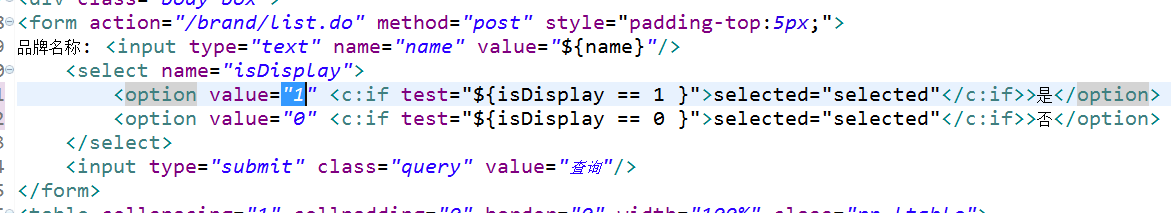
入参： 名称、是否可用、页号

返回值：分页对象（结果集、分页（当前页、每页数、总条数）



### 页面

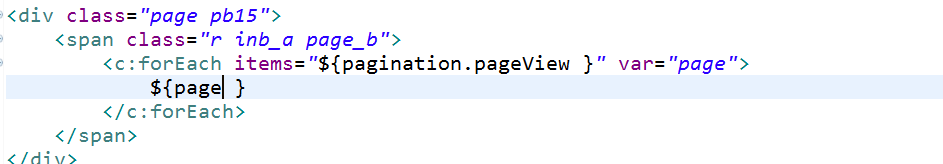
查询条件



显示数据

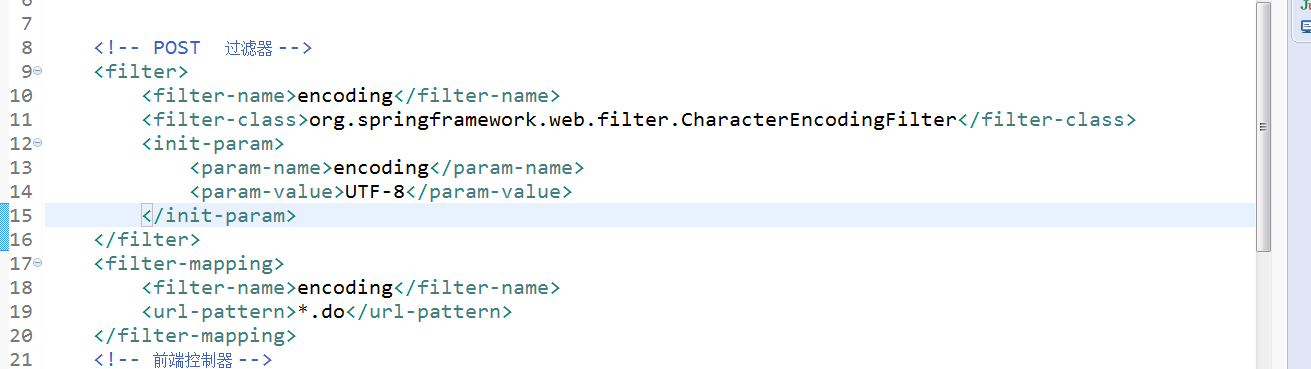


分页



### POST请求乱码

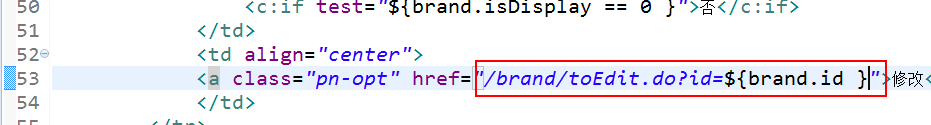
Web.xml



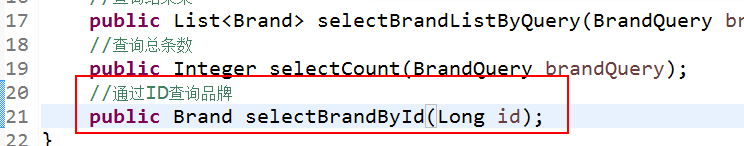
# 品牌管理

## 去修改页面

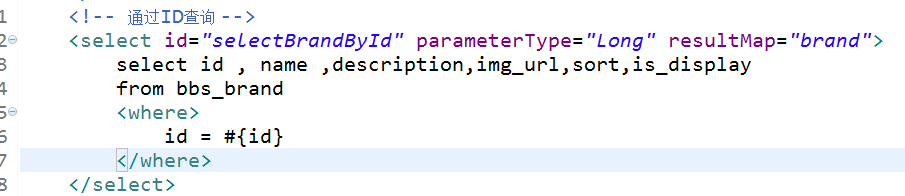
### 设置路径



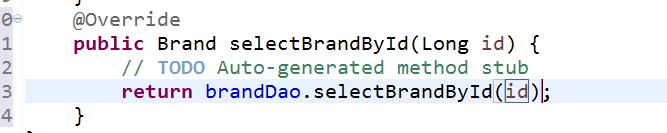
### Dao



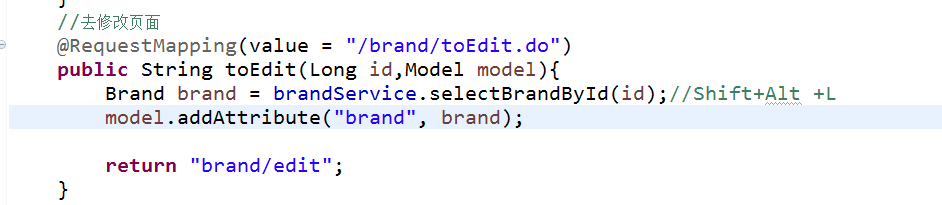
### Mapper



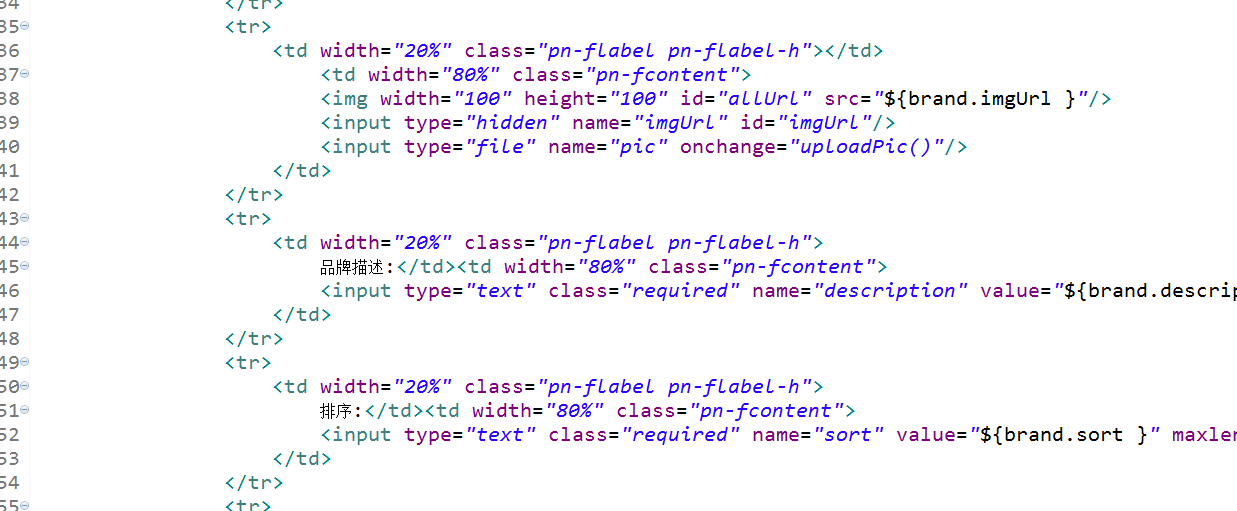
### Service



### Controller



### 页面回显

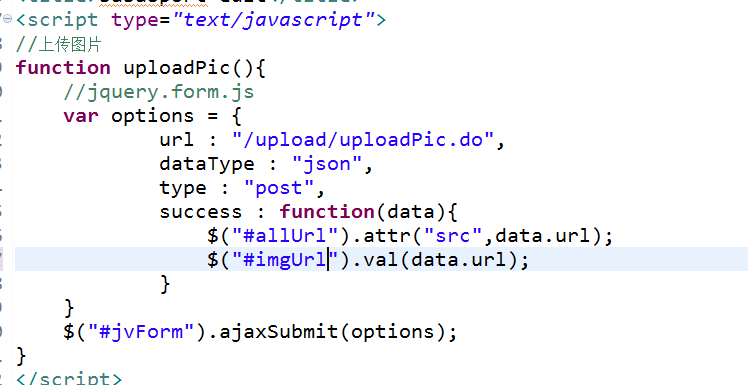


## 上传图片

### 事件

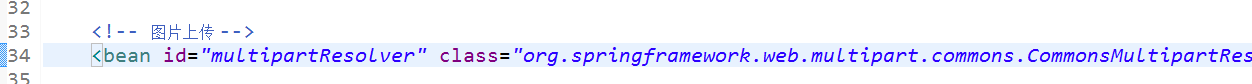


### Function

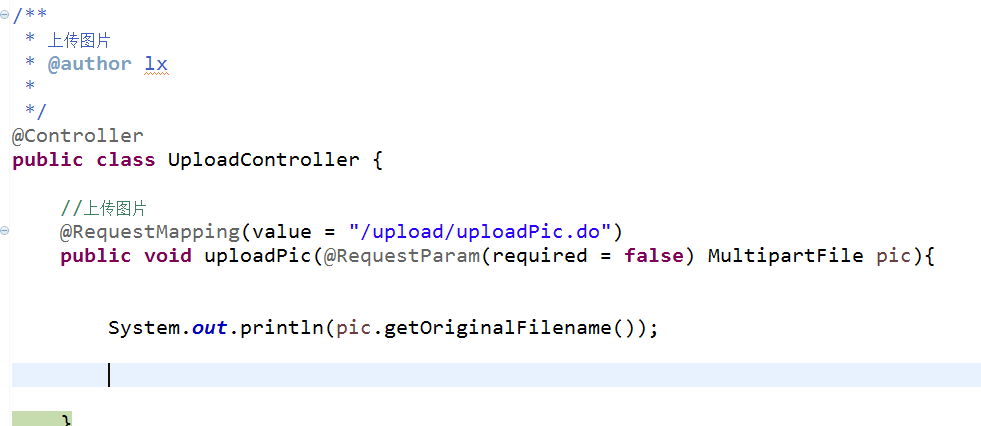


Ajax上传图片全过程

### Springmvc配置上传图片

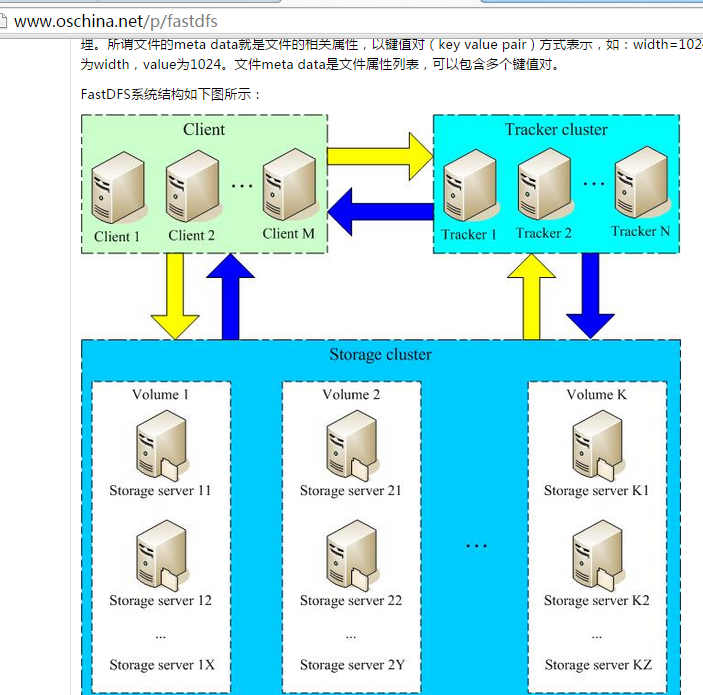


### 接收图片



### FastDFS原理

<http://www.oschina.net/p/fastdfs>



### FastDFS的Java接口使用

验证FastDFS启动成功

[http://192.168.200.128/group1/M00/00/01/wKjIgFWOYc6APpjAAAD-qk29i78248.jpg](http://自己机器的IP/group1/M00/00/01/wKjIgFWOYc6APpjAAAD-qk29i78248.jpg)

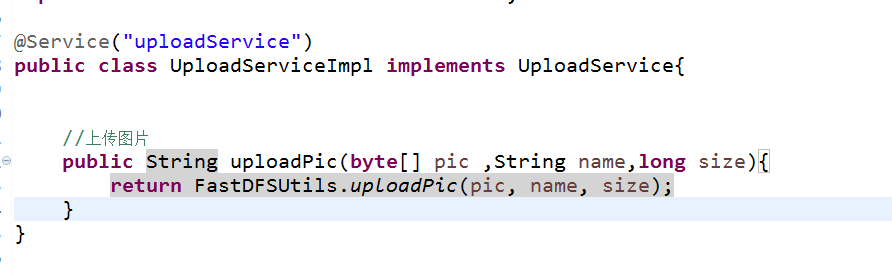
#### 全局配置文件

|  |
| --- |
| # connect timeout in seconds  # default value is 30s  connect\_timeout=30  # network timeout in seconds  # default value is 30s  network\_timeout=60  # the base path to store log files  base\_path=/home/fastdfs  # tracker\_server can ocur more than once, and tracker\_server format is  # "host:port", host can be hostname or ip address  tracker\_server=192.168.200.128:22122  #tracker\_server=192.168.101.4:22122  #standard log level as syslog, case insensitive, value list:  ### emerg for emergency  ### alert  ### crit for critical  ### error  ### warn for warning  ### notice  ### info  ### debug  log\_level=info  # if use connection pool  # default value is false  # since V4.05  use\_connection\_pool = false  # connections whose the idle time exceeds this time will be closed  # unit: second  # default value is 3600  # since V4.05  connection\_pool\_max\_idle\_time = 3600  # if load FastDFS parameters from tracker server  # since V4.05  # default value is false  load\_fdfs\_parameters\_from\_tracker=false  # if use storage ID instead of IP address  # same as tracker.conf  # valid only when load\_fdfs\_parameters\_from\_tracker is false  # default value is false  # since V4.05  use\_storage\_id = false  # specify storage ids filename, can use relative or absolute path  # same as tracker.conf  # valid only when load\_fdfs\_parameters\_from\_tracker is false  # since V4.05  storage\_ids\_filename = storage\_ids.conf  #HTTP settings  http.tracker\_server\_port=80  #use "#include" directive to include HTTP other settiongs  ##include http.conf |

#### FastDFSUtils

|  |
| --- |
| /\*\*  \* 上传图片到Fast  \* @author lx  \*  \*/  public class FastDFSUtils {  public static String uploadPic(byte[] pic ,String name,long size){  String path = null;  //ClientGloble 读配置文件  ClassPathResource resource = new ClassPathResource("fdfs\_client.conf");  try {  ClientGlobal.init(resource.getClassLoader().getResource("fdfs\_client.conf").getPath());  //老大客户端  TrackerClient trackerClient = new TrackerClient();  TrackerServer trackerServer = trackerClient.getConnection();  StorageServer storageServer = null;  StorageClient1 storageClient1 = new StorageClient1(trackerServer, storageServer);  String ext = FilenameUtils.getExtension(name);    NameValuePair[] meta\_list = new NameValuePair[3];  meta\_list[0] = new NameValuePair("fileName",name);  meta\_list[1] = new NameValuePair("fileExt",ext);  meta\_list[2] = new NameValuePair("fileSize",String.valueOf(size));      // group1/M00/00/01/wKjIgFWOYc6APpjAAAD-qk29i78248.jpg  path = storageClient1.upload\_file1(pic, ext, meta\_list);  } catch (Exception e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  return path;  }  } |

### 保存图片到FastDFS分布式文件系统



Controller

