1. 后端

后端代码存放于根目录下，启动文件是server.js，server.js定义了服务端的socket，并开启了各类事件监听器。其中client.js、test.js、testcase.js是早期使用的测试文件，除参考价值外无实际意义。

后端除server.js外，还包括两个文件夹，util和proxy。Util中存放通用功能类js文件，其中dbutil.js最为重要，封装了所有数据库的操作函数，提供外部数据接口。lolutil中封装了多个与lol美服api相关的功能函数，需要注意的是，访问lol美服API需要官方提供的key，而目前使用的开发者key两天过期一次，如果key过期的情况下访问这些读取API的功能函数会导致程序崩溃。Lolcfg中存放英雄的静态数据，dbcfg中存放默认的数据库连接配置。Logutil对console.log进行了简单的封装。

Proxy中存放服务端全局变量和各类监听器的实际实现。UserProxy.js中存放所有与用户信息相关的监听器实现，TeamProxy.js中存放所有与房间、队伍有关的监听器的实现，battleProxy.js中存放所有与对战相关的监听器的实现，socketProxy.js中存放所有与客户端Socket有关的操作的实现。FeedBackProxy.js目前已经废弃，存有一些历史版本的监听器。

1. 前端

前端主要是js文件和html文件，html文件分为两类，一类是带有Swig脚本的模板页面，这类html页面直接加载时为空页面，只有传入所需的json对象时才会正常渲染。另外一类页面则是普通的html页面。

前端主要的js文件有：

Bind\_lol.js：实现英雄联盟账号绑定按钮的单击事件响应函数。

Create\_room.js：实现房间创建按钮的单击事件响应函数。

Game\_history\_query.js：实现英雄联盟官方战绩查询功能。

Invite\_battle.js：实现自主约战的对战邀请功能。

Invite\_friend.js：实现在房间中邀请好友加入房间的功能。

Login.js：实现登录和注册功能。

Lolcfg.js：存放英雄联盟英雄信息的静态数据，每当英雄联盟推出新英雄需更新此文件。

Lolutil.js：存放读取LOL美服官方API的函数，包括通过名字查询玩家账号ID，通过名字、时间查询玩家一段时间比赛的详细信息等。

Message\_center.js：消息中心的实现代码。

Message\_operation.js：消息操作的实现代码。消息的操作主要有两种，接受和拒绝。

Personal\_center.js：提供个人中心的数据展示功能。

Refresh\_formed\_room.js：刷新对战大厅中的房间。

Request\_rank\_list.js：实现排行榜查看功能。

Router.js：过时的文件，里面存放了一些导航栏按钮的单击事件响应函数。

Socket\_util.js：前端最重要的js文件，存放了客户端各类事件监听器和渲染界面的方法。

Swig.js：提供用于渲染界面的loadSwigView函数，该函数极其重要，共需要两个参数，第一个参数是任意html页面，第二个参数是一个json对象。该函数会将json对象中的数据加载到swig模板页的swig脚本中，函数的返回值是一个字符串，字符串的内容是填有数据的swig模板页的html代码。将该字符串，用$(‘#main-view’).html(returnedHTMLString)，即可完成页面的渲染。

其他js文件的说明请参考HudsonJoe的youtube主页。

前端的HTML文件经历过两次大的版本变更，目前实际开发中只需要使用以swig为前缀的页面即可，以spa为前缀或无前缀的html页面仅用于参考swig页面渲染完成后应有的样式和布局。

主要的HTML文件有：

Swig\_battle.html：对战大厅的swig页面，显示所有已创建的队伍。

Swig\_battleres.html：对战结果的swig页面，用于显示一场对局的结果。

Swig\_dataquery.html：英雄联盟官方战绩查询页面。

Swig\_fight.html：对战双方在对战过程中的swig页面。

Swig\_friend\_list\_head.html：好友列表上半部分的swig页面（显示当前用户的名字和在线状态）。

Swig\_friend\_list.html：好友列表的下半部分的swig页面（显示所有好友的名字和在线状态）。

Swig\_home.html：主页面。所有动态加载的页面都加载到id为main-view的div标签中。登录、注册、创建房间等模态窗口的代码也在该页面中。

Swig\_matches.html：英雄联盟官方数据查询结果的左边栏，显示玩家进行了哪些场比赛，每场使用了什么英雄以及KDA等数据。

Swig\_match\_detail.html：英雄联盟官方数据查询结果的右边栏，显示玩家每场比赛的详细数据，包括双方名字，使用的英雄，kda，伤害数值，承受伤害数值以及出装等数据。

Swig\_menu.html：系统菜单的HTML页面。

Swig\_messages.html：消息中心的Swig页面。

Swig\_myroom\_info.html：房间的基础信息Swig页面。

Swig\_myroom\_teamate.html：房间已加入成员的Swig页面。

Swig\_myroom\_frame.html：房间的框架页面，Swig\_myroom\_info.html和Swig\_myroom\_teamate.html这两个模板页面都需要加载到此页面中，再将此页面加载到Swig\_home的main-view中即可正确显示房间。

Swig\_pagination.html：提供分页页面。

Swig\_personal\_basic.html：个人中心的Swig页面。

Swig\_queryres.html：英雄联盟官方数据查询结果的页面，Swig\_matches.html和Swig\_match\_details.html是加载在该页面上的。

Swig\_rank.html：排行版Swig页面。

Swig\_team\_detail.html：对战大厅查看某一支队伍详情的详情页。

Swig\_tournament.html：赛事Swig页面。

1. 前后端对接方法

首先在前端代码中添加一个js文件，使用jquery添加按钮的click响应函数，在click的回调函数中，通过jquery获取用户输入到文本框、单选框等输入框中的信息，组织成json串，调用socket.emit方法，发送给服务端。服务端在相应的proxy.js中添加对应消息的监听器，并实现，在sever.js中调用已经实现的监听器。在监听器中，获取到客户端传来的数据，进行处理和进一步操作（读数据库）等，之后将处理的结果组织成json串，通过socket.emit发送给客户端。客户端在socket\_util.js中添加对服务端返回的数据包的监听器，并在监听器的回调函数中，读取服务端传回的数据，根据数据内容渲染界面。渲染界面的方法是：调用var htmlString = douniu.loadSwigView方法，第一个参数是一个swig模板html页面，第二个参数是传入swig模板页的json对象，返回值是一个字符串，字符串内容是swig模板加载数据后得到的html代码。将该html代码字符串，用$(‘#view’).html(htmlString)方法，把该字符串添加到前端页面预留的某个div标签中，即可完成页面的渲染。

1. 其他说明

前端页面目前迭代到第三版，使用swig模板和matrialize前端框架。实际开发过程中，只需关心前缀是swig的html文件即可，如swig\_home.html。

服务端返回给客户端的数据包，包头名为feedback，feedback中共四个值，errorCode、type、text、extension。其中errorCode中存放错误码，0为结果正常，其他为结果异常。Text中存放提示信息，type是feedback的数据包类型，extension中存放数据。

服务端发送给客户端的数据包，如果包头是message，则该数据包代表用户‘消息中心’中的消息，该类型的数据包必须有一个名为messageType的值，用于区别消息类型。其他值可自定义，但是需要与swig\_messages.html中的swig脚本相对应。

1. 全局变量说明

服务端全局变量分散在各个proxy.js中，可通过每个proxy文件中的init方法，得知该proxy文件提供了哪些全局变量。客户端的全局变量存放在socket\_util.js中，分别为socket、userInfo、teamInfo、roomInfo、battleInfo、versusLobbyInfo、messageInfo。

Socket为服务端socket对象，用于向服务器发数据。

UserInfo在用户登录之后有值，存放用户的基本信息和好友列表。

RoomInfo在用户创建或加入房间后有值，存放用户所在的房间信息。

TeamInfo在用户所在队伍已发布的时候有值，存放用户所在的队伍信息。

BattleInfo在对战已建立时有值，存放对战双方的队伍信息。

MessageInfo是个数组，存放消息中心中的所有message。