

网络 - 基于AFNetworking-详见工程HttpTool.h

普通情况

```
- (void)qryUserInfo
{
    //
    NSMutableDictionary *reqDic = [[NSMutableDictionary alloc] init];    [reqDic se
tObject:@"aa111" forKey:@"userName"];
    [reqDic setObject:@"111aaa" forKey:@"password"];
    // post
    [HttpTool postUrl:GJS_QRY_USER_INFO params:reqDic success:^(id
responseObj) { //
        [self req_callback:responseObj];
    } failure:^(NSError *error) {
        //
    }]];
}

- (void)req_callback:(id)data
{
    if (data) {
        NSDictionary *body = [NSDictionary
dictionaryWithDictionary:data]; //
    }
}
```

带header情况

```

- (void)qryUserInfo
{
    //
    NSMutableDictionary *reqDic = [[NSMutableDictionary alloc] init];    [reqDic se
tObject:@"aa111" forKey:@"userName"];
    [reqDic setObject:@"111aaa" forKey:@"password"];
    // header
    NSMutableDictionary *headerDic = [[NSMutableDictionary
    alloc] init];
    [headerDic setObject:@"aa111" forKey:@"userName11"];
    [headerDic setObject:@"111aaa" forKey:@"password11"];
    // post
    [HttpTool postUrl:GJS_QRY_USER_INFO params:reqDic
    headerDic:headerDic success:^(id responseObj) {
    //
        [self req_callBack:responseObj];
    } failure:^(NSError *error) {
    //
    }]];
}

- (void)req_callBack:(id)data
{
    if (data) {
        NSDictionary *body = [NSDictionary
        dictionaryWithDictionary:data]; //
    }
}

```

HUD-基于CustomFirstRateButton。详见 CustomFirstRateButton.h

```

- (void)validPhoneAction
{
    [_nextBtn showLoadingStatus];
    __weak typeof(self) weakSelf = self;
    RegistVerifyPhoneViewModel *viewModel = [[RegistVerifyPhoneViewModel alloc] initWith];
    [viewModel setReqBlockWithRetValue:^(id returnValue) {

        __strong typeof(weakSelf) strongSelf = weakSelf;
        if (strongSelf) {

            [strongSelf.nextBtn hiddenLoadingStatus];
        }

        } withErrorBlock:^(id errorCode, NSString *errorNote) {

            [weakSelf.nextBtn hiddenLoadingStatus];

        } withFailureBlock:^(

            [weakSelf.nextBtn hiddenLoadingStatus];

        )];
    [viewModel startRequestCheckUserName:self.name];
}

```

refresh控件-基于MJRefresh-详见RefreshTool.h

```

GJWeakSelf;
//
_tableView.mj_header = [RefreshTool initWithHeaderRefresh:^(
    GJStrongSelf;
    if (strongSelf) {
        [strongSelf refreshCurData];
    }
}];

```

```

//
_tableView.mj_footer = [RefreshTool initWithFooterRefresh:^(
    GJStrongSelf;
    if (strongSelf) {
        [strongSelf loadMoreData];
    }
}];

```

分类、自定义

主要包含在BaseFrame/Main/Category和BaseFrame/Main/CustomExtension中

- 所有的UIViewController基于CustomBaseViewController或者CustomBaseLayoutViewController
- 所有的主按钮基于CustomFirstRateButton，次等按钮基于CustomSecondRateButton
- 所有的警告框基于CustomShowWarningMsgView或者CustomShowWarningFrameLayout
- 所有的navbar使用CustomNavTopView或者CustomNaTopFrameLayout替代
- 所有的Layout基于MyLayout库
- 所有的UITableview基于CustomTableView
- 所有的无网络提示框使用NoDataOrNetworkView展示
- 所有的输入框优先使用CustomTextField或者SafeTextField
- 所有的弹出框基于CustomAlertView
- 所有的网络错误提示判断都需要执行[HttpTool handleErrorCodeFromServer:withNote:]
- MSVC框架按需使用
- MyLayout框架按需使用
- 数据采集类GjFaxDataCollection等分为三期，按需使用
- 富文本TBCityCoreText库等
- CustomKeyBoard安全键盘

用户信息单例使用示例

如果需要存储用户信息类型在UserInfoTypeDef.h中找不到，可在相应枚举中添加

```
// BOOL 类型信息
UserInfoUtil->setUserInfoWithBool(UserLoginStatusType, NO);
UserInfoUtil->getUserInfoWithBool(UserLoginStatusType);

// 值类型信息
UserInfoUtil->setUserInfoWithValue(UserUserNameType, @"aaal11");
UserInfoUtil->getUserInfoWithValue(UserUserNameType);
```