Current: Q8

1.进入LoginActivity

2.读取本地asset中的loading

3.显示loading/login.html页面作为登陆界面

4.输入用户名和密码，点击登陆（显示数据加载中）

与服务器交互：

String info = ApiHelper.authentication(mContext, username, URLs.MD5(password));

使用http post请求：

1. 显示服务器验证的具体链接

Post http://121.199.38.185:4567

1. 向服务器推送的参数

发送用户名密码信息至http://121.199.38.185:4567/api/v1/android/username/password/authentication

红色的是用户输入的用户名和密码

Device：{

"os": "Samsung Galaxy S5 - 4.4.4 - API 19 - 1080x1920",

"platform": "android",

"uuid": "772f48dd8599a5b0",

"os\_version": "4.4.4",

"name": "Samsung Galaxy S5 - 4.4.4 - API 19 - 1080x1920"

}

1. 服务器响应的信息

responseJSON: {

"group\_name": "大区(全部)商行(全部)",

"role\_id": 1,

"group\_id": 165,

"code": 200,

"app\_ids": [2, 3, 4 ],

"analyse\_ids": [1, 2, 3, 4, 8, 10, 11, 12],

"device\_state": true,

"kpi\_ids": [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 18, 17],

"user\_name": "张磊",

"user\_device\_id": 355,

"user\_num": "test6",

"device\_uuid": "772f48dd8599a5b0",

"user\_id": 3020,

"role\_name": "测试人员",

"loading\_md5": "c10780bc76c8971d78ab5244b65b4b77",

"assets\_md5": "7e58d787137b055d5ec877134396c7ab"

"assets": {

"stylesheets\_md5": "1d10ad9a61e9cd0662a298610680bd26",

"javascripts\_md5": "9287afcbd064e2ac30378d0847cf3fad",

"fonts\_md5": "5901960c857600316c3d141401c3af08",

"images\_md5": "1afa5893736851156a3612b34c1d88a6"

},

}

1. 服务器响应什么，才认为验证成功

若状态码为200则验证成功

1）验证用户名和密码，服务器返回状态码：

400：网络未连接

401：用户名或密码不正确

408：连接超时

200：登陆成功，返回设备相关信息

1）用户名或密码为空，显示请输入用户名或密码

2）用户名或密码输入错误，显示用户名或密码错误



1. 进入MainActivity，并把LoginActivity finish掉

1）检测服务器静态资源是否更新，用本地资源MD5与服务器资源MD5值匹配，下载MD5值不同的资源

2）加载mainActivity的界面，分为仪表盘、分析、应用、消息、设置

服务器的界面路径：

1. 仪表盘的界面路径?

http://121.199.38.185:4567/mobile/role/1/group/165/kpi

1. 分析的界面路径?

http://121.199.38.185:4567/mobile/role/1/analyse

1. 应用的界面路径?

http://121.199.38.185:4567/mobile/role/1/app

1. 消息的界面路径?

<http://121.199.38.185:4567/mobile/role/1/group/165/user/3020/message>

Q4: 上述服务器链接是如何在应用中展示的？（上面四个随便拿一个）

运行mRunnableForDetecting线程，与服务器交互，若连接服务器成功，返回状态码200，得到get信息，进一步得到其中的body信息。

Q5: mRunnableForDetecting作用是什么？会出现几种结果？

监测与服务器的连接状态

200、304：运行mRunnableWithAPI线程，重新从服务器get需要的资源

400、408：展示没有网络时的webview

401：展示该设备被禁止使用的webview

Body为html网页，webview控件加载body信息中的html。

同时调用js的pageLink方法，跳转至subjectActivity，把bannerName、link、objectID、objectType信息交给subjectActivity处理。

Q5: JS与Android 交互使用到的知识点？

html中通过js调用java代码：

webview设置一个和js交互的接口

mWebView.addJavascriptInterface(new JavaScriptInterface(), "AndroidJSBridge");

android调用js：

mWebView.loadUrl(urlStringForLoading);



分别点击仪表盘、分析、应用中的报表，跳转至SubjectActivity ，

服务器的界面路径：

1. 仪表盘中点击某个报表的界面路径?（随便一个）

Q4: 报表链接是如何在应用展示的？

点击某个报表后，进入SubjectActivity，加载从与服务器交互，若连接服务器成功，返回状态码200，得到get信息，进一步得到其中的body信息。

Body为html网页，webview控件加载body信息中的html。

调用APiHelper的reportData方法获取报表网页的数据。

urlString：http://121.199.38.185:4567/api/v1/group/groupID/report/reportID/attachment

filename： template\_data\_group\_ groupID \_report\_ reportID.js

filePath：assetsPath/assets/javascripts/fileName

Q4：点击仪表盘后，有报表界面展示，这一过程是如何实现的？

加载本地的资源库：<file:////data/data/com.intfocus.yh_android/files/User-3020/HTML/_mobile_role_1_group_165_kpi.html>

Q5: 服务器端的HTML写入本地文件经过了那么操作？

调用FileUtil的writeFile方法，将HTML作为字符串写入本地文件

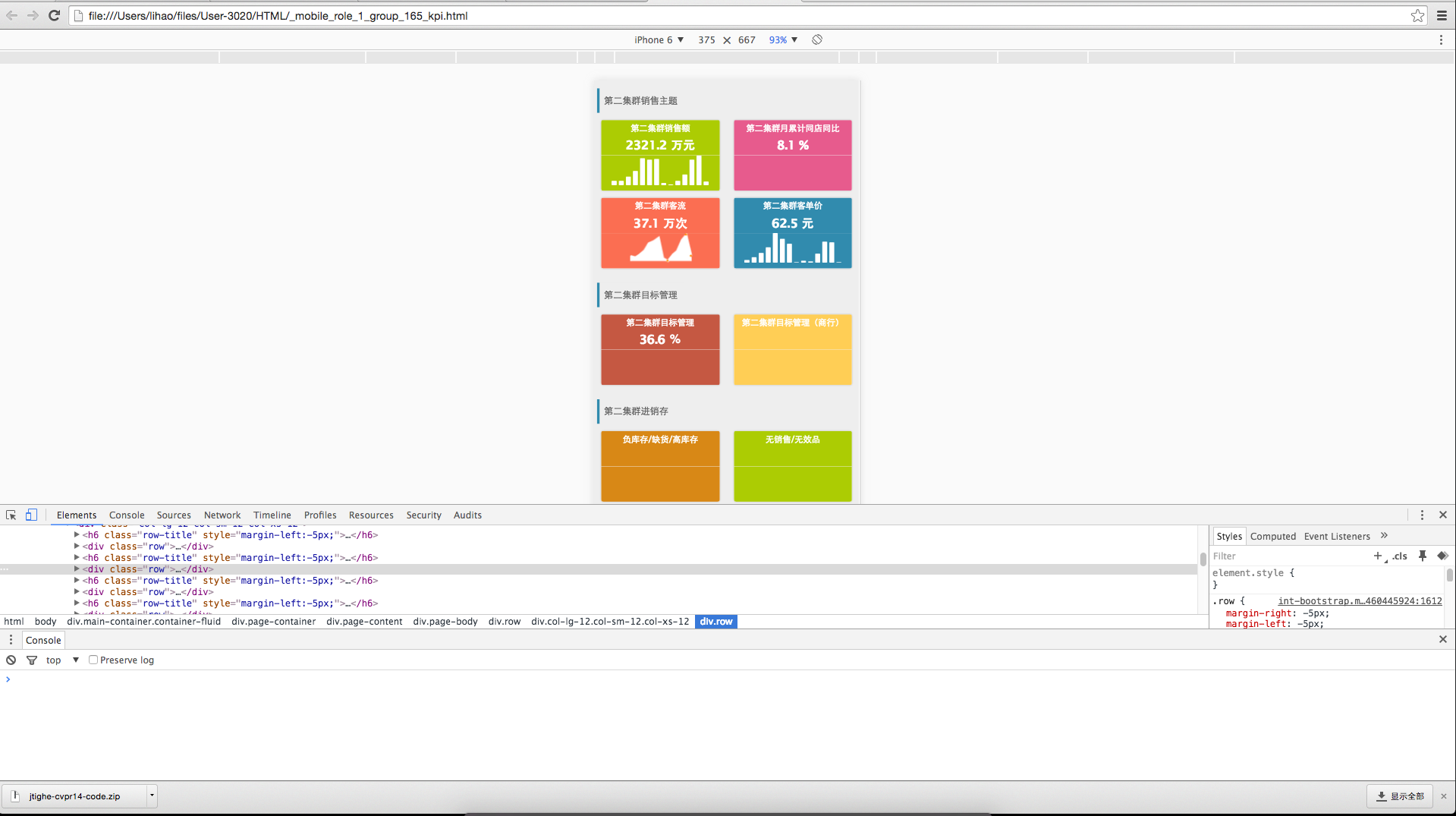
Q5: HTML中的样式文件是如何加载的？

<link href='../../Shared/assets/stylesheets/mobile.css' rel='stylesheet'>

<http://121.199.38.185:4567/mobile/report/10/group/165>

Q6: 把HTML拷贝至电脑上，然后浏览器打开（目标: 与移动端看到的界面一致）

调用的是上两层的Shared/assets/stylesheets中的css，需要保持目录结构



1. 分析中点击某个分析的界面路径? （随便一个）

Q4: 分析链接为pdf时，是如何在应用中展示的？

Q5： pdf控件读取的链接还是文件？

读取文件：/data/data/com.intfocus.yh\_android/files/Cached/f3cd3e034effc757cf85c9164735d314.pdf

使用PDFView控件读取pdf文件，

http://123.59.75.85:8080/yhportal/horseRacing/horseRacing.pdf

1. 应用中点击某个应用的界面路径? （随便一个）

http://izoom.mobi/demo/saoyisao.html?userid=3020&timestamp=20160411023537



1. 点击评论，进入CommentActivity，即可填写评论，点击确认，开启mRunnableForLogger线程，调用ApiHelper.actionLog(mContext, logParams)方法，上传用户行为

Q8: 上传用户行为，服务器api, 参数，响应

服务器：http://121.199.38.185:4567/api/v1/android/logger

参数：

{

"action\_log": {

"action": "解屏",

"app\_version": "a1.2.14",

"user\_name": "张磊",

"user\_id": 3020,

"user\_device\_id": 734

},

"user": {

"user\_pass": "d27004b28a82aa04a7d6ef3026a4eaf4",

"user\_name": "张磊"

}

}

响应：

{

"info": "创建成功",

"code": 201

}

Q9: 那些用户行为被上传到服务器了

登录、解屏

1. 评论的界面路径？

http://121.199.38.185:4567/mobile/id/objectID/type/objectType/comment

http://121.199.38.185:4567/mobile/id/3/type/3/comment

2．Q4: 发表评论的服务器链接及参数、响应

链接：http:// 121.199.38.185:4567/api/v1/user/3020/id/6/type/1

参数：userID、objectID、objectType

Q6: 参照登录逻辑的写法重构一下

服务器地址：http://121.199.38.185:4567/ api/v1/user/userID/id/ objectID/type/ objectType";

上传参数：params：{“content”:”test”,”user\_name”:”张磊”,”object\_title”:” 测试报表模版2”}

服务器响应：responseBody:{

"info":"评论(21) 创建成功.",

"code":201

}



8.点击设置，进入SettingActivity



1. 点击修改登录密码，进入ResetPasswordActivity，输入新密码

重置密码的界面路径？

<http://121.199.38.185:4567/mobile/update_user_password>

Q8: 界面路径，即移动端界面的源代码文件路径

file:////data/data/com.intfocus.yh\_android/files/User-3020/HTML/\_mobile\_update\_user\_password.html

Q6: 提交的服务器链接、参数、响应信息、后续操作？

http://121.199.38.185:4567/api/v1/update/3020/password

上传参数：params：{“ password”:” 96e79218965eb72c92a549dd5a330112”}

服务器返回：{"info":"密码更新成功","code":201}

退出重新登录

Q6: android java bug： 查看日志，找一下bug原因

Bug已经修复完成

提交修改的密码后需要重新登录



1. 点击校正，检测服务器静态资源是否更新，并下载

Q4: 校正的逻辑

检测服务器静态资源是否更新，若更新则下载

读取本地资源的MD5值与服务器资源MD5值进行比对，一样则无需更新

Q7: 从服务器下载资源后的操作（HTML不会加载zip的）

创建DownloadAssetsTask线程，使用输入输出流来传输zip文件

下载资源的链接：http://121.199.38.185:4567/api/v1/download/ assetName.zip

3）点击检测版本更新，判断版本号，偶数时为正式版本

1. 点击启用锁屏，设置密码，返回至桌面后重新进入app需要输入锁屏密码

如果判断是否使用了锁屏？

将userJSON中的use\_gesture\_password设置为true。

设置锁屏后操作？

输入两次锁屏密码

Q7: 什么情况下会触发锁屏界面的出现？

在LoginActivity中的onCreate方法中，获取intent的信息。如果是从触屏界面过来，即intent有from\_activity的信息并且from\_activity中包含ConfirmPassCodeActivity则直接进入主界面。

不是的话，相当于直接启动应用，则检测是否有设置锁屏，若设置锁屏，进入ConfirmPassCodeActivity。

Q8： 是否有设置锁屏？ 即有锁屏，与无锁屏，分别的业务逻辑

有锁屏：返回至桌面后，无需重新登录，直接进入解锁界面，输入正确密码后进入mainActivity

无锁屏：返回至桌面后，需要重新登录

若输入锁屏密码时点击取消，则跳转至LoginActivity。

Q7: 锁屏、输入正确密码后的后续操作

输入密码后，与本地的/data/data/com.intfocus.yh\_android/files/user.plist文件校验锁屏密码是否正确，正确则程序跳转至mainactivity的仪表盘界面

Q8: 这是表象，看一下这块的逻辑代码

读取本地的/data/data/com.intfocus.yh\_android/files/user.plist，使用readConfigFile方法，将user.plist转化为JSON格式，如果密码正确，上传解屏的用户行为信息，进入mainactivity的仪表盘界面

