# **FdogIM即时通讯软件开发文档**

作者：花狗Fdog(张旭) 2021.7.24

-------------------------------------------------------------------------------------------------------------

# 前言

FdogIM中的Fdog源之于我的江湖之名，三个月前，突然有了一些灵感，想着将自己所学知识进行一个融会贯通，做出来一个小demo也不枉自己的大学生活，同时也希望简历上项目一栏不至于空白，于是画了两个月的闲暇时间，做出了一个简陋的FdogIM，收到了一些反馈，比如说web后端写的一团糟，之前在学校，并没有清晰的前后端分离的概念，在html切入java代码变成了jsp，听取了一些建议，这次web后端使用了go，并将前后端分离，也学习到了一些不足，编码前只是进行了简单分析，导致后面代码越来越多，业务逻辑也越来越混乱，数据库用到才开始设计，到最后我意识到了问题严重性，我决定从头再来。

给大家一些建议：

1. 写代码之前应该先完成文档的编写，包括整体框架，功能需求，数据库设计，以及函数的命名，而不是遇到一个写一个，业务量上去之后绝对行不通。

Ps：如果你的代码只有短短几百行，我想文档倒是会变成累赘，我曾向别人问过一道奇葩的问题，这道题大多数人都会经历，请看：

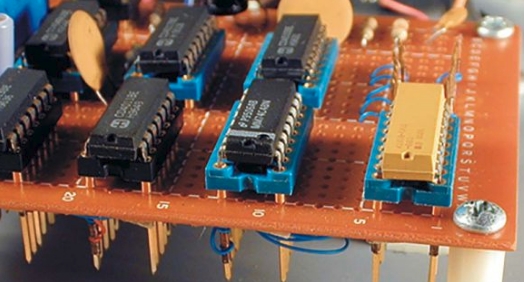
题目：请打印一个三角形。

我想大多数人看见这道题一定会想到使用for循环来做，没错，但是有没有想过使用三条printf函数也是可以完成这道题的，反而比for更加简单，在算法复杂度的角度也是如此，for倒是变的格格不入。

1. 多写活代码，而不是死代码，使用配置文件代替代码中关键变量，当变量改变时，只需要修改配置文件，而无需修改代码本身。

Ps：例如我的程序需要连接数据库，但是我数据库的名字可能会改变，当我的程序编译完成，改变数据库的名字，意味着我需要修改我的代码并且重新编译，而使用配置文件，我只需重新运行我的程序即可。

1. 编写测试程序对你的功能进行测试。
2. 正式场景中，不可能使用控制台输出错误，考虑将日志加入到你的程序中。
3. 好好理解高内聚低耦合，将功能具体集中，只提供尽量少的接口来完成业务。

低内聚 高耦合 高内聚 低耦合

联系方式：

Mail：2506897252@qq.com

我的博客：<https://zxfdog.blog.csdn.net/>

-------------------------------------------------------------------------------------------------------------

# FdogIM简介

FdogIM即FdogInstantMessaging(Fdog即时通讯软件)，采用多线程实现高并发，客户端与服务端之间通信使用SDK提供接口，服务端与SDK之间使用thrift协议生成不同语言版本的接口，您无需担心您的客户端是由何种语言构成，同样也无需考虑服务端内部实现，只需根据SDK文档提供对应参数即可，当然您也可以对服务端进行修改，来达到您满意为止。

该项目除了Server端之外还特意提供了相应的Web端，Pc端，Android端(计划中)，供您参考，对于初学者，我想这是一个很好的机会，如果对您有所帮助，请给我star吧！

Server端：使用C++编写，运行在服务器上，作为后台负责业务逻辑，并使用thrift协议对接SDK，保证后期不同语言的客户端调用SDK，如java。

Web：前端使用使用html+js，后端使用golang(有使用gin框架)，用于Fdog官网及账号注册，账号找回，软件下载等业务。

Pc端：使用Qt框架进行客户端界面绘制开发，使用提供的SDKC++版本和服务端进行通信。

短信服务提供商：腾讯云。

数据库为Mysql。

Thrift：是Facebook的一个开源项目，主要是一个跨语言的服务开发框架。

项目的部署非常简单，代码无需大量修改，相关变量将以配置文件的形式展示给您，您只需要根据文档中相应部署说明进行填写即可。

项目源码已上传github，欢迎star！

https://github.com/HuaGouFdog/FdogInstantMessaging/

-------------------------------------------------------------------------------------------------------------

# 编译，运行环境说明

首先，服务器应当拥有C++以及Mysql运行环境，通过配置文件配置服务端的ip，端口，使其成功运行在服务器之中，通过SDK对外提供对应的接口供客户端调用，为防止SDK被恶意调用，采用授权的方式进行鉴权，通过后，方可正常使用SDK提供的接口。

-------------------------------------------------------------------------------------------------------------

# 客户端对于SDK的需求

## 服务端需对客户端进行鉴权

## 登录客户端

## 填充客户端信息

## 用户通讯

# SDK文档

## AppidAndAppSecretVerify

函数原型：

string \* AppidAndAppKeyVerify(const string Appid, const string Appkey) const

说明：

接入方提供对应的Appid和Appkey作为参数调用该接口，鉴权成功，该接口将返回一个字符串，作为NewClient参数之一，鉴权失败，将返回空字符串。

1. NewClient

函数原型：

Client \* NewClient(const string token,)

说明:

## AccountPasswordVerify

验证手机号，密码是否正确

## GetAccountInfo

获取当前用户基本信息

## Dsa

## Dsa

## Ds

## Dsa

## Dsa

## Dsad

## Sada

## dsadsa

# 服务端的编写

# 客户端的编写

# 测试工具编写

# 数据库设计