

(1)

V4

<https://edu.fanruan.com/video/191>

OAuth2 http://www.ruanyifeng.com/blog/2019/04/oauth_design.html

OAuth2 <http://www.ruanyifeng.com/blog/2019/04/oauth-grant-types.html>

<https://work.weixin.qq.com/api/doc/90000/90135/91020>

access_token<https://work.weixin.qq.com/api/doc/90000/90135/91039>

API

PC/APP/H5 LoginResource

RemoteDesignResource

F12pathinfo

APILoginService

1EventDispatcher.fire(LoginOutEvent.LOGIN, new LoginOutResultInfo(req, res, loginRequestInfoBean.getUsername(), true));

EventDispatcher

2APITransmissionTool.decrypt(boolean encrypted, boolean supportCustomEncrypt, String encryptionInfo)

PCAPI

3SQLWebServiceUtils.containsSQLChars(String inputString)

4IPWebServiceUtils.getIpInfoFromRequest(HttpServletRequest req)

5UserService.getInstance().getUserByUserName(String username);

UserService

6IDControllerFactory.getInstance().getAuthenticController(String userID)

PassportProvider

7AuthenticController#dealWithLoginLock(User user, Device device, String lockObject, String lockObjectValue);

lockObjectIP; lockObjectValueIP

8AuthenticController#doAuthentication(User user, String password, Device device)

9FSConfig.getInstance().getLoginConfig().getLoginTimeout()

FSConfiginedb

10fine_auth_token

```
DecisionStatusService.loginStatusService().put(String token, LoginClientBean clientBean, new
Converter<LoginClientBean>() {
    @Override
    public String[] createAlias(LoginClientBean loginClientBean) {
        return new String[]{loginClientBean.getUsername()};
    }
}, long tokenTimeout);
```

11MessageService.getInstance().sendMessageByUser(User user, String content, String url, MessageUrlType urlType);

```
12String token = LoginService.getInstance().login(HttpServletRequest req, HttpServletResponse res, String username)
```

1

22

a.

b.

395%http queryheaderbodysocket

42

a.MD5/SHA

b.OAuth2CAS

[demo](#)