

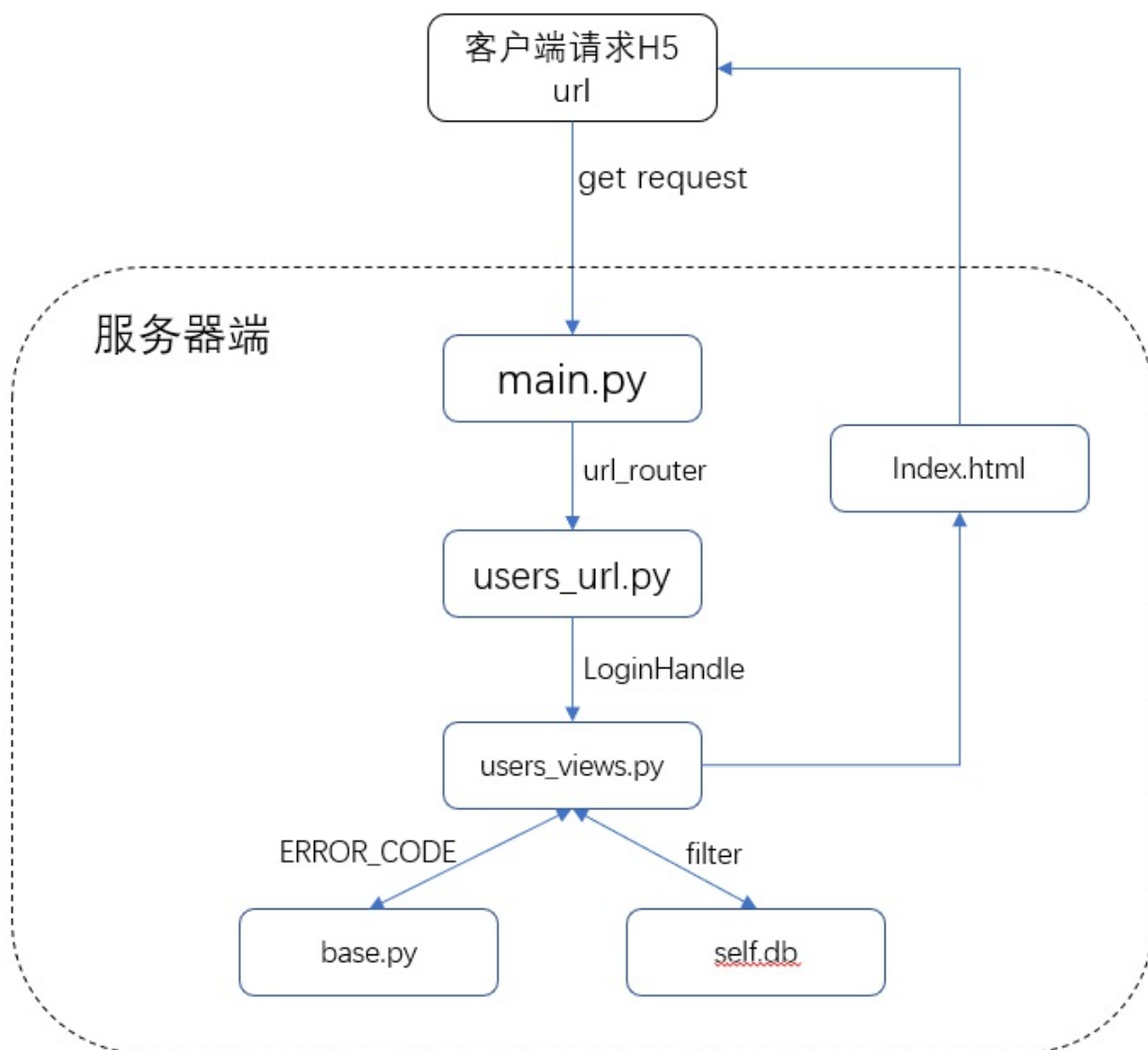
服务器接收客户端请求，并返回 H5 页面

在前几节中，我们讲解了客户端与服务器端数据的交互及图片的上传加载，这一节将讲解 H5 页面的请求及加载。

在 App 客户端的设计中，一般的公司都会要求两个端，分别为 Android 和 iOS 端。如果是客户端负责页面的生成，那 Android 端和 iOS 端都将分别做重复的工作，另一个问题是，可能由于开发人员不一样，页面的设计有出入。现在主流的思想会倾向于使用 H5 嵌入客户端当中，H5 文件存放在服务器端，客户端只负责请求并加载。这样就给服务器端提出了一个问题：如何将服务器上的 H5 文件返回给客户端？本小节将解答这个问题。

调用逻辑

在第 6 小节的讲解中，用户的信息已注册并写入数据库，这一小节中，我们将模拟该用户登录请求，并在登录成功后，返回 App 首页（即本小节预设的 H5 页面）。下面是本小节涉及的请求流程图。



由于我们需要通过浏览器代替 App 客户端进行用户注册请求模拟，此次客户端请求将使用 GET 方法。请求进入服务器端的 `main.py` 后，将调用 `url_router` 转发到 `users_url.py` 中，在 `users_urls.py` 中，对应的 URL 将调用 `users_views.py` 的 `LoginHandle` 类。`LoginHandle` 为真正的代码处理逻辑，在校验用户信息正确的情况下，返回 `index.html` 页面给客户端，客户端加载该页面。

服务器端实现

由上面的调用逻辑图可知，我们将从 main.py 开始修改，由于在第 6 小节中，main.py 针对 users 的路由已配置，这里 main.py 不需要修改。接下来修改 users_url.py，在 users_url.py 中增加 LoginHandle 的调用，添加如下两行。

```
#!/usr/bin/python
# -*- coding:utf-8 -*-

from __future__ import unicode_literals
from users_views import (
    RegistHandle,
    LoginHandle
)

urls = [
    #从/users/regist过来的请求，将调用users_views里面的RegistHandle类
    (r'regist', RegistHandle),
    (r'login', LoginHandle)
]
```

完整代码如下：

```

#!/usr/bin/python3
# -*- coding:utf-8 -*-

from __future__ import unicode_literals
from .users_views import (
    RegistHandle,
    LoginHandle
)

urls = [
    #从/users/regist过来的请求，将调用users_views里面的
    #RegistHandle类
    (r'regist', RegistHandle),
    (r'login', LoginHandle)
]

```

接下来，添加真正的代码处理，修改 `users_views.py`，增加 `LoginHandle` 类代码如下：

```

class LoginHandle(tornado.web.RequestHandler):
    """handle /user/regist request
    :param phone: users sign up phone
    :param password: users sign up password
    """

    @property
    def db(self):
        return self.application.db

    def get(self):
        try:
            #获取入参

```

```

        phone = self.get_argument( "phone" )
        password = self.get_argument(
"password" )
    except:
        #获取入参失败时，抛出错误码及错误信息
        logger.info("LoginHandle: request
argument incorrect")
        http_response(self,
ERROR_CODE['1001'], 1001)
        return

    #从数据库 Users 表查找入参中的 phone 是否存在
    ex_user =
self.db.query(Users).filter_by(phone=phone).first
()

    if ex_user:
        #如果手机号已存在，返回首页 H5 页面
index.html
        logger.debug( "LoginHandle: get user
login: %s" %phone )
        self.render( "index.html" )
        self.db.close()
        return
    else:
        #用户不存在，提示用户未注册
        http_response( self,
ERROR_CODE['1003'], 1003 )
        self.db.close()
        return

```

这里的新增错误码 1003 表示用户未注册，需在配置文件中添加此错误码，编辑 base.py 增加如下代码：

"1003": "用户尚未注册，请先注册"，

```
11
12  ERROR_CODE = {
13      "0": "ok",
14      #Users error code
15      "1001": "入参非法",
16      "1002": "用户已注册，请直接登录",
17      "1003": "用户尚未注册，请先注册",
18
19      "2001": "上传图片不能为空"
20  }
21
```

至此，服务器端的代码逻辑已基本完成，现在唯一缺少的就是 index.html 这个文件。

H5 页面代码

由于本小册的重点并不是讲解 H5，这里请读者直接按照指导将 H5 涉及的代码输入服务器端，本小册不做另外的讲解。

新增 index.html 文件

进入“templates”目录，创建并编辑 index.html 文件，输入如下代码：

```
[root@VM_0_8_centos demo]# ls
common  conf  log  main.py  models.py  models.pyc  static  templates
[root@VM_0_8_centos demo]# cd templates/
[root@VM_0_8_centos templates]# touch index.html
[root@VM_0_8_centos templates]# ls
index.html
[root@VM_0_8_centos templates]#
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <title>css网页布局</title>
```

```
<link rel="stylesheet"
href="../static/css/index.css">
</head>
<body>
<!--整体到部分，左到右，上到下-->
<!--头部-->
<div class="header">
    <div class="logo">
        <ul>
            <li>DEMO</li>
        </ul>
    </div>
    <div class="nav">
        <ul>
            <li>About</li>
        </ul>
    </div>
</div>
<!--主体-->
<div class="main">
    <div class="top">
        
    </div>
    <!--遮罩层-->
    <div class="topplayer"></div>
    <!--最上层的内容-->
    <div class="topplayer-top">
        <div class="word">MY DEMO H5</div>
    </div>
    <div class="middle">
        <div class="m-top">
```

```
<div class="demo-layer demo1">
  
  <div class="demo">DEM01</div>
</div>
<div class="demo-layer demo1">
  
  <div class="demo">DEM02</div>
</div>
<div class="demo-layer demo3">
  
  <div class="demo">DEM03</div>
</div>
<div class="clear"></div>
</div>
<div class="m-middle">
  "Life is like riding a bicycle.
To keep your balance, you must keep moving."
</div>
<div class="m-bottom">
  <div class="m-com">
    
    <div class="demo4">Cool
Demo</div>
    <div class="demo5">Make it
cool</div>
  </div>
  <div class="m-com">
    
```



```

        <div class="demo4">Great
Demo</div>
        <div class="demo5">Make it
great</div>
    </div>
    <div class="m-com">
        
        <div class="demo4">Wonderful
Demo</div>
        <div class="demo5">Make it
wonderful</div>
    </div>
</div>
</body>
</html>

```

新增 index.css 文件

进入“static/css”目录，创建并编辑 index.css 文件，输入如下代码：

```

[root@VM_0_8_centos demo]# ls
common  conf  log  main.py  models.py  models.pyc  static  templates  views
[root@VM_0_8_centos demo]# cd static/css/
[root@VM_0_8_centos css]# touch index.css
[root@VM_0_8_centos css]# ls
index.css
[root@VM_0_8_centos css]#

```

```

* {
    margin: 0;
    padding: 0;
}
.header {

```

```
    width: 100%;
    height: 100px;
}
.header img {
    width: 300px;
    height: 85px;
    padding-left: 100px;
    padding-top: 8px;
}
.header .logo {
    float: left;
    margin-top: 40px;
    margin-left: 40px;
}
.header .nav {
    float: right;
}
.header .nav ul {
    margin-right: 20px;
}
.header .nav ul li {
    float: left;
    list-style: none;
    width: 80px;
    height: 100px;
    line-height: 100px;
    color: #7d7d7d;
    font-size: 15px;
    font-weight: bolder;
}
.main .top {
    width: 100%;
    height: 600px;
```

```
}  
.main .top img {  
    width: 100%;  
    height: 600px;  
}  
.main .toplayer {  
    position: absolute;  
    top: 100px;  
    background: #000000;  
    width: 100%;  
    height: 600px;  
    opacity: 0.5; /* 透明度 */  
}  
.main .toplayer-top {  
    width: 500px;  
    height: 300px;  
    position: absolute;  
    top: 400px;  
    margin-top: -150px;  
    z-index: 2;  
    right: 50%;  
    margin-right: -250px;  
}  
.main .toplayer-top .word {  
    padding-top: 100px;  
    color: #ffffff;  
    font-size: 45px;  
    font-weight: bolder;  
    text-align: center;  
    font-family: "微软雅黑";  
}  
.main .toplayer-top button {  
    width: 200px;
```

```
height: 60px;
margin-top: 50px;
color: #ffffff;
background: #f5704f;
font-family: 微软雅黑;
text-align: center;
font-weight: bolder;
font-size: 14px;
border-radius: 8px; /* 圆角 */
margin-left: 150px;
}
.main .middle {
width: 1000px;
margin: 0 auto;
}
.main .middle .m-top .demo-layer {
float: left;
width: 33.3%;
padding-top: 50px;
text-align: center;
}
.main .middle .m-top .demo-layer img {
width: 100px;
height: 100px;
}
.main .middle .m-top .demo-layer .demo {
font-size: 20px;
color: #7d7c7f;
font-weight: bold;
padding-top: 20px;
}
.main .middle .m-middle {
font-size: 25px;
```

```
    color: #000000;
    font-weight: bold;
    padding-top: 50px;
    text-align: center;
    padding-bottom: 50px;
}
.clear {
    clear: both;
}
.main .middle .m-bottom .m-com {
    float: left;
    padding: 10px;
    text-align: center;
    font-weight: bold;
    font-size: 20px;
}
.main .middle .m-bottom .m-com img {
    width: 310px;
    height: 260px;
}
.main .middle .m-bottom .demo4 {
    padding-top: 20px;
    color: #7d7d7f;
}
.main .middle .m-bottom .demo5 {
    padding-top: 10px;
    color: #bdbdbc;
}
.main .bottom {
    width: 1000px;
    margin: 0 auto;
}
.footer {
```

```
width: 100%;  
height: 100px;  
text-align: center;  
line-height: 100px;  
background: #292c35;  
color: white;  
font-family: "微软雅黑";  
font-size: 15px;  
}
```

上传 H5 页面图片

进入“static/image”目录，创建 index 文件夹，上传 H5 页面图片。

```
mkdir index  
cd index/  
rz -be
```

具体如下图所示：

```
[root@VM_0_8_centos demo]# ls  
common conf log main.py models.py models.pyc static templates  
[root@VM_0_8_centos demo]# cd static/image/  
[root@VM_0_8_centos image]# ls  
demo.jpg  
[root@VM_0_8_centos image]# mkdir index  
[root@VM_0_8_centos image]# ls  
demo.jpg index  
[root@VM_0_8_centos image]#
```

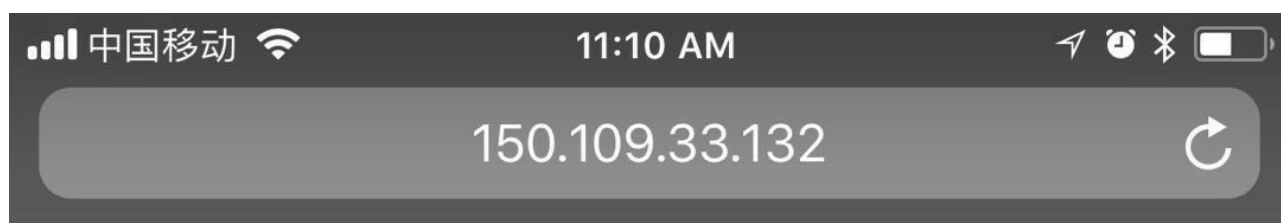
```
[root@VM_0_8_centos image]# cd index/  
[root@VM_0_8_centos index]# rz -be  
rz waiting to receive.  
zãÊ¼ zmodem «Êäîf ° ctrl+c ÈîÛîf  
[root@VM_0_8_centos index]# ls  
1.jpg 2.jpg 3.jpg 4.jpg 5.jpg 6.jpg index.jpg  
[root@VM_0_8_centos index]#
```

至此，H5 的页面准备工作已结束， 服务器端的代码已全部完成。接下来将测试请求是否成功。

客户端请求 H5 页面

由于 App 客户端嵌入 H5 页面和手机浏览器直接打开 H5 页面的效果一样，这里在手机的浏览器中直接输入请求 URL 进行测试，URL 为：

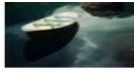
http://150.109.33.132:8000/users/login?
phone=18866668888&password=demo123456。请求成功后，
可以看到加载的效果如下：



• DEMO

About





DEMO1



DEMO2



DEMO3

"Life is like riding a bicycle. To keep your balance, you must keep moving."



Cool Demo

Make it cool



Great Demo

Make it great



Wonderful Demo

Make it wonderful



代码下载

到目前为止，服务器端代码及图片如下：

[demo10 \(https://github.com/Jawish185/demo10.git\)](https://github.com/Jawish185/demo10.git)

小结

这一小节中，我们探讨了为什么建议客户端嵌套 H5 页面，并完成了客户端向服务器端请求 H5 页面的整个代码逻辑学习过程，希望读者能触类旁通，提高产品上线效率。