OPEN AI LAB

鉴权用户开发手册

2018-10-17

OPEN AI LAB

变更记录
(Reversion Record)

日期 (Date)	版本 (Rev)	说明 (Change Description)	作者 (Author)
0.1.0	姚宏贵	初稿	2018.09.28
0.2.0	郭汇江	增加authid说明	2018.10.8
0.3.0	姚宏贵	增加网页操作图样	2018.10.17

目录(**catalog**)

1 前言

1.1 目的

用于指导用户开发使用 OpenAILab 的 SDK，共需要的人参考

1.2 术语

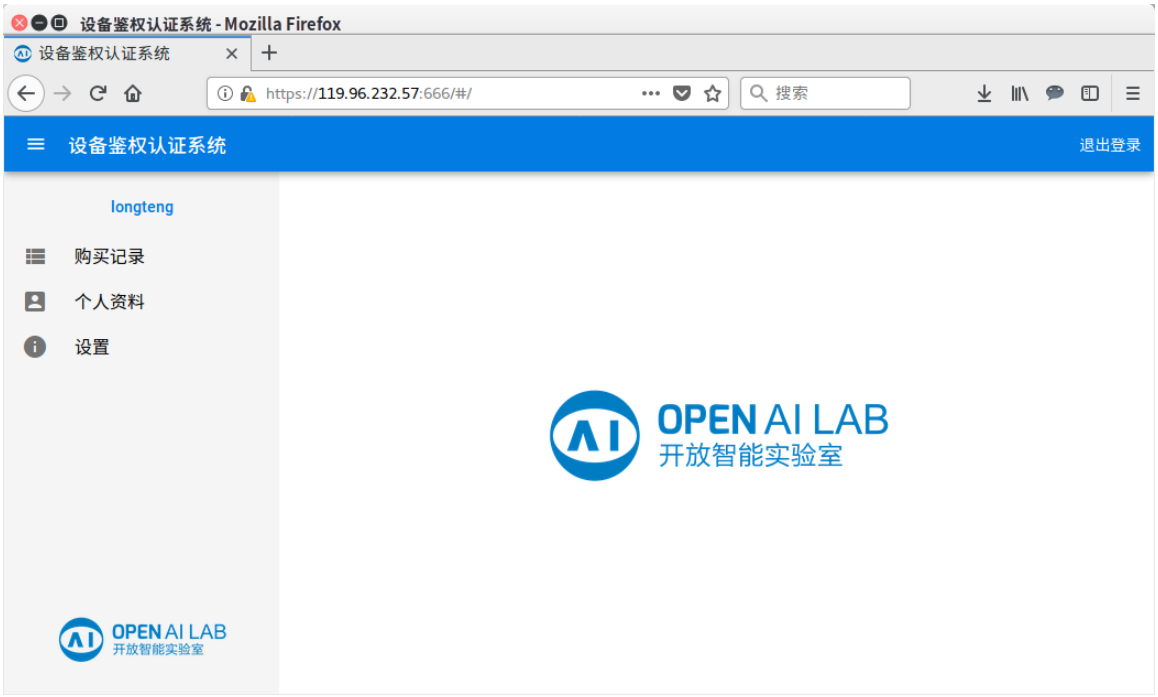
- **设备**：分为联网设备和非联网设备
- **联网设备**：设备可以与云端的鉴权服务器直接联系上。
- **非联网设备**：设备不能自己直接联系鉴权服务器，需要借助 PC 来间接和鉴权服务器通信。
- **合同号**：用户与 OpenAILab 签订合同，得到合同号。
- **用户授权密码**：用户设备去服务器请求授权时需要提供密码。

2 网页功能简述

- 客户登陆界面

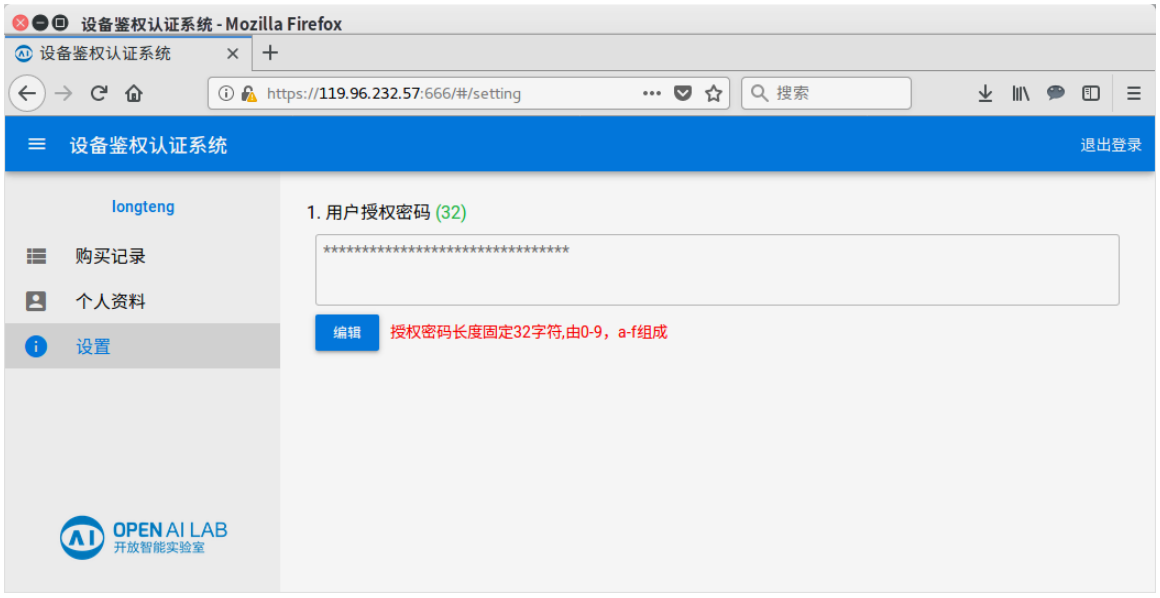
  在和 Open AI Lab 签订协议后，会得到用户名及密码。然后在此登陆

- 客户功能首页面

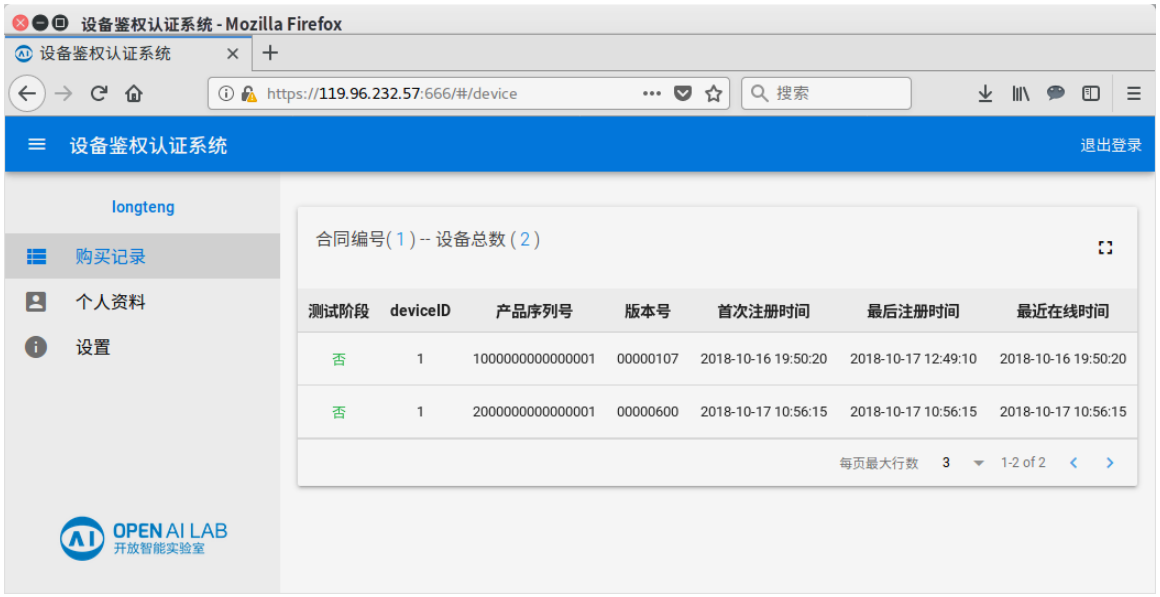


- 客户账号管理页面

1) 用户上传“用户授权密码”Password（16bytes，用户采用 32 个 hex 字符（0~9a~f）输入，为了防止别人盗用客户授权。按合同不同而不同。用户需要记住此 Password（服务器端只是保持了此密码的 MD5 校验码），一旦丢失，，没法找回，只能重新上传新的 Password）

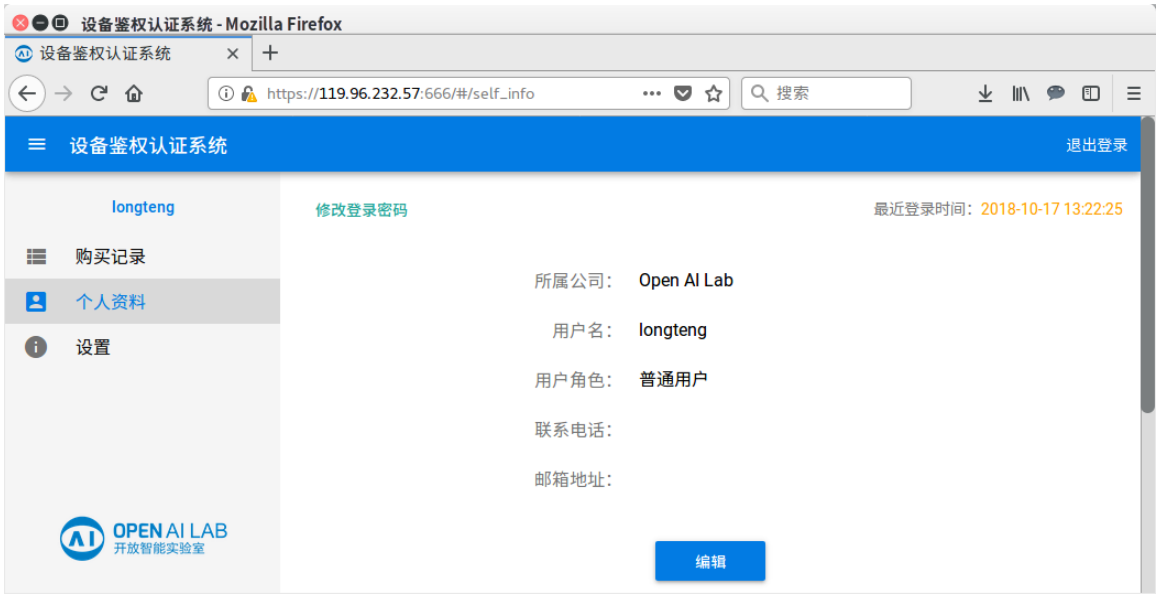


2) 查看用户公司的设备使用情况



3) 查看用户公司的合同信息

4) 用户的个人资料



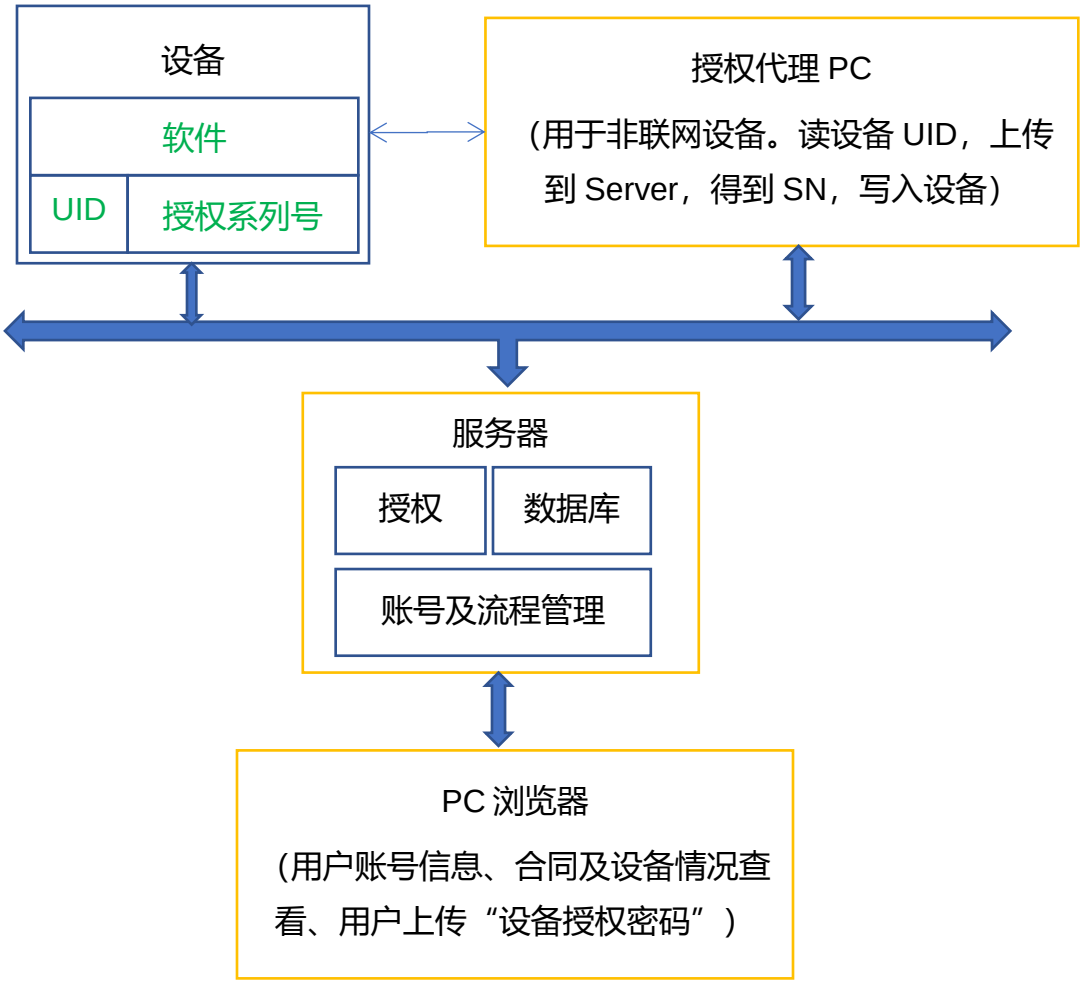
3 客户设备授权全流程

1) 客户与 Open AI Lab 签订合同
2) 客户付款
3) Open AI Lab 给客户分配 “授权号” （OEMID<16 个 hex 字符>，ContractIDs<4 个 OpenAILab 分配的 hex 字符>），供客户写入设备。此外用户需要提供设备唯一 ID （DevID < 1 2 个 hex 字符，表示设备唯一 ID，比如 MAC 地址>），
   - 比如 OEMID 为 3a017fc58f506db9
   - 比如 ContractID 为 0001
   - 比如 DevID 为 75871234db97

   则授权号（authID）为 3a017fc58f506db90001 75871234db97
4) 用户上传 “用户授权密码” Password（16bytes，为了防止别人盗用客户授权）到服务器
5) 用户在拿到 OpenAILab 的 SDK 后，集成到自己的产品中时
   - 实现 SN 的读写函数
   - 使用 “授权号” 及 “用户授权密码” 来初始化 SDK，比如 oaid_init(authID, Password)。

- 对于联网设备，SDK 会自动去服务器鉴权，得到 SN。非联网设备，用户需要自己去用 PC 来读取设备的 UID，去服务器得到 SN，然后写入设备。

4 鉴权服务器与其它设备关系图



5 设备端 API

1) oaid_init

设置与本设备相关联的合同 ID 及 Password。由用户初始化 oaid 时调用。本函数由 OpenAILab 实现。

// authID – 授权号，输入参数

// password – 用户授权密码，输入参数

// return – 0 for ok, else for error
int oaid_init(uint8_t authID[16], uint8_t password [16]);

2) SaveSerialNumber

将得到的 SN 保存到本设备。由用户实现

//sn – 软件系列号，输入参数

// return – 0 for ok, else for error
Int SaveSerialNumber(uint8_t sn[128]);

3) ReadSerialNumber

读取保存在本设备的 SN。由用户实现

//sn – 软件系列号，输出参数

// return – 0 for ok, else for error
int ReadSerialNumber(uint8_t sn[128]);

4) GenUID

得到设备的 UID，**本函数由 OpenAILab 实现**。如果用户想自己**代理服务**发 UID 去我们的服务器得到 SN，可以调用此函数来得到 UID。

// 生成设备的 UID。

// UID 里包括合同 ID、软件 ID、硬件 ID 等等。此函数由我们来实现（发布给每个厂家的各个 SDK 里的实现，都可能会有小的区别）。

// authID – 授权号，输入参数

// password – 用户授权密码，输入参数

// uid – 设备 UID， 输出参数

```
// return – 0 for ok, else for error
int genUID(uint8_t authID[16], uint8_t password[16], uint8_t uid[156]);
```

5) GetSNwithUID

由用户实现。当用户采用**代理服务**得到设备的 SN 时，可以实现这个函数。发送 UID 到服务器，得到 SN。无代理服务器时，可以直接返回-1。

```
// uid – 设备 UID， 输入参数。由 GenUID 产生
//sn – 软件系列号，输出参数
// return – 0 for ok, else for error
Int GetSNwithUID(uint8_t uid[156]，uint8_t sn[128]);
```

此函数采用下面接口与服务器通讯获取系列号。

设备或代理 PC 与服务器之间的“注册”接口描述如下：

1） 接口地址：http://ipaddr:port/getDevSn

输入参数

参数名称	必选	类型	描述
devId	是	string	设备 UID

输出参数

参数名称	类型	描述
data	“sn 码”	包含 sn 等信息
errCode	Int	错误码，默认为 0，1 为警告信息，2 表示运行出错
msg	String	警告信息内容

6 设备端使用 SDK 的示例代码

1) 联网设备直接去 OpanAILab 服务器鉴权

```
int GetSNwithUID(uint8_t uid[156]，uint8_t sn[128]){
   return -1;
}
int main（）
```

```
{
    const uint8_t[8] OEMID = {0x3a, 0x01, 0x7f, 0xc5, 0x8f, 0x50, 0x6d, 0xb9};
    const uint8_t[8] ContractID= {0x00, 0x01};
    const uint8_t[8] DevID= {0x75, 0x87, 0x12, 0x34, 0xdb, 0x97}; //比如设为 MAC 地址
    uint8_t[16] authID;
    memcpy(authID, OEMID, 8);
    memcpy(&authID[8], ContractID, 2);
    memcpy(authID, DevID, 6);
    const uint8_t[16] password = {……};
    if(oaid_init(authID, password)){
        return ERROR;
    }
    //init is ok
    //……

}
```

2) 联网设备通过代理服务器鉴权之方法一

```
//这个函数会在适当的地方被自动调用
int GetSNwithUID(uint8_t uid[156]，uint8_t sn[128]){
    //实现代码去连接自己的服务器
    //……
    return 0;
}
int main（）
{
    //同 1 中的定义
    const uint8_t[8] OEMID = {0x3a, 0x01, 0x7f, 0xc5, 0x8f, 0x50, 0x6d, 0xb9};
    const uint8_t[8] ContractID= {0x00, 0x01};
    const uint8_t[8] DevID= {0x75, 0x87, 0x12, 0x34, 0xdb, 0x97}; //比如设为 MAC 地址
    uint8_t[16] authID;
    memcpy(authID, OEMID, 8);
```

```
    memcpy(&authID[8], ContractID, 2);
    memcpy(authID, DevID, 6);
    const uint8_t[16] password = {……};
    //初始化 OAID
    if(oaid_init(authID, password)){
        return ERROR;
    }
    //init is ok
    //……

}
```

3) 联网设备通过代理服务器鉴权之方法二

```
//这个函数会在适当的地方被自动调用
int GetSNwithUID(uint8_t uid[156]，uint8_t sn[128]){
    return -1;
}
int main（）
{
    //同 1 中的定义
    const uint8_t[8] OEMID = {0x3a, 0x01, 0x7f, 0xc5, 0x8f, 0x50, 0x6d, 0xb9};
    const uint8_t[8] ContractID= {0x00, 0x01};
    const uint8_t[8] DevID= {0x75, 0x87, 0x12, 0x34, 0xdb, 0x97}; //比如设为 MAC 地址
    uint8_t[16] authID;
    memcpy(authID, OEMID, 8);
    memcpy(&authID[8], ContractID, 2);
    memcpy(authID, DevID, 6);
    const uint8_t[16] password = {……};
    uint8_t uid[156];
    if(genUID(authID, password,uid)){
        return ERROR;
```

```
    }
    //实现代码去连接自己的服务器，上传 UID，得到 SN
    //……
    //成功得到 SN，则保存到设备
    if(SaveSerialNumber(sn)){
        return ERROR;
    }
    //初始化 OAID
    if(oaid_init(contract_id, password)){
        return ERROR;
    }
    //init is ok
    //……

}
```