# **总览**

# **基础接口**

约定, 所有接口暴露成 libffi 风格(标准 C)接口接口

所有接口提供俩种模式, 文件路径模式和流模式

* 文件路径模式

形式如下:

int XXXByPath(char \* path, char \*outpath)

// path作为输入文件的路径

// outpath 作为输出文件的路径

// 返回1 代表成功 负数代表错误代码

* 流模式

XXXCtx CreateXXXBlobCtx();

// 建立一个上下文

int XXXBlobChunk(XXXCtx\* ctx, char \* out\_buffer, int out\_buffer\_size);

// 根据上下文和一次最多能接受的 buffer 得尺寸和 buffer, 进行处理

// 返回处理的字数 当这个数小于 out\_buffer\_size 处理完成

流模式需要一个上下文, 根据上下文的内容继续之前的处理, 上下文主要存有当前调用执行的进度, 再次调用会根据上下文的信息继续进度

## **加水印**

文件模式

int EnWaterMark(char\* path, char \*content, char \* outpath) //加水印

int DeWaterMark(char\* path, char \*content) //解水印 , 结果放入 content

流模式

EnWaterMarkCtx CreateEnWaterMarkBlobCtx(char \* input, char \*content);

// 建立一个加水印上下文

int EnWaterMarkBlobChunk(EnWaterMarkCtx\* ctx, char \* out\_buffer, int out\_buffer\_size);

// 根据上下文, 每次返回 out\_buffer\_size的结果, 结束返回 0, 失败返回负数

DeWaterMarkCtx CreateDnWaterMarkBlobCtx(char \* input);

int DeWaterMarkBlobChunk(DeWaterMarkCtx\* ctx, char \* out\_buffer, int out\_buffer\_size);

// 根据上下文, 每次返回 out\_buffer\_size的结果, 结束返回 0, 失败返回负数

## **加解密**

文件模式

int EncryptFileByPath(char\* path, char \*key, char \* outpath) //加密

int DecryptFileByPath(char\* path, char \*key, char \* outpath) //解密

流模式

EncryptFileCtx CreateEncryptFileBlobCtx(char \* input, char \*key);

// 建立一个加密上下文

int EncryptFilekBlobChunk(EncryptFileCtx\* ctx, char \* out\_buffer, int out\_buffer\_size);

// 根据上下文, 每次返回 out\_buffer\_size的结果, 结束返回 0, 失败返回负数

DecryptFileCtx CreateDecryptFileCtx(char \* input, char\* key); //解密上下文

int DecryptFileBlobChunk(WaterMarkCtx\* ctx, char \* out\_buffer, int out\_buffer\_size);

// 根据上下文, 每次返回 out\_buffer\_size的结果, 结束返回 0, 失败返回负数

## **提取截图**

// 根据路径提取

int GetThumbnailByPath(char \* path, char\* output);

# **流程**

### **视频上传**

1. 用户上传视频
2. 客户端提取缩略图
3. 服务器接收视频 和 截图, 分开存放存入 origin , thumbnail
4. 服务器生成密钥, 进行加密, 加密后放入 encrypted,
5. 记录入库 返回响应

### **视频下载**

1. 用户选择视频, 下载密文, 存入本地
2. 客户端解密密文, 获得视频二进制文件, 加水印, 加密放入本地
3. 客户端解密播放视频

### **水印解密**

1. 用户选择文件, 解出水印