**导出消息sdk接口说明**

**整体流程：**

业务方通过企业微信提供的sdk，可以进行聊天记录数据的获取、媒体数据的获取。

以下是交互时序图，本文重点介绍sdk的内容。sdk包括如下接口：

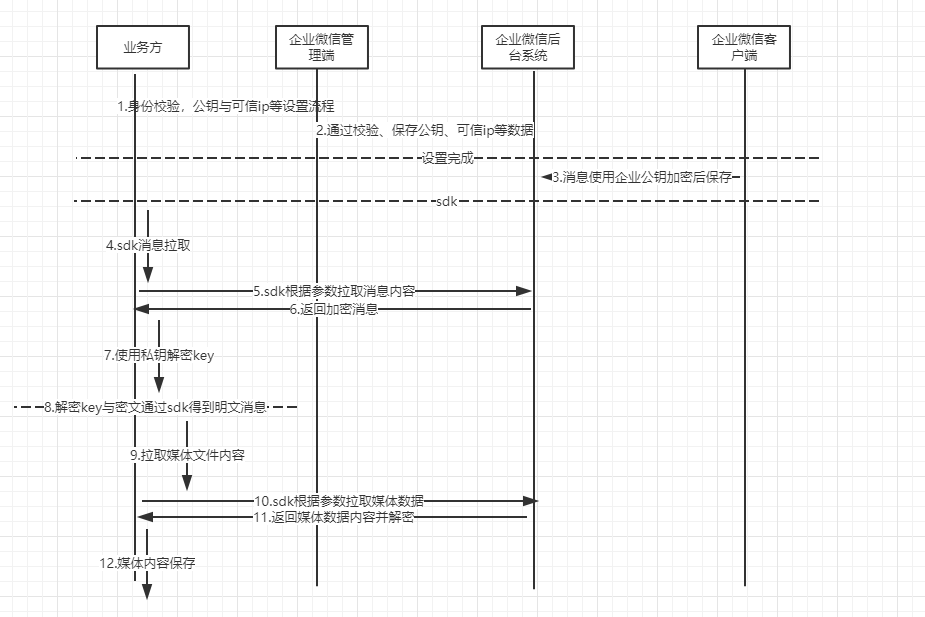
1. 初始化接口；

2. 提供分页批量拉取聊天记录接口的sdk接口；

3. 对聊天数据进行解密接口；

4. 对图片、文件等媒体数据，拉取媒体数据内容的sdk接口;

5. 工具函数，进行数据的析构等处理.



**初始化接口：**

**第一步：**

WeWorkFinanceSdk\_t\* NewSdk();

获取sdk对象，首次使用初始化

**第二步：**

/\*\*

\* 初始化函数

\* Return值=0表示该API调用成功

\*

\*

\* @param [in] sdk NewSdk返回的sdk指针

\* @param [in] corpid 调用企业的企业id，例如：wwd08c8exxxx5ab44d，可以在企业微信管理端--我的企业--企业信息查看

\* @param [in] secret 聊天内容存档的Secret，可以在企业微信管理端--管理工具--聊天内容存档查看

\*

\*

\* @return 返回是否初始化成功

\* 0 - 成功

\* !=0 - 失败

\*/

int Init(WeWorkFinanceSdk\_t\* sdk, const char\* corpid, const char\* secret);

**获取聊天记录数据：**

通过本sdk接口来获取公司一段时间内的聊天记录。一次拉取调用上限1000条聊天记录，可以通过分页拉取的方式来依次拉取。调用频率不可超过600次/分钟。

/\*\*

\* 拉取聊天记录函数

\* Return值=0表示该API调用成功

\*

\*

\* @param [in] sdk NewSdk返回的sdk指针

\* @param [in] seq 从指定的seq开始拉取消息，注意的是返回的消息从seq+1开始返回，seq为之前接口返回的最大seq值。首次使用请使用seq:0

\* @param [in] limit 一次拉取的消息条数，最大值1000条，超过1000条会返回错误

\* @param [in] proxy 使用代理的请求，需要传入代理的链接。如：socks5://10.0.0.1:8081 或者 http://10.0.0.1:8081

\* @param [in] passwd 代理账号密码，需要传入代理的账号密码。如 user\_name:passwd\_123

\* @param [in] timeout 超时时间，单位秒

\* @param [out] chatDatas 返回本次拉取消息的数据，slice结构体.内容包括errcode/errmsg，以及每条消息内容。示例如下：

{"errcode":0,"errmsg":"ok","chatdata":[{"seq":196,"msgid":"CAQQ2fbb4QUY0On2rYSAgAMgip/yzgs=","publickey\_ver":3,"encrypt\_random\_key":"ftJ+uz3n/z1DsxlkwxNgE+mL38H42/KCvN8T60gbbtPD+Rta1hKTuQPzUzO6Hzne97MgKs7FfdDxDck/v8cDT6gUVjA2tZ/M7euSD0L66opJ/IUeBtpAtvgVSD5qhlaQjvfKJc/zPMGNK2xCLFYqwmQBZXbNT7uA69Fflm512nZKW/piK2RKdYJhRyvQnA1ISxK097sp9WlEgDg250fM5tgwMjujdzr7ehK6gtVBUFldNSJS7ndtIf6aSBfaLktZgwHZ57ONewWq8GJe7WwQf1hwcDbCh7YMG8nsweEwhDfUz+u8rz9an+0lgrYMZFRHnmzjgmLwrR7B/32Qxqd79A==","encrypt\_chat\_msg":"898WSfGMnIeytTsea7Rc0WsOocs0bIAerF6de0v2cFwqo9uOxrW9wYe5rCjCHHH5bDrNvLxBE/xOoFfcwOTYX0HQxTJaH0ES9OHDZ61p8gcbfGdJKnq2UU4tAEgGb8H+Q9n8syRXIjaI3KuVCqGIi4QGHFmxWenPFfjF/vRuPd0EpzUNwmqfUxLBWLpGhv+dLnqiEOBW41Zdc0OO0St6E+JeIeHlRZAR+E13Isv9eS09xNbF0qQXWIyNUi+ucLr5VuZnPGXBrSfvwX8f0QebTwpy1tT2zvQiMM2MBugKH6NuMzzuvEsXeD+6+3VRqL"}]}

\*

\* @return 返回是否调用成功

\* 0 - 成功

\* !=0 - 失败

\*/

int GetChatData(WeWorkFinanceSdk\_t\* sdk, unsigned long long seq, unsigned int limit, const char \*proxy,const char\* passwd,int timeout,Slice\_t\* chatDatas);

**请求参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| sdk | 是 | 初始化的sdk对象 |
| seq | 是 | 本次请求获取消息记录开始的seq值。首次访问填写0，非首次使用上次企业微信返回的最大seq。允许从任意seq重入拉取。Uint64类型，范围0-pow(2,64)-1 |
| limit | 是 | 一次调用限制的limit值，不能超过1000.uint32类型 |
| proxy | 否 | 使用代理的请求，需要传入代理的链接。如：socks5://10.0.0.1:8081 或者 [http://10.0.0.1:8081.如不使用代理可以设置为空. 支持sock5跟http](http://10.0.0.1:8081.如不使用代理可以设置为空.%20支持sock5跟http)代理 |
| paswd | 否 | 代理账号密码，需要传入代理的账号密码。如 user\_name:passwd\_123 |
| chatDatas | 是 | 返回本次拉取消息的数据.密文消息 |

注：获取聊天记录内容不能超过3天，如果企业需要全量数据，则企业需要定期拉取聊天消息。

**ChatDatas详解：**

{"errcode":0,"errmsg":"ok","chatdata":[{"seq":196,"msgid":"CAQQ2fbb4QUY0On2rYSAgAMgip/yzgs=","publickey\_ver":3,"encrypt\_random\_key":"ftJ+uz3n/z1DsxlkwxNgE+mL38H42/KCvN8T60gbbtPD+Rta1hKTuQPzUzO6Hzne97MgKs7FfdDxDck/v8cDT6gUVjA2tZ/M7euSD0L66opJ/IUeBtpAtvgVSD5qhlaQjvfKJc/zPMGNK2xCLFYqwmQBZXbNT7uA69Fflm512nZKW/piK2RKdYJhRyvQnA1ISxK097sp9WlEgDg250fM5tgwMjujdzr7ehK6gtVBUFldNSJS7ndtIf6aSBfaLktZgwHZ57ONewWq8GJe7WwQf1hwcDbCh7YMG8nsweEwhDfUz+u8rz9an+0lgrYMZFRHnmzjgmLwrR7B/32Qxqd79A==","encrypt\_chat\_msg":"898WSfGMnIeytTsea7Rc0WsOocs0bIAerF6de0v2cFwqo9uOxrW9wYe5rCjCHHH5bDrNvLxBE/xOoFfcwOTYX0HQxTJaH0ES9OHDZ61p8gcbfGdJKnq2UU4tAEgGb8H+Q9n8syRXIjaI3KuVCqGIi4QGHFmxWenPFfjF/vRuPd0EpzUNwmqfUxLBWLpGhv+dLnqiEOBW41Zdc0OO0St6E+JeIeHlRZAR+E13Isv9eS09xNbF0qQXWIyNUi+ucLr5VuZnPGXBrSfvwX8f0QebTwpy1tT2zvQiMM2MBugKH6NuMzzuvEsXeD+6+3VRqL"}]}

**返回参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 0表示成功，错误返回非0错误码，需要参看errmsg。Uint32类型 |
| errmsg | 返回信息，如非空为错误原因。String类型 |
| chatdata | 聊天记录数据内容。数组类型。包括seq、msgid等内容 |
| seq | 消息的seq值，标识消息的序号。再次拉取需要带上上次回包中最大的seq。Uint64类型，范围0-pow(2,64)-1 |
| msgid | 消息id，消息的唯一标识，企业可以使用此字段进行消息去重。String类型 |
| publickey\_ver | 加密此条消息使用的公钥版本号。Uint32类型 |
| encrypt\_random\_key | 使用publickey\_ver指定版本的公钥进行非对称加密后base64加密的内容，需要业务方先base64 decode处理后，再使用指定版本的私钥进行解密，得出内容。String类型 |
| encrypt\_chat\_msg | 消息密文。需要业务方使用将encrypt\_random\_key解密得到的内容，与encrypt\_chat\_msg，传入sdk接口DecryptData,得到消息明文。String类型 |

encrypt\_random\_key内容解密说明：

encrypt\_random\_key是使用企业在管理端填写的公钥（使用模值为2048bit的秘钥），采用RSA加密算法进行加密处理后base64 encode的内容，加密内容为企业微信产生。RSA使用PKCS1。

企业得到消息内容后，需先进行base64 decode，使用消息指明版本的私钥，使用RSA PKCS1算法进行解密，得到解密内容，为下一步进行消息明文解析做准备。

**SDK解密接口：**

/\*\*

\* @brief 解析密文.企业微信自有解密内容

\* @param [in] encrypt\_key, getchatdata返回的encrypt\_random\_key,使用企业自持对应版本秘钥RSA解密后的内容

\* @param [in] encrypt\_msg, getchatdata返回的encrypt\_chat\_msg

\* @param [out] msg, 解密的消息明文

\* @return 返回是否调用成功

\* 0 - 成功

\* !=0 - 失败

\*/

int DecryptData(const char\* encrypt\_key, const char\* encrypt\_msg, Slice\_t\* msg);

请求参数说明：

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| encrypt\_key | 是 | 企业私钥解密encrypt\_random\_key后的内容 |
| encrypt\_msg | 是 | GetChatdata接口返回的加密消息encrypt\_chat\_msg |
| msg | 是 | 消息明文 |

**消息格式说明：**

本小节对解析出的消息格式进行说明。消息类型支持：文本、图片、语音、视频、位置等多种消息类型。

文本消息示例：

1. {"msgid":"CAQQluDa4QUY0On2rYSAgAMgzPrShAE=","action":"send","from":"XuJinSheng","tolist":["icefog"],"roomid":"","msgtime":1547087894783,"msgtype":"text","text":{"content":"666"}}

| **参数** | **说明** |
| --- | --- |
| msgid | 消息id，消息的唯一标识，企业可以使用此字段进行消息去重。String类型 |
| action | 消息动作，撤回消息是为recall，其他为send。String类型 |
| from | 消息发送方id,userid。String类型 |
| tolist | 消息接收方列表，可能是多个，同一个企业内容未userid，非相同企业为openid。数组，内容为string类型 |
| roomid | 群聊消息的群id。String类型 |
| msgtime | 消息发送时间戳，utc时间，ms单位。 |

|  |  |
| --- | --- |
| mstype | 文本消息为：text。String类型 |
| content | 消息内容。String类型 |

图片消息示例：

{"msgid":"CAQQvPnc4QUY0On2rYSAgAMgooLa0Q8=","action":"send","from":"XuJinSheng","tolist":["icefog"],"roomid":"","msgtime":0,"msgtype":"image","image":{"md5sum":"50de8e5ae8ffe4f1df7a93841f71993a","filesize":70961,"sdkfileid":"CtYBMzA2OTAyMDEwMjA0NjIzMDYwMDIwMTAwMDIwNGI3ZmU0MDZlMDIwMzBmNTliMTAyMDQ1YzliNTQ3NzAyMDQ1YzM3M2NiYzA0MjQ2NjM0MzgzNTM0NjEzNTY1MmQzNDYxMzQzODJkMzQzMTYxNjEyZDM5NjEzOTM2MmQ2MTM2NjQ2NDY0NjUzMDY2NjE2NjM1MzcwMjAxMDAwMjAzMDExNTQwMDQxMDUwZGU4ZTVhZThmZmU0ZjFkZjdhOTM4NDFmNzE5OTNhMDIwMTAyMDIwMTAwMDQwMBI4TkRkZk1UWTRPRGcxTVRBek1ETXlORFF6TWw4eE9UUTVOamN6TkRZMlh6RTFORGN4TWpNNU1ERT0aIGEwNGQwYWUyM2JlYzQ3NzQ5MjZhNWZjMjk0ZTEyNTkz"}}

|  |  |
| --- | --- |
| mstype | 图片消息为：image。String类型 |
| sdkfileid | 媒体资源的id信息。String类型 |
| md5sum | 图片资源的md5值，供进行校验。String类型 |
| filesize | 图片资源的文件大小。Uint32类型 |

注：以下消息类型不再描述msgid、action、from、tolist、roomid、msgtime等通用字段，着重描述msgtype以及不同消息类型对应的json格式的消息内容。

撤回消息类型：

|  |  |
| --- | --- |
| mstype | 撤回消息为：revoke。String类型 |
| pre\_msgid | 标识撤回的原消息的msgid。String类型 |

同意协议消息类型：

|  |  |
| --- | --- |
| mstype | 同意消息为：agree。String类型 |
| userid | 同意协议者的userid，外部企业默认为openid。String类型 |
| agree\_time | 同意协议的时间，utc时间，ms单位。 |

语音类型：

|  |  |
| --- | --- |
| mstype | 语音消息为：voice。String类型 |
| voice\_size | 语音消息大小。Uint32类型 |
| play\_length | 播放长度。Uint32类型 |
| sdkfileid | 媒体资源的id信息。String类型 |
| md5sum | 图片资源的md5值，供进行校验。String类型 |

视频类型：

|  |  |
| --- | --- |
| mstype | 视频消息为：video。String类型 |
| sdkfileid | 媒体资源的id信息。String类型 |
| md5sum | 资源的md5值，供进行校验。String类型 |
| filesize | 资源的文件大小。Uint32类型 |
| play\_length | 视频播放长度。Uint32类型 |

名片类型：

|  |  |
| --- | --- |
| mstype | 名片消息为：card。String类型 |
| corpname | 名片所有者所在的公司名称。String类型 |
| Userid | 名片所有者的id，同一公司是userid，不同公司是openid。String类型 |

地址位置信息类型：

|  |  |
| --- | --- |
| mstype | 位置消息为：location。String类型 |
| longitude | 精度，单位double |
| latitude | 维度，单位double |
| address | 地址信息。String类型 |
| title | 位置信息的title。String类型 |
| zoom | 缩放比例。Uint32类型 |

表情信息类型：

|  |  |
| --- | --- |
| mstype | 表情消息为：emotion。String类型 |
| type | 表情类型，png或者gif.1表示gif 2表示png。Uint32类型 |
| width | 表情图片宽度。Uint32类型 |
| height | 表情图片高度。Uint32类型 |
| sdkfileid | 媒体资源的id信息。String类型 |
| md5sum | 资源的md5值，供进行校验。String类型 |
| imagesize | 资源的文件大小。Uint32类型 |

文件类型：

|  |  |
| --- | --- |
| mstype | 文件消息为：file。String类型 |
| sdkfileid | 媒体资源的id信息。String类型 |
| md5sum | 资源的md5值，供进行校验。String类型 |
| filename | 文件名称。String类型 |
| fileext | 文件类型后缀。String类型 |
| filesize | 文件大小。Uint32类型 |

链接消息：

|  |  |
| --- | --- |
| mstype | 链接消息为：link。String类型 |
| title | 消息标题。String类型 |
| description | 消息描述。String类型 |
| link\_url | 链接url地址。String类型 |
| image\_url | 链接图片url。String类型 |

小程序消息：

|  |  |
| --- | --- |
| mstype | 消息为：weapp。String类型 |
| title | 消息标题。String类型 |
| description | 消息描述。String类型 |
| username | 用户名称。String类型 |
| Displayname | 小程序名称。String类型 |

聊天记录消息：

|  |  |
| --- | --- |
| mstype | 消息为：chatrecord。String类型 |
| title | 聊天记录标题。String类型 |
| item | 消息记录内的消息内容，批量数据 |

聊天消息item定义：

|  |  |
| --- | --- |
| type | 每条聊天记录的具体消息类型：ChatRecordText/ ChatRecordFile/ ChatRecordImage/ ChatRecordVideo/ ChatRecordLink/ ChatRecordLocation …. |
| msgtime | 消息时间，utc时间，单位是秒。 |
| content | 消息内容。Json串，内容为对应类型的json。String类型 |
| from\_chatroom | 是否来自群会话。Bool类型 |

消息容量评估：以平均群大小评估，每条消息平均大小约1k；最大群消息评估，每条消息约50k

**获取媒体文件：**

针对图片、文件等媒体数据，提供sdk接口拉取数据内容。

/\*\*

\* 拉取媒体消息函数

\* Return值=0表示该API调用成功

\*

\*

\* @param [in] sdk NewSdk返回的sdk指针

\* @param [in] sdkFileid 从GetChatData返回的聊天消息中，媒体消息包括的sdkfileid

\* @param [in] indexbuf 媒体消息分片拉取，需要填入每次拉取的索引信息。首次不需要填写，默认拉取512k，后续每次调用只需要将上次调用返回的outindexbuf填入即可。

\* @param [in] proxy 使用代理的请求，需要传入代理的链接。如：socks5://10.0.0.1:8081 或者 http://10.0.0.1:8081

\* @param [in] passwd 代理账号密码，需要传入代理的账号密码。如 user\_name:passwd\_123

\* @param [in] timeout 超时时间，单位秒

\* @param [out] media\_data 返回本次拉取的媒体数据.MediaData结构体.内容包括data(数据内容)/outindexbuf(下次索引)/is\_finish(拉取完成标记)

\* @param [out] outindexbuf 返回下次访问需要的索引信息

\* @param [out] data 返回本次访问返回的媒体数据内容，解密后内容，如分片数据需累加到文件存储.单次最大返回512K字节

\* @param [out] is\_finish 标识是否全部拉取完成，1完成、0未完成

\*

\* @return 返回是否调用成功

\* 0 - 成功

\* !=0 - 失败

\*/

int GetMediaData(WeWorkFinanceSdk\_t\* sdk, const char\* indexbuf,

const char\* sdkFileid,const char \*proxy,const char\* passwd, int timeout,MediaData\_t\* media\_data);

**请求参数说明：**

| **参数** | **说明** |
| --- | --- |
| Sdk | 初始化的sdk对象 |
| Indexbuf | 媒体消息分片拉取，需要填入每次拉取的索引信息。首次不需要填写，默认拉取512k，后续每次调用只需要将上次调用返回的outindexbuf填入即可。 |
| sdkfileid | 消息体内容中的sdkfileid信息。 |
| Media\_data | 返回的媒体数据，MediaData\_t结构体 |

**工具接口：**

/\*\*

\* @brief 释放sdk，和NewSdk成对使用

\* @return

\*/

void DestroySdk(WeWorkFinanceSdk\_t\* sdk);

Slice\_t\* NewSlice();

/\*\*

\* @brief 释放slice，和NewSlice成对使用

\* @return

\*/

void FreeSlice(Slice\_t\* slice);

/\*\*

\* @brief 为其他语言提供读取接口

\* @return 返回buf指针

\* !=NULL - 成功

\* NULL - 失败

\*/

char\* GetContentFromSlice(Slice\_t\* slice);

// 媒体记录相关工具

MediaData\_t\* NewMediaData();

void FreeMediaData(MediaData\_t\* media\_data);

char\* GetOutIndexBuf(MediaData\_t\* media\_data);

char\* GetData(MediaData\_t\* media\_data);

int IsMediaDataFinish(MediaData\_t\* media\_data);