大数据存储实践

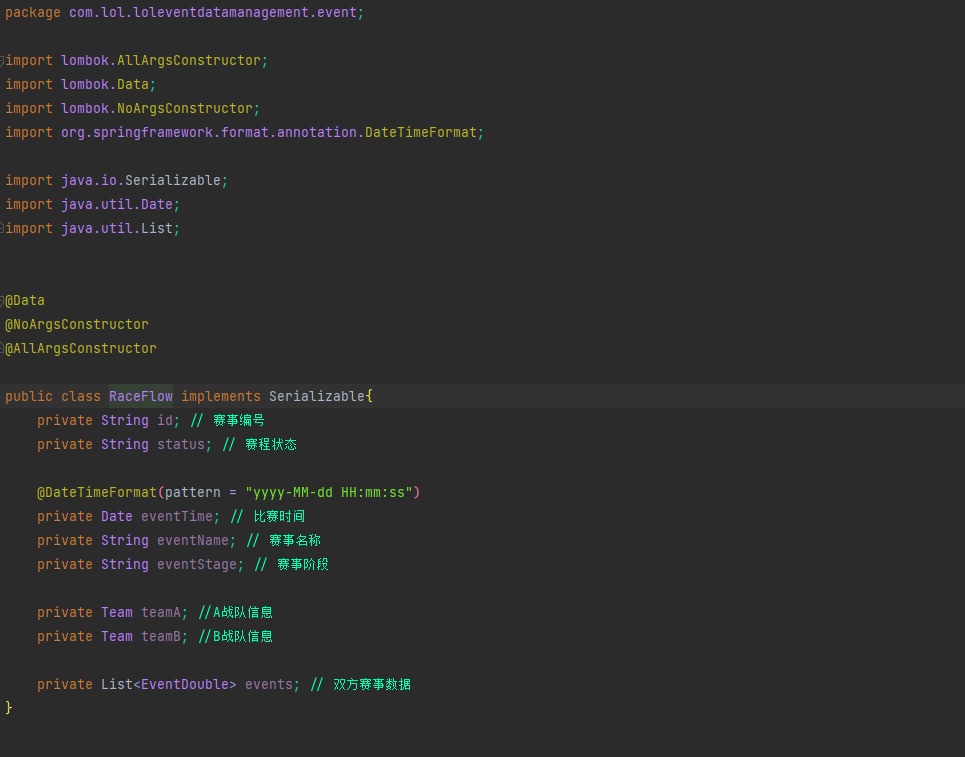
实习日志

班级：数管201 姓名：张今朝 学号：2020311228

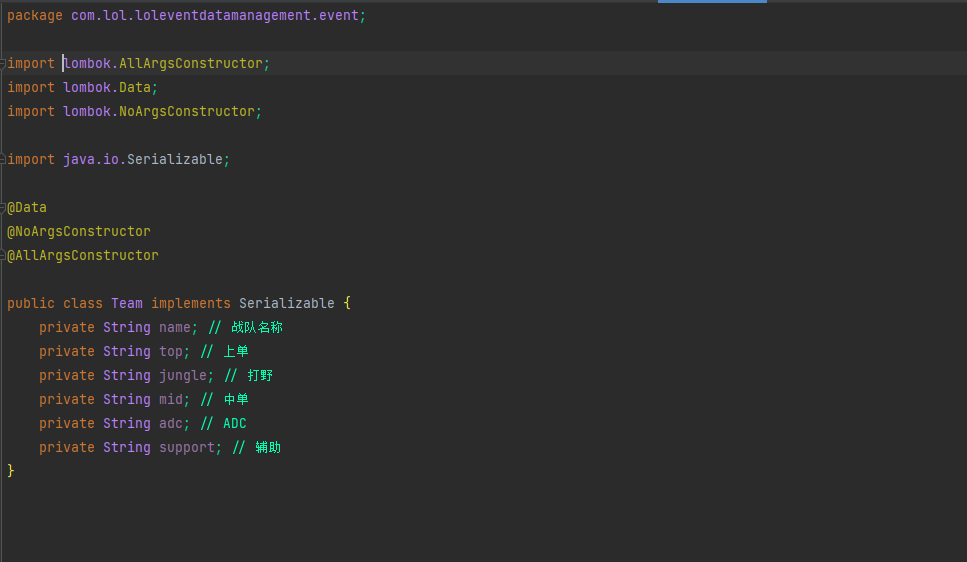
一、实习内容

1.实现战队预约比赛、赛事数据添加、赛事数据管理功能

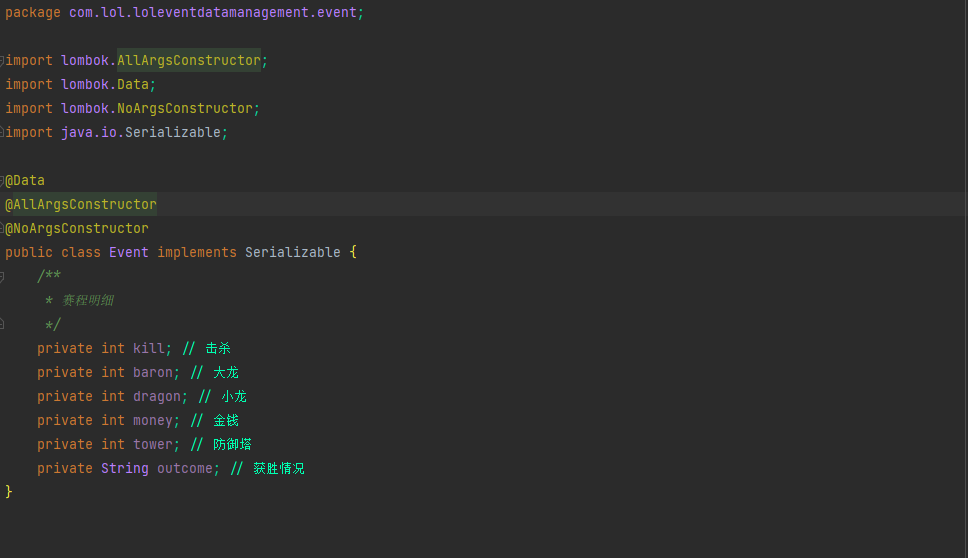
1.1RaceFlow类



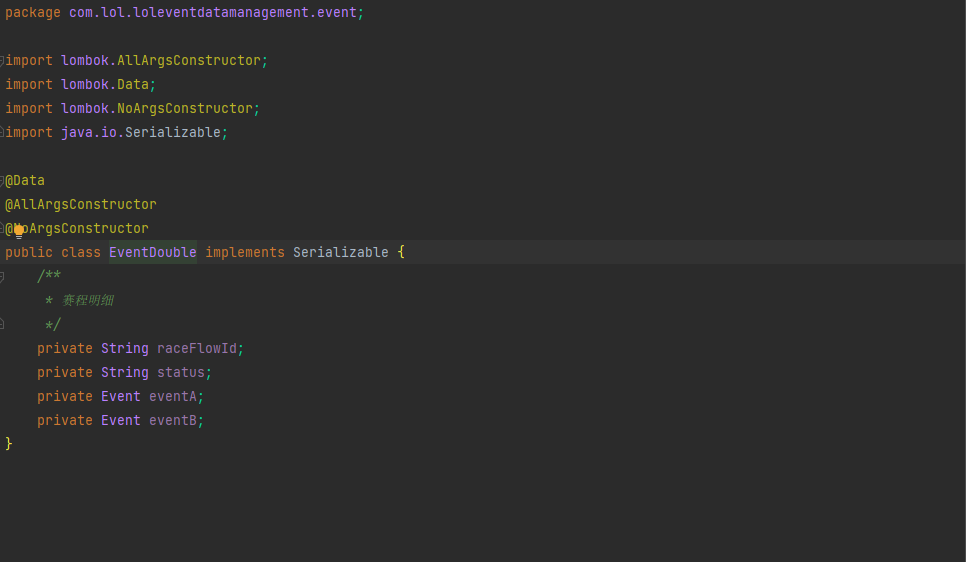
1.2Team类



1.3Event类



1.4EventDouble类



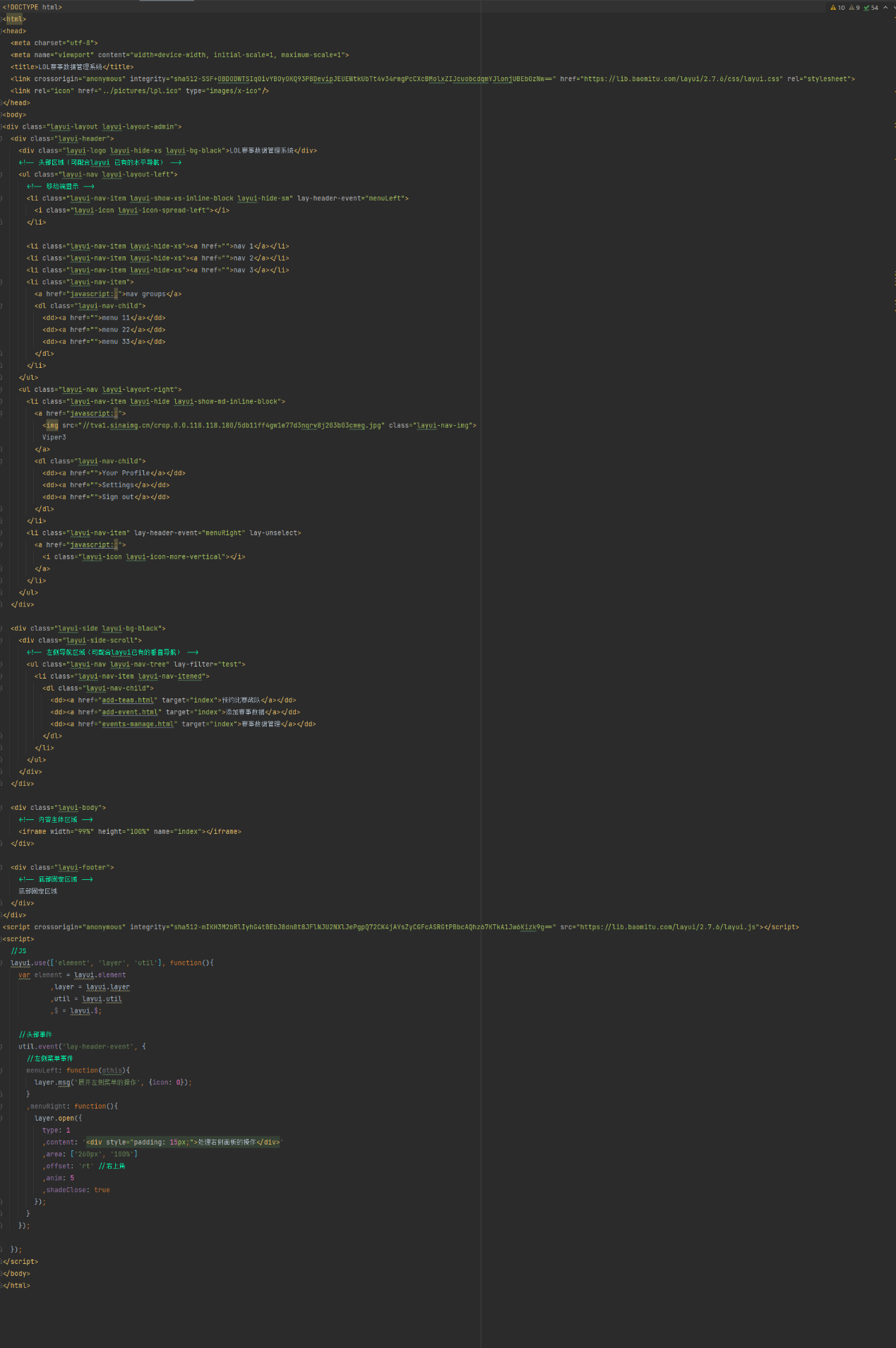
1.5EventController类



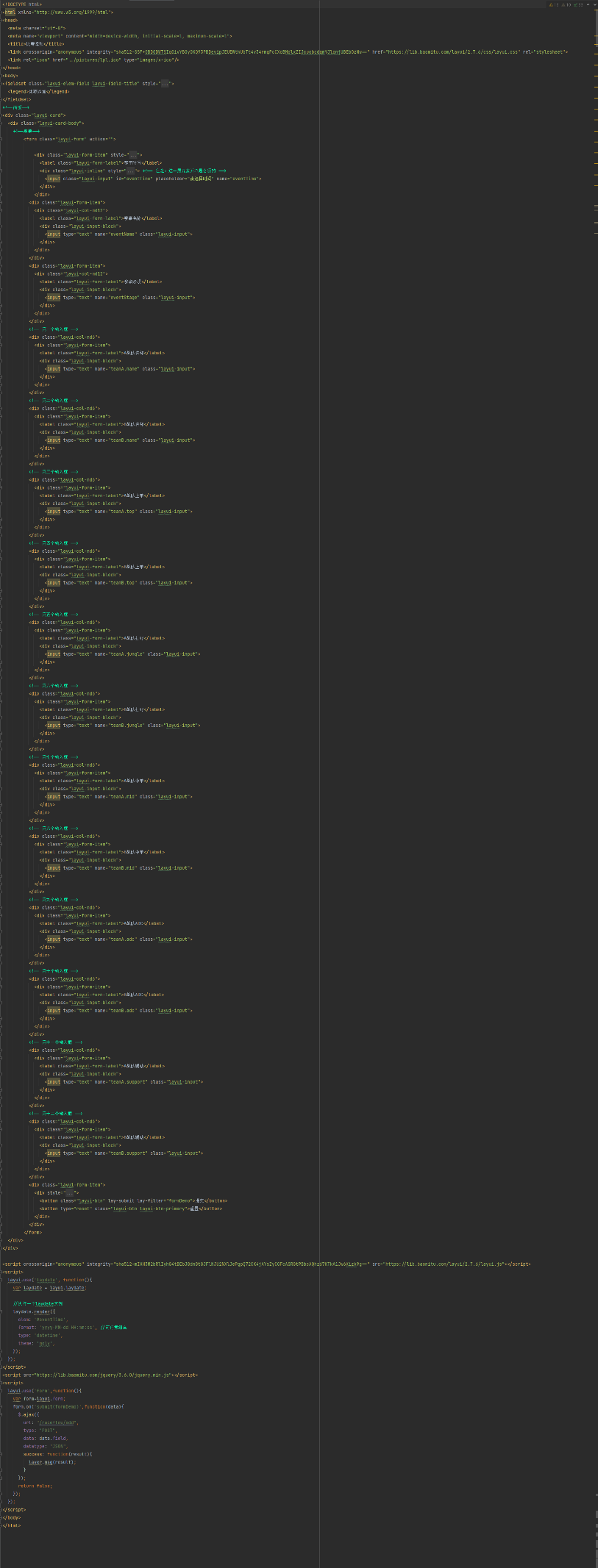
1.6EventService类



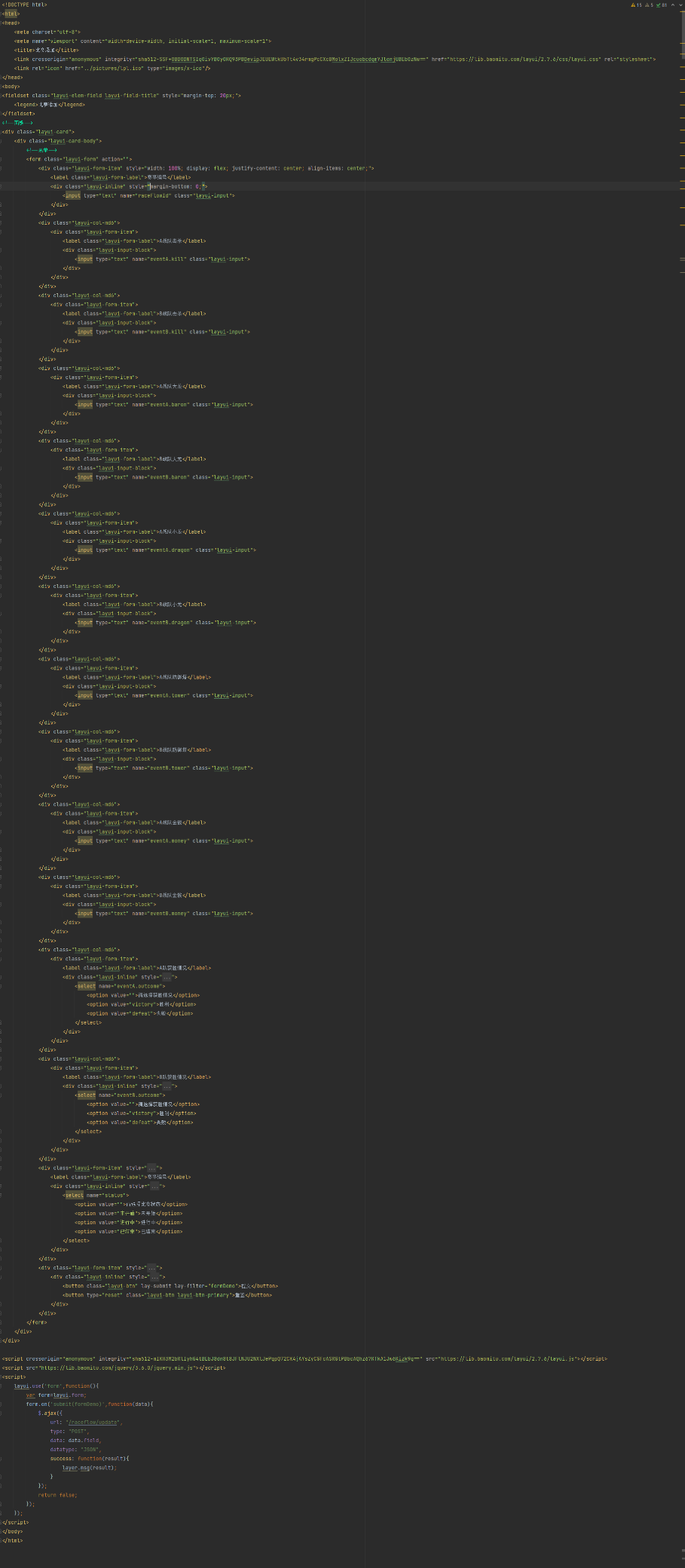
1.7index.html



1.8add-team.html



1.9add-event.html



二、收获与体会

1.创建基础类RaceFlow、Team、Event，EventDouble、RaceFlowVO为后续操作提供初始化的对象。

2.1RaceFlow类

1. private String id; // 赛事编号
2. private String status; // 赛程状态
3. private Date eventTime; // 比赛时间
4. private String eventName; // 赛事名称
5. private String eventStage; // 赛事阶段
6. private Team teamA; //A战队信息
7. private Team teamB; //B战队信息
8. private List<EventDouble> events; // 双方赛事数据

2.2Team类

1. private String name; // 战队名称
2. private String top; // 上单
3. private String jungle; // 打野
4. private String mid; // 中单
5. private String adc; // ADC
6. private String support; // 辅助

2.3Event类

1. private int kill; // 击杀
2. private int baron; // 大龙
3. private int dragon; // 小龙
4. private int money; // 金钱
5. private int tower; // 防御塔
6. private String outcome; // 获胜情况

2.4EventDouble类

private String raceFlowId; // 区别RaceFlow类中的id

private String status; // 用以更新赛程状态

private Event eventA; // 用以存储events列表中eventA字段

private Event eventB; // 用以存储events列表中eventB字段

2.5RaceFlowVO类

private String id; // 赛事编号

private String status; // 赛程状态

private Date eventTime; // 比赛时间

private String eventName; // 赛事名称

private String eventStage; // 赛事阶段

private String teamAName; //提取嵌入式文档的内层属性用以页面展示

private String teamBName; //提取嵌入式文档的内层属性用以页面展示

2.6.创建扩展类EventService，该类实现了将赛事信息写入MongoDB。

// 比赛编号根据雪花算法生成

raceFlow.setId(IdUtil.getSnowflake(1, 1).nextIdStr());

// 设置比赛状态

raceFlow.setStatus("未开始");

// 添加预约比赛至MongoDB

mongoTemplate.insert(raceFlow, "lol");

2.7.创建扩展类OrderController，该类使用了Resource、PostMapping接口，为赛事管理的控制层，负责实现添加赛事数据时传递参数。

1. 设置RequestMapping名为“raceflow”
2. 设置PostMapping名为“add”
3. 设置“add”操作为执行EventService中的addRaceFlow方法。

2.8.在index.html中通过设置左侧边栏a标签的target属性和url属性，target值为内容主体区域的iframe标签的name值。即可通过点击侧边栏的按钮，在内容主体区域中显示侧边栏按钮a标签中的url地址中的网页内容出来。

2.9.在预约比赛战队页面中，设置提交按钮的“lay-submit lay-filter”属性，值为formDemo。Script标签中有对应名的按钮处理程序form.on('submit(formDemo)',function(data)。点击提交按钮时，将表单中的信息传递给JavaScript处理，其中包含url、type、data、datatype属性，JavaScript程序又将表单提交的值传递至url路径中的程序处理，提交方式为 “POST”，数据为表单中传递的值。数据类型为“JSON”。

因此，对应至本项目，数据传递至EventController类中的addRaceFlow方法处理。EventController类中的addRaceFlow方法又调用对象eventService的addRaceFlow方法处理。随后将传递至EventService类对象eventService的参数赋值给形参RaceFlow类的对象raceFlow，并调用addRaceFlow方法，将订单信息写入MongoDB中。执行完毕后EventController类中的addRaceFlow方法返回“成功添加赛事流程，比赛即将开始---”。add-team页面收到返回信息之后，通过屏幕快闪的方式显示该信息。