

德索业务管理平台影像归档对接说明

业务说明：将影像资料归档（存放）到客户的账号下。

使用步骤：

1、用 Postman 导入接口定义文件（postman 是一个 Http 接口调试工具，[官方网址](#)）



德索业务管理平台
对接-影像资料.post

导入后可以看到有编号为 1-9 的 9 个接口：

The screenshot shows the Postman interface with a list of 9 API endpoints under the collection '德索业务管理平台对接-影像资料'. The endpoints are:

- POST 1、登录接口
- POST 2、账户影像资料归档
- POST 3、废弃归档
- POST 4、生成临时账户接口
- POST 5、文件上传接口
- POST 6、客户影像资料归档
- POST 7、查询账户影像参数
- GET 8、下载or预览文件
- POST 9、分页查客户信息

2、修改调用地址

几个接口调用地址根据各家情况自行替换。将协议、ip、端口部分替换为和业务管理系统一致的。

The screenshot shows a POST request in Postman. The URL field contains 'http://180.169.30.6:10229/crm/login/jwt'. A red underline is placed under the URL, and the text '将 协议、ip、端口 部分替换成和'业务管理平台'一样' is written in red at the bottom right.

3、接口调用方法：

下面就一些常用的使用场景分开说明：

◆ 归档影像资料场景：

- 先调用**接口 1**进行登录，从 http header 里面获取到 Authorization，其他接口调用的时候都会要用到它，Authorization 的有效时间默认是 1 小时。**注意：**当请求是 https 协议时，最好跳过证书的检测。因为大部分期货公司用的证书可能是自签名的。
- 可选。**根据情况，看是否需要调用**接口 3**新增账号。因为有的业务类型的账号是系统里就有的（通常存量的经纪账户、居间账户是内置好的）。
- 可选。**根据情况，看是否需要调用**接口 5**单独上传文件。有的文件体积比较大不适合使用 base64 编码来上

传。

- d、调用接口 2 进行账户的影像归档。注意 URL 中有一个参数，他是业务管理平台为调用者颁发的系统编号。第三方系统进行对接的话，联系我们颁发新编号即可 (查询已有编号 sql: `select t.* from ldm.oauth2_registered_client t`)。



e、可选。接口 3 是用作废弃接口 2 或接口 6 归档进来的资料，可以理解为是对接口 2 或接口 6 的撤销。

f、可选。调用接口 6 进行客户的影像归档。

◆ 查询下载单个影像资料场景：

- a) 先调用接口 1 进行登录。具体见上面。
- b) 调用接口 7 查询单个账户下面的影像 (主要是获取 fileId)。
- c) 调用接口 8 下载文件 (通过 fileId 获取真实资料文件)。

◆ 查询下载多个影像资料场景：

- a) 先调用接口 1 进行登录。具体见上面。
- b) 调用接口 9 分页查询出满足条件的客户。
- c) 调用接口 7 查询单个账户下面的影像 (主要是获取 fileId)。
- d) 调用接口 8 下载文件 (通过 fileId 获取真实资料文件)。

4、接口中需要说明的字段

(其中有字段取值码表值，下面码表查询方法示例：

```
select t.* from ldm.adm$cde_code t where t.comm_cd = 'ACCT_BIZ_TYP'; )
```

接口 编号	字段	名称	取值说明
1	username	crm 账号	
1	password	crm 账号的密码	
2	title	本次导入的标题或者概述	
2	businessId	本次导入的业务 id 或者是流程编号等,主要用来溯源	
2	accountBizType	客户账号业务类型	码表值： ACCT_BIZ_TYP
2	accountNo	客户账号号码	
2	createTime	归档时间。可选，默认当前时间。格式： yyyy-mm-dd hh:mi:ss	

2	extras	附加信息。主要用来备查问题	
2	materials.code	资料编号, 不能为空	业务管理平台中的影像参数代码。
2	materials.base64File.name	要归档的文件名称, 与 fileId 二者传一即可	
2	materials.base64File.base64Data	要归档文件的 base64 数据	
2	materials.fileId	要归档的文件 Id, 与 base64File 二者传一即可	
	<pre>public class ServiceAccess { @Schema(description = "数据接入 Id") private String id; @Schema(description = "主题") private String title; @Schema(description = "业务 id 或者是流程编号等,主要用来溯源") private String businessId; @Schema(description = "客户的账户 id。给账户归档时用到") private String accountId; @Schema(description = "客户 id。给客户归档时用到") private String customerId; @Schema(description = "客户的账号业务类型") private String accountBizType; @Schema(description = "导入数据服务的 id") private String serviceId; @Schema(description = "导入资料列表") private String materialCodes; @Schema(description = "附加信息。主要用来备查问题") private String extras; @Schema(description = "导入资料的数量") private Long materialCount; @Schema(description = "创建时间") private Date createTime; @Schema(description = "归档是否被废弃") private boolean discarded; @Schema(description = "废弃") private Date discardedTime; @Schema(description = "后续业务 id 或者是流程编号等") private String postBusinessId; @Schema(description = "后续业务信息") private String postMsg; }</pre>		
3	serviceAccessId	归档 id	调用归档接口（接口 2）归档时候返回的 id
3	discardEmpCode	进行废弃操作的员工	

3	discardReason	废弃原因不能为空	
			无出参
4	deptCode	账号的所属部门	
4	accountNo	账号号码	
4	accountBizType	账号业务类型	码表值: ACCT_BIZ_TYP
4	accountName	账号所有者名称 (客户名称)	
4	accountCustomerType	账号级的客户身份类型	码表值: ACCT_CUST_TYP
4	customerType	客户类型	0 自然人, 1 非自然人
4	accountStatus	账号状态	0 销户, 1 有效
4	idType	客户证件类型	码表值: ID_TYP
4	idNo	客户证件号码	
4	extraCode	客户附加码	经纪业务 (也就是 accountBizType=0) 时才需要填写。
	<pre>public class CustomerAccountResponse { @Schema(description = "客户信息") private Customer customer; @Schema(description = "账户信息") private Account account; @Schema(description = "部门信息") private Department department; }</pre> <pre>public class Customer { @Schema(description = "客户 id") private String customerId; @Schema(description = "客户类型。0 自然人, 1 非自然人") private String customerType; @Schema(description = "客户姓名") private String customerName; @Schema(description = "客户单位全称") private String customerFullName; @Schema(description = "客户部门编号") private String deptCode; @Schema(description = "客户部门编号") private String oriDeptCode; @Schema(description = "客户证件类型。码表值: ID_TYP") private String idType; @Schema(description = "客户证件号码") private String idNo; @Schema(description = "客户组织机构代码附加码") private String extraCode;</pre>		出参

```
@Schema(description = "客户税务登记号")
private String raxRegNo;
@Schema(description = "客户营业执照号")
private String licenseNo;
@Schema(description = "客户来源类型。默认值 1")
private String entityType = "1";
@Schema(description = "客户创建日期。格式 yyyy-MM-dd")
private String createDate;
@Schema(description = "客户创建时间。格式 h24:mm:mi")
private String createTime;
@Schema(description = "数据添加系统的标识")
private String srcType;
@Schema(description = "原 id,修改时用")
protected String originalCustomerId;
@Schema(description = "客户洗钱风险等级")
private String amlRiskCode;
@Schema(description = "德索客户洗钱风险等级")
private String amlRiskCodeName;
@Schema(description = "所有证件")
private List<CustomerIdNoAll> idNos;
}
```

```
public class Account {
    @Schema(description = "客户 id")
    protected String customerId;
    @Schema(description = "账号 id")
    protected String accountId;
    @Schema(description = "账号号码")
    protected String accountNo;
    @Schema(description = "账号业务类型。码表值：ACCT_BIZ_TYP")
    protected String accountBizType;
    @Schema(description = "账号状态。0 销户，1 有效")
    protected String accountStatus;
    @Schema(description = "账号所有者名称")
    protected String accountName;
    @Schema(description = "账号所有者全称")
    protected String accountFullName;
    @Schema(description = "账号级的客户身份类型。码表值：
ACCT_CUST_TYP")
    protected String accountCustomerType;
    @Schema(description = "账号所属部门")
    protected String deptCode;
    @Schema(description = "账号开户日期。格式 yyyy-MM-dd")
    protected String openAcctDate;
```

	<pre> @Schema(description = "账号注销日期。格式 yyyy-MM-dd") protected String writeoffDate; @Schema(description = "账号来源, srcType=counter 时表示数据来源与柜台, 即正式账号") private String srcType; @Schema(description = "账号类型") private AccountCategory accountCategory; @Schema(description = "投资者标签") private List<AccountLabel> accountLabelList; } public class Department { @Schema(description = "部门编号") protected String deptCode; @Schema(description = "部门名称") protected String deptName; @Schema(description = "父部门编号") protected String parentDeptCode; } </pre>	
--	--	--

5	createTime	创建时间。可选, 默认当前时间。格式: yyyy-mm-dd hh:mi:ss	
	<pre> public class File { @Schema(description = "id") private String id; @Schema(description = "文件大小") private Long fileSize; @Schema(description = "文件扩展名") private String fileExtName; @Schema(description = "文件的 http content-type") private String fileContentType; @Schema(description = "文件的原始名称") private String fileOriginalName; @Schema(description = "文件 chunk 的关键字") private String fileChunkKey; @Schema(description = "上传人的员工编号") private String empCode; @Schema(description = "多图上传的时候的原图列表") private String imageSourceFileChunkKeys; @Schema(description = "压缩原图") private String compressSourceFileChunkKey; @Schema(description = "是否为临时文件。临时文件会在指定的之间之后进行清除") } </pre>	出参	

	<pre> private boolean isTemp; @Schema(description = "失效时间。为临时文件时的失效时间") private Date tempExpireTime; @Schema(description = "创建时间") private Date createTime; @Schema(description = "修改时间") private Date updateTime; } </pre>	
--	---	--

6	idType	客户证件类型	码表值: ID_TYP
6	idNo	客户证件号码	
6	extraCode	客户附加码	经纪业务（也就是 accountBizType=0）时才需要填写。
6	title	本次导入的标题或者概述	
6	businessId	本次导入的业务 id 或者是流程编号等,主要用来溯源	
6	createTime	归档时间。可选, 默认当前时间。格式: yyyy-mm-dd hh:mi:ss	
6	extras	附加信息。主要用来备查问题	
6	materials.code	资料编号, 不能为空	业务管理平台中的影像参数代码。
6	materials.base64File.name	要归档的文件名称, 与 fileId 二者传一即可	
6	materials.base64File.base64Data	要归档文件的 base64 数据	
6	materials.fileId	要归档的文件 Id, 与 base64File 二者传一即可	
	同 2 号接口		出参

7	accountNo	账号号码	
7	accountBizType	账号业务类型	码表值: ACCT_BIZ_TYP
7	withCustomer	是否携带客户的影像	
7	withHis	是否返回历史的影像	
7	materialCodes	指定返回哪些影像参数的影像。null 时为所有。	
	<pre> public class MaterialInfo extends MaterialAccess { @Schema(description = "id") private Long id; @Schema(description = "客户的账户 id") private String accountId; @Schema(description = "客户的账户号码") private String accountNo; } </pre>		出参

```
@Schema(description = "客户的账号业务类型")
private String accountBizType;
@Schema(description = "客户账号的部门编号")
private String accountDeptCode;
@Schema(description = "资料类型编码")
private String materialCode;
@Schema(description = "资料类型名称")
private String materialName;
@Schema(description = "业务类型")
private String bizCode;
@Schema(description = "业务类型名称")
private String bizName;
@Schema(description = "提交资料员工编号")
private String submitterEmpCode;
@Schema(description = "提交资料员工名称")
private String submitterEmpName;
@Schema(description = "资料文件 id 列表")
private List<String> fileIds;
@Schema(description = "资料文件列表")
private List<File> files;
@Schema(description = "资料来源")
private SourceType sourceType;
@Schema(description = "外部服务的导入 id")
private String serviceAccessId;
@Schema(description = "接入服务的名称")
private String serviceName;
@Schema(description = "接入服务的标题")
private String serviceAccessTitle;
@Schema(description = "流程实例 id")
private String processInstanceId;
@Schema(description = "流程名称")
private String processName;
@Schema(description = "创建时间")
private Date createTime;
@Schema(description = "客户的账户 id")
private String customerId;
@Schema(description = "历史创建时间")
private Date hisCreateTime;
}

public class File { // 同 5 号接口 }
```

8	{fileId}	请求的 url 中传递 fileId。接口 5 上传文件后和接口 7 查询接口都会返回 fileId。一个 fileId 表示一个物理文件。	示例： http://180.169.30.6:10229/crm/dms/files/ 40287e8c8d1f558f018d1f5e16940041 /file?previe p=true 其中的 40287e8c8d1f558f018d1f5e16940041 就是 fileId
8	preview	是否以预览方式打开。可选，在浏览器中使用时的动作。	
文件二进制流			出参
9	materialCodes	查询包含该影像资料的账号。多选时满足其一就算命中，可选。	
9	excludedMaterialCodes	查询不包含该影像资料的账号。多选时全部满足才算命中，可选。	
9	openAcctDateRange.start	开户时间范围起点，可选	
9	openAcctDateRange.end	开户时间范围终点，可选	
9	deptCodes	指定营业部，可选。	在查询者的部门权限范围内进一步细化
9	accountBizTypes	账号业务类型，可选。	码表值：ACCT_BIZ_TYP
9	accountCustomerTypes	账号级的客户身份类型	码表值：ACCT_CUST_TYP
9	pageNumber	分页查询，页码。	
9	pageSize	分页查询，页大小。	
	<pre>public class PageResponse<T> { @Schema(description = "当前页码") private int number; @Schema(description = "分页大小") private int size; @Schema(description = "总页数") private int totalPages; @Schema(description = "本次分页查询返回数据条数") private int numberElements; @Schema(description = "总数据条数") private long totalElements; @Schema(description = "是否由上一页") private boolean previousPage; @Schema(description = "是否是首页") private boolean first; @Schema(description = "是否有下一页") private boolean nextPage; @Schema(description = "是否是末页") private boolean last; }</pre>		

```
private boolean last;  
@Schema(description = "本次分页查询的数据内容")  
private List<T> content = Collections.emptyList();  
}  
此处泛型 T 为 CustomerAccountResponse，具体见 4 号接口。
```

5、接口返回说明

http code 为 200-299 内都是成功，其他都是失败。

The screenshot shows a POST request to `http://180.169.30.6:10229/crm/login/jwt`. The request body is a JSON object with fields `username` and `password`. The response status is 200 OK, indicating success.

Request Details:

- Method: POST
- URL: `http://180.169.30.6:10229/crm/login/jwt`
- Body (JSON):

```
1 {  
2   "username": "mt1",  
3   "password": "1111111111112312312312312"  
4 }
```

Response Status:

200 OK 120 ms 1015 B

POST http://180.169.30.6:10229/crm/login/jwt

Params Authorization Headers (11) Body **JSON** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Beautify

```
1 {  
2   "username": "mt1",  
3   "password": "1111111111112312312312312312312"  
4 }
```

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize **JSON** Save Response

401 Unauthorized 59 ms 395 B

1 Invalid Credentials