# 增加菜单

## 2.1新增一级菜单

菜单管理—新增

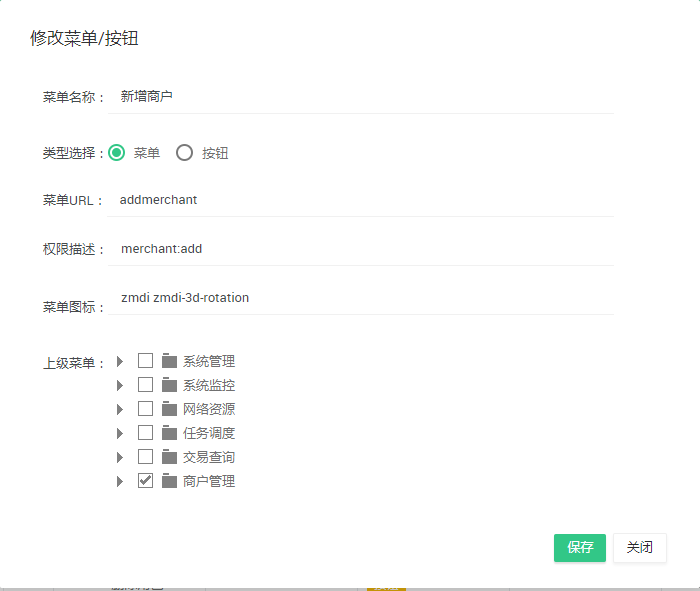
填写菜单名称，其他项目不用填写，点击保存。



## 2.2新增二级菜单

菜单管理—新增

填写菜单url，填写权限，勾选上级菜单，保存



# 导出配置

**增加一个交易，根据条件查询到所有要导出的数据，调用FileUtils.*createExcelByPOIKit（）进行导出。***

导出excel：

@RequestMapping("trans/excel")

@ResponseBody

**public** ResponseBo transExcel(Trans trans) {

**try** {

List<Trans> list = **this**.transService.findTransWithSel(trans);

**return** FileUtils.*createExcelByPOIKit*("交易明细表", list, Trans.**class**);

} **catch** (Exception e) {

e.printStackTrace();

**return** ResponseBo.*error*("导出Excel失败，请联系网站管理员！");

}

}

导出csv：

@RequestMapping("trans/csv")

@ResponseBody

**public** ResponseBo transCsv(Trans trans) {

**try** {

List<Trans> list = **this**.transService.findTransWithSel(trans);

**return** FileUtils.*createCsv*("交易明细表", list, Trans.**class**);

} **catch** (Exception e) {

e.printStackTrace();

**return** ResponseBo.*error*("导出Csv失败，请联系网站管理员！");

}

}

**对应的实体配置要导出的列**

在对应的实体列上添加注解@ExportConfig("平台流水")

此注解可进行格式化例如

时间：

@ExportConfig(value = "商户交易日期", convert = "c:cc.mrbird.common.util.poi.convert.TimeConvert")

常量：

@ExportConfig(value = "交易状态", convert = "s:00=成功,01=失败,07=处理中")

# Spring Boot项目打包成war包

在pom.xml文件中，将打包方式改为war：

|  |  |
| --- | --- |
| 1 | <packaging>war</packaging> |

然后添加如下的Tomcat依赖配置，覆盖Spring Boot自带的Tomcat依赖：

|  |  |
| --- | --- |
| 1  2  3  4  5 | <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-tomcat</artifactId>  <scope>provided</scope>  </dependency> |

在*<build></build>*标签内配置项目名（该配置类似于server.context-path=mrbird）：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ...  <build>  ...  <finalName>mrbird</finalName>  </build>  ... |

添加启动类ServletInitializer：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | import org.springframework.boot.builder.SpringApplicationBuilder;  import org.springframework.boot.web.support.SpringBootServletInitializer;  public class ServletInitializer extends SpringBootServletInitializer {  @Override  protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {  return application.sources(Application.class);  }  } |

其中Application为Spring Boot的启动类。

准备完毕后，运行*mvn clean package*命令即可在target目录下生产war包：

