# 阿里云
# IoT设备身份认证

## 我是设备厂商

文档版本：20200312

# 法律声明

**阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。**

1. **您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。**
2. **未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。**
3. **由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。**
4. **本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。**
5. **阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。**
6. **如若发现本文档存在任何错误，请与阿里云取得直接联系。**

# 通用约定

| 格式 | 说明 | 样例 |
| --- | --- | --- |
|  | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  **禁止:**<br>重置操作将丢失用户配置数据。 |
|  | 该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  **警告:**<br>重启操作将导致业务中断，恢复业务时间约十分钟。 |
|  | 用于警示信息、补充说明等，是用户必须了解的内容。 |  **注意:**<br>权重设置为0，该服务器不会再接受新请求。 |
|  | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 |  **说明:**<br>您也可以通过按Ctrl + A选中全部文件。 |
| > | 多级菜单递进。 | 单击设置 > 网络 > 设置网络类型。 |
| **粗体** | 表示按键、菜单、页面名称等UI元素。 | 在结果确认页面，单击**确定**。 |
| `Courier字体` | 命令。 | 执行`cd /d C:/window`命令，进入Windows系统文件夹。 |
| *## * | 表示参数、变量。 | `bae log list --instanceid`<br>*`Instance_ID`* |
| []或者[a\|b] | 表示可选项，至多选择一个。 | `ipconfig [-all\|-t]` |
| {}或者{a\|b} | 表示必选项，至多选择一个。 | `switch {active\|stand}` |

# 目录

# 1 服务端对接

本文介绍了用户的云端如何与ID²的服务端对接。

本文档适用于通过其他平台接入的IoT设备对接ID²的场景。如果您通过阿里云物联网平台接入IoT设备，那么请忽略此文档。如果您通过阿里云物联网平台接入IoT设备，那么请忽略此文档。

1. 前提条件

准备好AccessKey。您需要在阿里云账号下生成AccessKey，该账号必须与ID²管理控制台、购买ID²授权的账号保持一致。

> 说明:
> AccessKey生成请参见文档：*AccessKey*生成

您需要下载并集成适合您业务平台的服务端SDK：

- 获取Java SDK：*Java SDK*
- 获取Node JS SDK：*Node JS SDK*

2. 服务端接口对接

**2.1 依赖包安装**

参考SDK目录下aliyun-id2-sample/README，安装依赖包。

**2.2 服务端接口对接**

服务端接口对接请参考服务端*API*手册，您也可以参考/aliyun-id2-sample中的示例代码。

3. 服务端接口验证（可选）

- 由于设备端适配还没有完成，您可以通过调试ID²来模拟设备端，配合完成服务端对接的验证。此处建议您使用调试类ID²进行接口验证。
- 您也可以跳过此步骤，在设备端适配完成后，进行全链路的自主验证。

> 说明:
> 调试类ID²仅限于在“对接调试”阶段使用，不能作为正式的ID²使用。

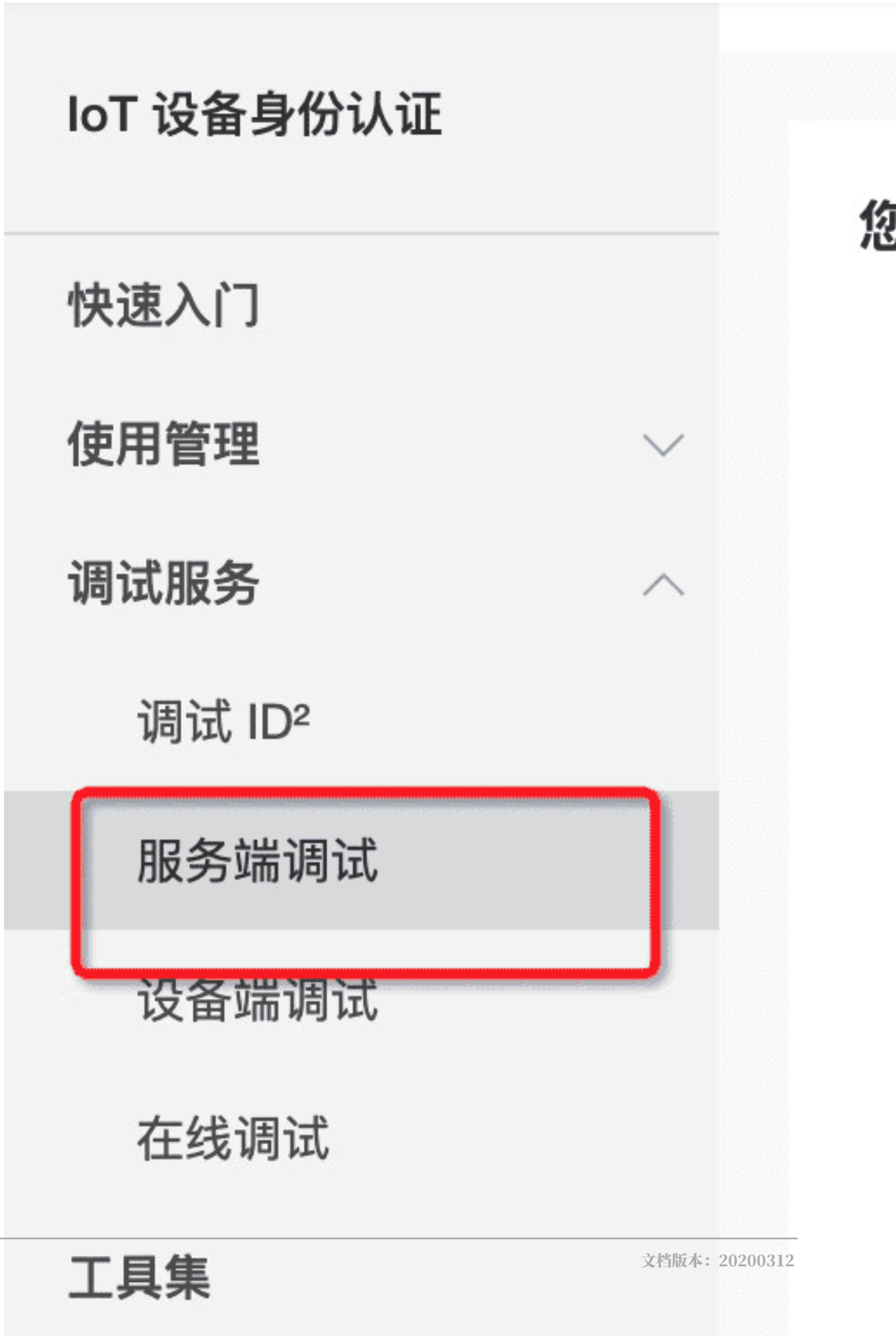
**3.1 生成调试ID²**

登录*IoT*设备身份认证控制台进入“调试服务”-“调试ID²”页面，单击生成调试ID²，获取到的调试ID²如下：


IoT 设备身份认证
调试
快速入门
使用管理
产品管理
ID² 管理
数据报表
调试服务
调试 ID²
服务端调试

### 3.2 开启服务端调试

**进入服务端调试页面**


IoT 设备身份认证
快速入门
使用管理
调试服务
调试 ID²
服务端调试
设备端调试
在线调试
工具集

### 3.3 获取authCode

**服务端调试提供了authCode生成助手，用于模拟设备端生成authCode。您可以此authCode在服务端调用verify接口发起认证请求。**

**说明:**

**调试ID[2]归属在固定产品下，产品的ProductKey在图中所示位置获取。**


阿里云
华东2（上海）
IoT 设备身份认证
快速入门
使用管理
调试服务
调试 ID²
服务端调试
设备端调试
在线调试
工具集
产品文档
服务端调试
您可以通过下列步骤，
1. 服务
您需要
下载
2. 获取
您调试
produ
注意：
3. 服务

### 3.3.1 选择ID²

**在调试ID²列表中选择一个ID²，单击生成。authCode生成成功，单击复制复制authCode。**


阿里云
华东2（上海）
IoT 设备身份认证
服务端调试
快速入门
使用管理
调试服务
调试 ID²
服务端调试
设备端调试
产品文档
您可以通过下列步骤，
服务端
您需要下
下载 J
获取授
您调试用
produ
注意：非
服务端

### 3.3.2 验证

**使用上一步获取的authCode，在服务端调用 verify 接口。接口返回 code 值为 200 表示服务端接口调试成功。其他code值请参考服务端错误码。**

# 2 自主验证

[TOC]

该文档介绍了如何利用”设备端适配验证“完成ID²全链路的调试验证。

您完成ID²在设备端的适配后，可以通过ID²”设备端适配验证工具“验证ID²的设备认证、解密功能。设备端适配验证工具适用于不同的载体（如TEE，SE，MCU），流程如下。


准备工作

- 在设备端上集成安全SDK，立即获取*ID² Client SDK*。
- ID²数据烧录到设备上，如果还没有完成烧录，请查看产线烧录*ID²*到设备-产线烧录工具*java*版。
- 需要用到“ID²认证授权”，如果没有ID²认证授权，请点击立即购买。

1.1 获取设备端调试工具。

1.2 编译生成自主验证的固件（id2_app）

**1.3 自主验证的固件烧录到设备，运行得到调试结果：**

```
============ ID2 Validation Json Message ============
{
        "reportVersion":        "1.0.0",
        "sdkVersion":   "2.0.0",
        "date": "Aug 26 2019 17:02:06",
        "testContent":  [{
                        "api":  "id2_client_get_id",
                        "args": {
                        },
                        "result":       "000FFFFFEA
                }, {
                        "api":  "id2_client_get_cha
                        "args": {
                                "challenge":    "55
                                "extra":        "ab
                        },
                        "result":       "2~2~B0EF39
                }, {
                        "api":  "id2_client_get_time
                        "args": {
                                "timestamp":    "15
                                "extra":        "ab
                        },
                        "result":       "3~2~B0EF39
                }]
}
```

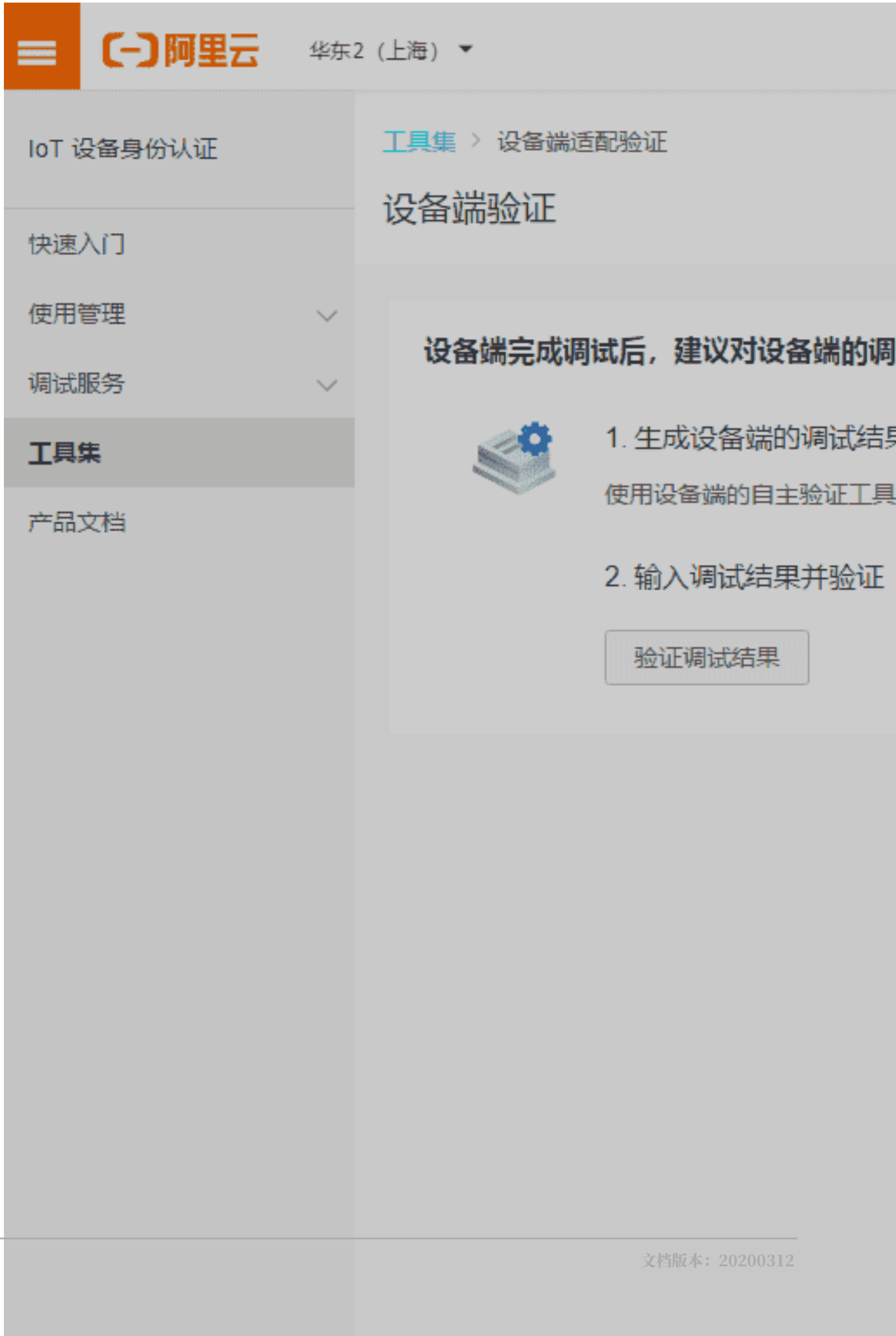
2. 在服务端验证调试结果

**2.1 登录*ID²*控制台，扩展服务-设备端验证，将设备端生成的调试结果黏贴到验证窗口。**

**2.2 进行调试结果的验证**


阿里云
华东2（上海）
IoT 设备身份认证
快速入门
使用管理
调试服务
工具集
产品文档
工具集 > 设备端适配验证
设备端验证
设备端完成调试后，建议对设备端的调
1. 生成设备端的调试结
使用设备端的自主验证工具
2. 输入调试结果并验证
验证调试结果

### 2.3 查看验证结果


阿里云
华东2（上海）
IoT 设备身份认证
快速入门
使用管理
调试服务
工具集
产品文档
工具集 > 设备端适配验证
设备端验证
设备端完成调试后，建议对设备端的调
1. 生成设备端的调试结
使用设备端的自主验证工具
2. 输入调试结果并验证
验证调试结果

3. 在设备端解密

**3.1 获取服务端生成的密文**

**参考步骤2.3中的验证结果，复制密文。**

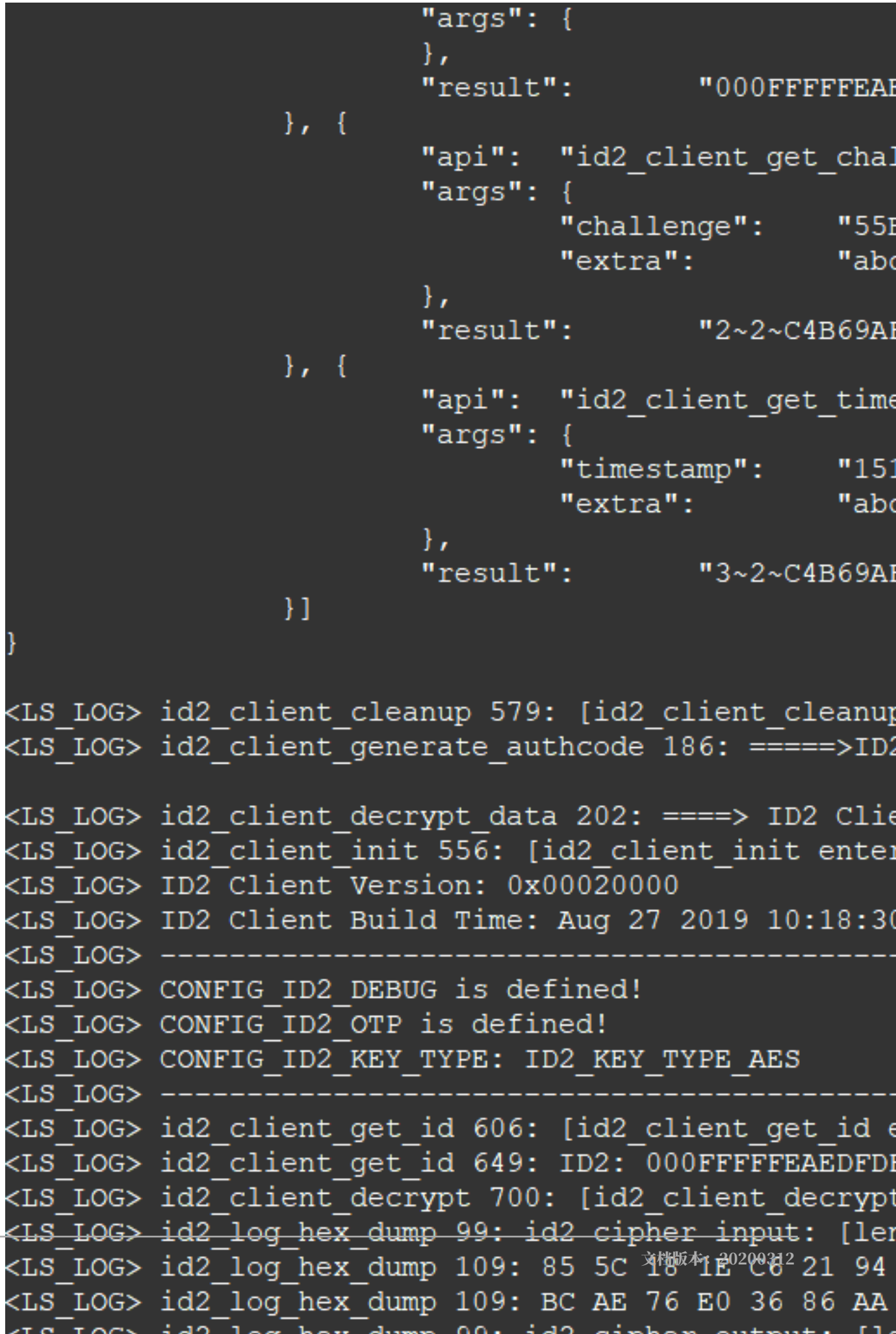

阿里云
华东2（上海）
IoT 设备身份认证
快速入门
使用管理
调试服务
工具集
产品文档
工具集 > 设备端适配验证
设备端验证
设备端完成调试后，建议对设备端的调
1. 生成设备端的调试结
使用设备端的自主验证工具
2. 输入调试结果并验证
验证调试结果

### 3.2 在设备端导入密文

```c
/*
 * Copyright (C) 2017-2019 Alibaba Group Holding L
 */

#include "id2_test.h"

/* Hex String, getting from id2 console */
#define ID2_CIPHER_DATA     "34632fec4a366aa384579

int main(int argc, char *argv[])
{
    int ret;
    uint32_t cipher_len = 0;
    char *cipher_data = ID2_CIPHER_DATA;

    if (argc >= 2) {
        if (!strcmp(argv[1], "-set_data")) {
            cipher_data = argv[2];
        }
    }

    ret = id2_client_unit_test();
    if (ret < 0) {
        ID2_DBG_LOG("id2 client unit test fail!!\n
        return -1;
    }

    cipher_len = strlen(cipher_data);

    ret = id2_client_generate_authcode();
    if (ret < 0) {
        ID2_DBG_LOG("id2 client generate authcode
        return -1;
    }

    cipher_len = strlen(cipher_data);
    if (cipher_len > ID2_ID_LEN * 2) {
        ret = id2_client_decrypt_data(cipher_data,
```

### 3.3 在设备端进行解密的验证

- 重新编译生成固件（id2_app）
- 烧录固件到设备
- 运行固件，进行解密验证



```
                        "args": {
                        },
                        "result":       "000FFFFFEAE
                }, {
                        "api":  "id2_client_get_chal
                        "args": {
                                "challenge":    "55E
                                "extra":        "abc
                        },
                        "result":       "2~2~C4B69AE
                }, {
                        "api":  "id2_client_get_time
                        "args": {
                                "timestamp":    "151
                                "extra":        "abc
                        },
                        "result":       "3~2~C4B69AE
                }]
}

<LS_LOG> id2_client_cleanup 579: [id2_client_cleanup
<LS_LOG> id2_client_generate_authcode 186: =====>ID2

<LS_LOG> id2_client_decrypt_data 202: ====> ID2 Clie
<LS_LOG> id2_client_init 556: [id2_client_init enter
<LS_LOG> ID2 Client Version: 0x00020000
<LS_LOG> ID2 Client Build Time: Aug 27 2019 10:18:30
<LS_LOG> ----------------------------------------
<LS_LOG> CONFIG_ID2_DEBUG is defined!
<LS_LOG> CONFIG_ID2_OTP is defined!
<LS_LOG> CONFIG_ID2_KEY_TYPE: ID2_KEY_TYPE_AES
<LS_LOG> ----------------------------------------
<LS_LOG> id2_client_get_id 606: [id2_client_get_id e
<LS_LOG> id2_client_get_id 649: ID2: 000FFFFFEAEDFDE
<LS_LOG> id2_client_decrypt 700: [id2_client_decrypt
<LS_LOG> id2_log_hex_dump 99: id2_cipher_input: [len
<LS_LOG> id2_log_hex_dump 109: 85 5C 18 1E C6 21 94
<LS_LOG> id2_log_hex_dump 109: BC AE 76 E0 36 86 AA
```