设计说明

平台开发工程师 袁镇锋

日期：2015/3/17

1. 详细设计
   a) 前端用户界面

   用户界面主要包括注册界面、登录界面和头像上传界面。

   i. 注册界面

图 1 注册界面

   ii. 登录界面

图 2 登录界面

   iii. 头像上传界面

图 3 头像上传页面

b) 数据库

i. 字段说明

Yagra 需要存储到数据库的数据为：用户名(username)、密码(password)和盐值(salt)。

表 1 数据库字段

字段	类型	含义	备注
id	INT	主键、自增	PRIMARY KEY AUTO_INCREMENT
username	VARCHAR(128)	用户名	UNIQUE
password	VARCHAR(128)	密码	(盐值+用户密码)的 md5 值
salt	VARCHAR(32)	盐值	用于该用户的密码加密

ii. 建表语句

```
CREATE TABLE `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(128) BINARY NOT NULL,
  `password` varchar(128) NOT NULL,
  `salt` varchar(32) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

图 4 建表语句

c) 登录状态的保持

登录状态的保持通过 cookie 和 session 协同工作进行。对于客户端的每一次登陆，在服务器生成一个 session，作为文件存储在服务器上。文件命名为 yagra_sid.session，其中 sid 为服务器在客户端登录时生成的**唯一的随机字符串**。Session 文件存储的数据为用户名，同时，把 sid 作为 cookie 数据发回给客户端。在需要验证身份的页面中，只需要检测 cookie 中的 sid 是否有效就可以了，然后去对应的 session 文件里面获取用户数据。

d) 安全性问题

i. 表单数据

对于表单的数据，分别在前端和后端都做了检查，保证数据是合法的格式。另外，服务端还使用了 cgi.escape()来防止**脚本注入攻击**。针对 f) ii 静态页面显示模块，还使用 os.path.basename()来防止**目录目录遍历攻击**。

ii. 用户密码

数据库储存的是**加密过的密码**。所以就算数据库泄露了数据，密码也没有泄露。另一方面，密码的加密是在 md5 算法的基础上，加入了**盐值**，而且每个用户的盐值也是不一样的，大大地加大了密码的安全性。

iii. 登录状态

Cookie 中只存储 sid，而没有任何用户数据，使得 cookie 没那么容易伪造。

e) 头像访问 API

头像访问 API 由 f) 中的头像访问模块提供。访问方式为：

http://xxxxxxxx/avatar.py?avatar=**md5 of username**

若用户名无效或者用户没有上传过头像，则返回系统默认头像。

图 5 头像访问

f) 服务端模块

i. 工具模块(util.py)

该模块主要封装了其他模块会用到的一些操作的接口、包括对数据库的读写操作、session 的相关操作以及对接收到的账号密码的格式检测等。

ii. 静态页面获取模块(page_handler.py)

该模块用于静态页面的获取(只能获取 css 和 js 文件的内容)。

具体过程如下：

(1) 获取 QueryString 里面的 page 参数；

(2) 根据 page 参数，返回用户请求的页面；

比如想访问 css/common.css，那么链接地址为：

http://localhost/page_handler.py?page=common.css

iii. 注册模块(register.py)

当使用 get 方式访问时，显示注册页面；

当使用 post 方式访问时，用于响应用户的注册操作(客户端通过 ajax 发送请求)。

具体过程如下：

(1) 从 QueryString 里面获取 username 和 password 参数；

(2) 检测参数的有效性；

(3) 进行注册；

(4) 返回操作结果的 json 数据给客户端；

iv. 登录模块(login.py)

当使用 get 方式访问时，显示登录页面；

当使用 post 方式访问时，用于响应用户的登录操作(客户端通过 ajax 发送请求)。

具体过程如下：

(1) 从 QueryString 里面获取 username 和 password 参数；

(2) 检测参数的有效性；

(3) 进行登录；

(4) 返回操作结果的 json 数据给客户端；

若登录成功，还会设置 cookie 信息。

v. 登出模块(logout.py)

该模块用于登出操作。模块把 session 文件删除，并将 cookie 中的变量的有效期设为一个过期的时间，使得 cookie 中的数据无效。

vi. 上传头像模块(upload_avatar.py)

当使用 get 方式访问时，显示上传头像的页面；

当使用 post 方式访问时，用于处理用户上传的图片。

具体过程如下：

(1) 从获取登录用户的信息；

(2) 保存用户的头像；

若用户上传的文件非图片，则上传失败。

vii. 头像访问模块(avatar.py)

该模块提供头像访问功能。头像的访问有 2 种情况：

(1) 访问当前登录用户的头像，此时，api 接口为：

http://localhost/Yagra/avatar.py

若未登录，则返回默认头像

(2) 访问某个用户的头像，此时，api 接口为：

http://localhost/Yagra/avatar.py?avatar=**用户名的 md5 值**

如果该用户不存在，或者，该用户没有上传过头像，那么将返回系统默认的头像。