系统/子系统设计（结构设计）说明（后台部分）

Tomato

2018 年 1 月

1 引言

1.1 系统概述

后台分为两个部分，上层是 Controller，是用来处理业务逻辑的，下层是数据库，给上层的 Controller 提供相应的服务。下面会分两块来具体介绍这两部分。详见《接口设计说明》和《数据库（顶层）设计说明》。

1.2 文档概述

本文档将围绕 Spring Boot 架构来介绍后台部分。

2 系统级设计决策

本系统后台主要使用 Spring Boot 架构，它支持在开启服务的时候，接受 HttpRequest 或 websocket 并进行后台处理，然后返回相应的结果。系统接受的输入是 HttpRequest 或 websocket, 详见《接口设计说明》, 对于每一个输入的响应, 被 Controller 定义, 不同的 request 会有不同的 Controller 对其进行处理并返回相应的结果。系统处理的过程中，依赖于两部分的内容：用户的 request 中的具体内容和底层数据库已有的信息。详见《数据库（顶层设计说明》。该系统的数据库存在服务器上，服务器的管理人员可以通过 database asdan 来访问数据库并查看当前数据库中的所有信息。该系统对服务器的要求是要求服务器能通过 maven 运行项目，并配有 mysql 数据库。

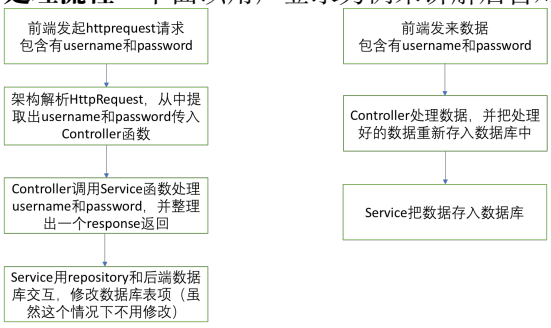
3 系统体系结构设计

3.1 系统总体设计

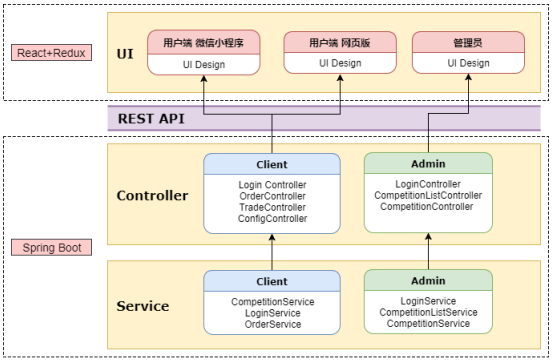
概述 本系统要实现的功能详见《prj9ASDAN 模拟商业竞拍交易大赛系统》。本系统要实现的性能：响应较快、支持并发、有一定的安全性、可在各个平台上运行、可移植等。本系统支持在 Windows、Unix、Linux 环境下运行。

设计思想 本系统顶层核心要处理的两个问题是对于 HttpRequest 的处理和对于 websocket 的处理，而底层核心要处理的问题是数据库的维护等问题。如果直接从头开始开发一套框架，则要实现的内容可能比我们当前写的多得多。所以我们后台根据这个需求找到了 Spring Boot 框架。Spring Boot 很好地通过了 Annotations 来实现我们需要的对于前端 web 端的响应，也实现了我们需要的和数据库的连接、并发的响应等等。本系统并不需要很高深的算法，但是对逻辑上的要求十分的多和细致。

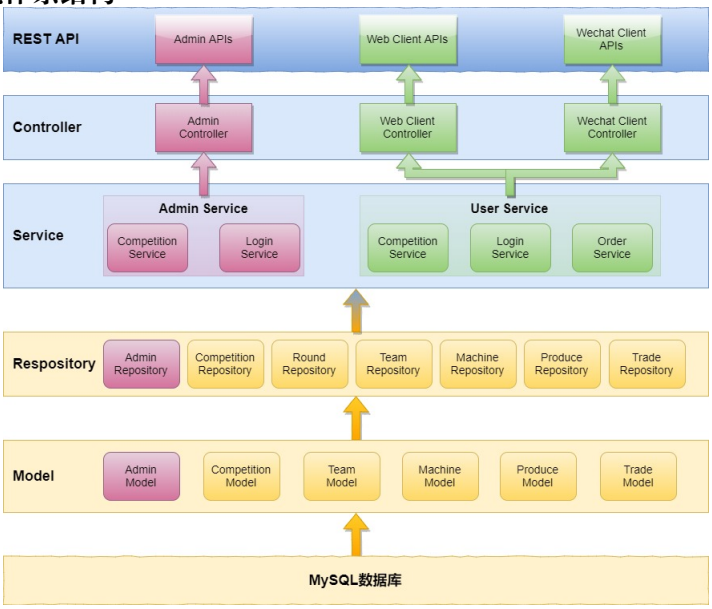
基本处理流程 下面以用户登录为例来讲解后台对于 HttpRequest 的处理流程。



当前端发来 HttpRequest 或 websocket 的时候，Controller 首先会提取出发来的数据内容，然后将根据业务逻辑处理数据，在处理数据的过程中，可能需要访问数据库中已有的数据，如在这个情况下需要对拍数据库中的账号和密码。也有可能需要修改后台数据库，但是图中的例子不需要。在处理完数据之后，Controller 会把数据重新包装，然后根据 API 文档的说明传回前端。整个数据处理的过程可以理解为从前端发来数据，然后 Controller 向数据库 query 数据，然后处理完数据之后又 save 到数据库中，最后返回给前端。

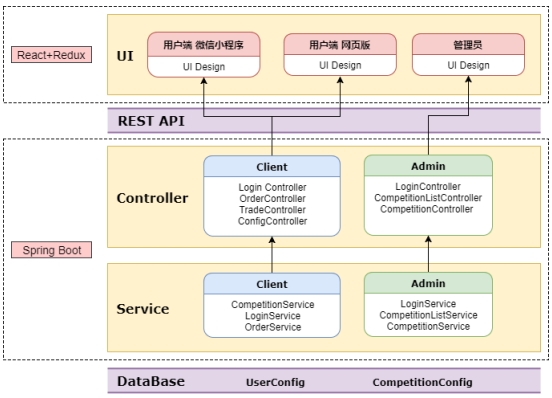


系统体系结构



如图，前端的功能是响应用户的操作，并把用户操作的数据根据 API 文档中的要求进行包装，作为 HttpRequest 或 websocket 发往后端，后端 Controller 对于包装好的数据解码，并 query Service，和数据库交互，完成数据的处理，并完成数据库的更新。最后将 HttpRequest 返回给前端或将 websocket 转发。

3.2 接口设计



4 系统出错处理设计

系统出错后，后台数据库的内容不会被清空，并且我们是在处理 request 流程中对数据库是实时更新的，宕机之后数据库的内容还是会保留，当下次启动服务器的时候它还能从原数据库中获得原来的数据，并根据这些数据进行处理。

当系统出错后，建议重启服务器。

5 尚待解决的问题

微信后端的 websocket 的测试问题。