# 步骤一:创建 CEF 账号

## 通过 PowerShell 创建

1. 登录 Azure China

Login-AzureRmAccount -Environment AzureChinaCloud

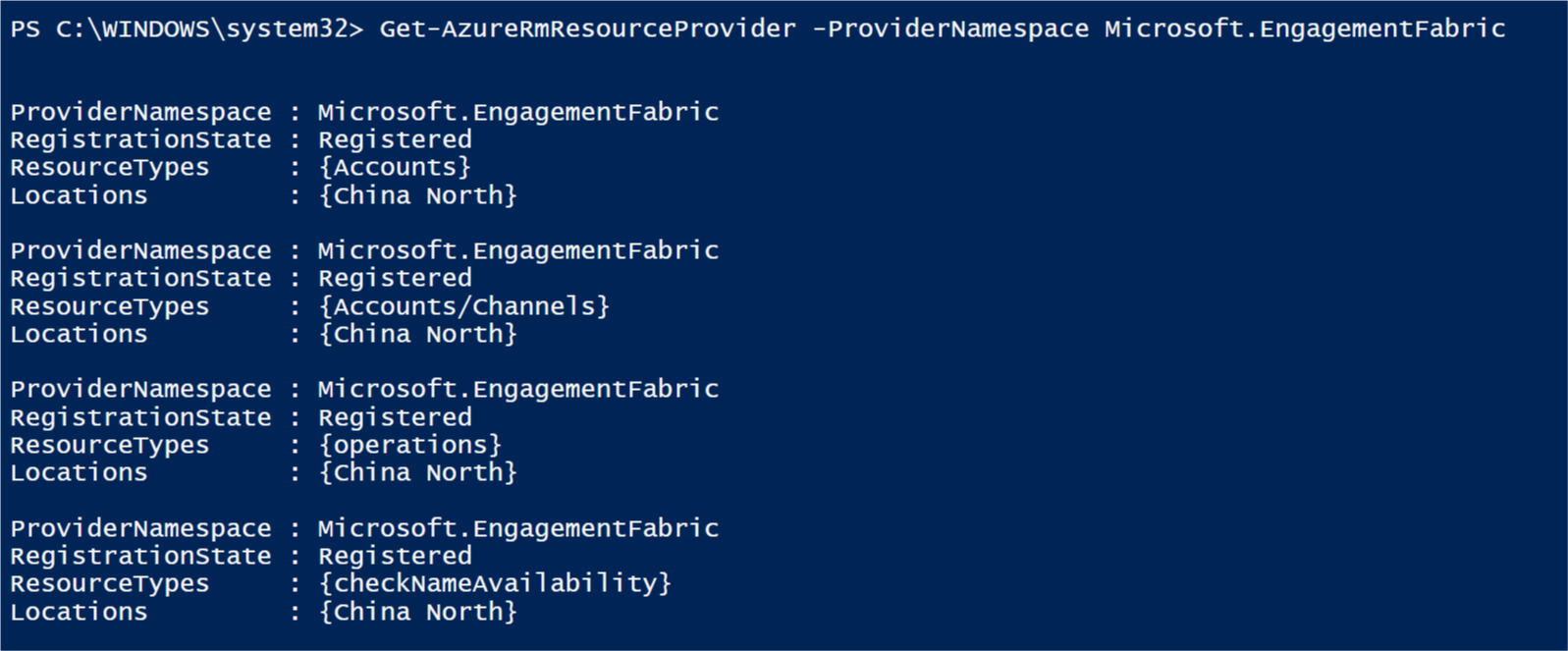
1. 选择你的 subscription (如果有多个 subscriptions)

Select-AzureRmSubscription -SubscriptionId <subscription ID>

1. 注册 CEF Resource Provider (每个 subscription 只需做一次)

Register-AzureRmResourceProvider -ProviderNamespace Microsoft.EngagementFabric

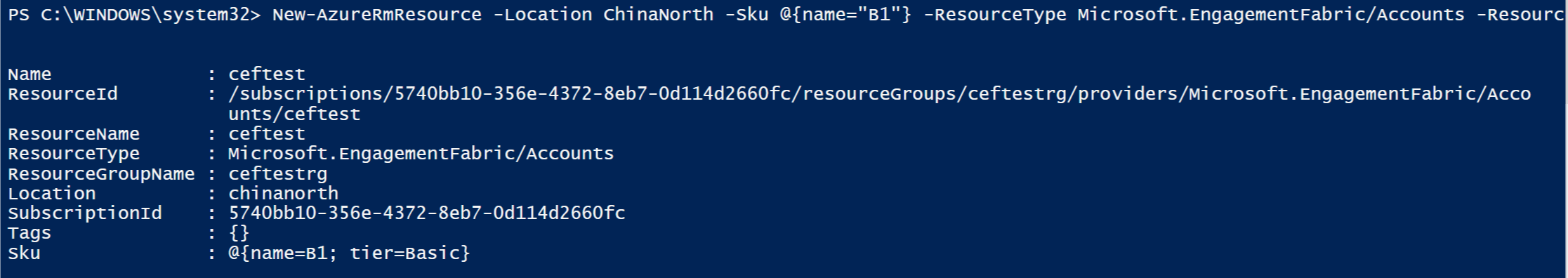
注意: 这是个异步操作，请调用下面这条命令直到“RegistrationState” 为“Registered”



Get-AzureRmResourceProvider -ProviderNamespace Microsoft.EngagementFabric

1. 创建 CEF 账号

New-AzureRmResource -Location ChinaNorth -Sku @{name="B1"} -ResourceType Microsoft.EngagementFabric/Accounts -ResourceGroupName ceftestrg -ResourceName ceftest



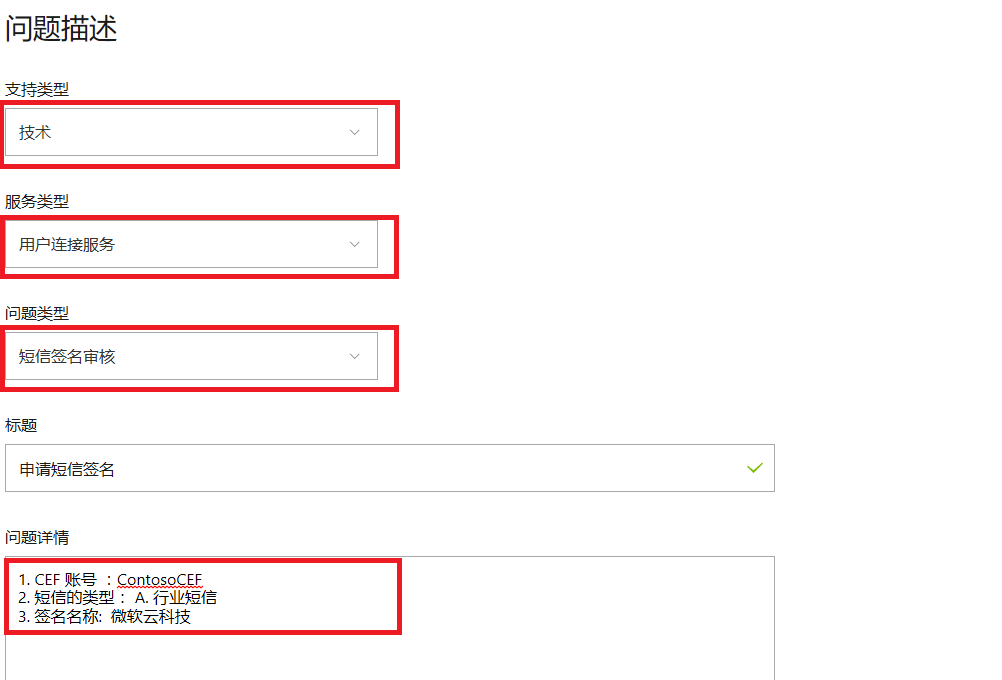
# 步骤二:申请签名

短信签名是作为短信发送者属性的一种标识，签名必须用于标识公司、产品或业务，不得涉及违规违法信息。用户需通过[**提交工单**](https://support.azure.cn/zh-cn/support/support-azure/)申请签名：选择**支持类型**“**技术**”->**服务类型**“**用户连接服务**”->**问题类型**“**短信签名审核**”。

工单中需提供：

1. 提供 CEF 账号 (参见[创建 CEF 账号](https://docs.azure.cn/zh-cn/customer-engagement-fabric/customer-engagement-fabric-create-cef-account-rest-api))
2. 选择发送短信的类型： A.行业短信 B.营销短信 C.行业+营销。
   * 如签名类型是 B 或 C，须提供营销短信样例。A 则不需要。
3. 提供**签名名称**（签名名称要求 2~12 个字，由中英文，数字和空格组成，不支持符号及全数字）
4. 提供公司营业执照扫描件
   * 如签名是属于以下类型还需提供补充材料：
     + 公司名全称或简称, 提供营业执照扫描件即可，无需补充材料。
     + APP 名全称或简称，需提供任一应用商店的下载链接与该应用商店的后台管理截图
     + 网站名全称或简称，需提供域名备案服务商的后台备案截图
     + 商标全称或简称，需提供商标注册证书截图
5. 下载[《客户信息安全保障责任书》](https://cefproddoc.blob.core.chinacloudapi.cn/public/%E5%AE%A2%E6%88%B7%E4%BF%A1%E6%81%AF%E5%AE%89%E5%85%A8%E4%BF%9D%E9%9A%9C%E8%B4%A3%E4%BB%BB%E4%B9%A6.docx)，并将签字盖章后的扫描件上传。

## 工单例子

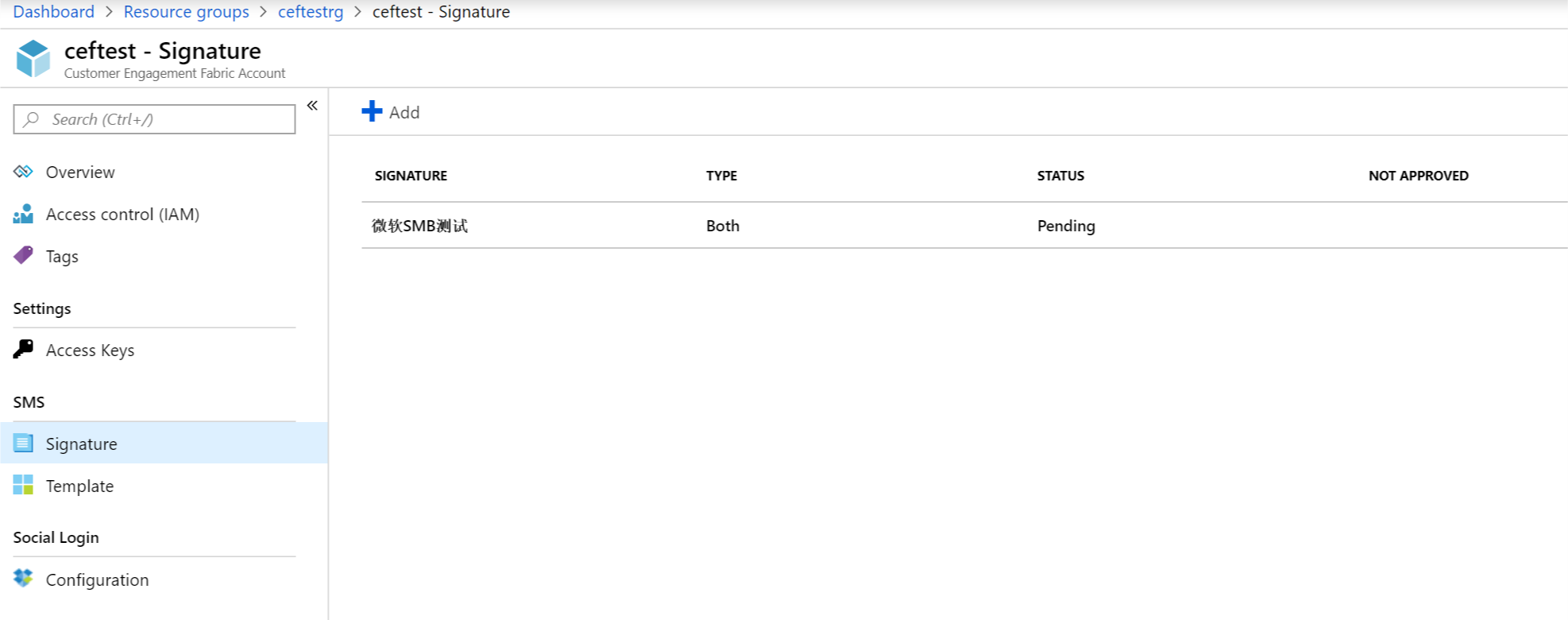




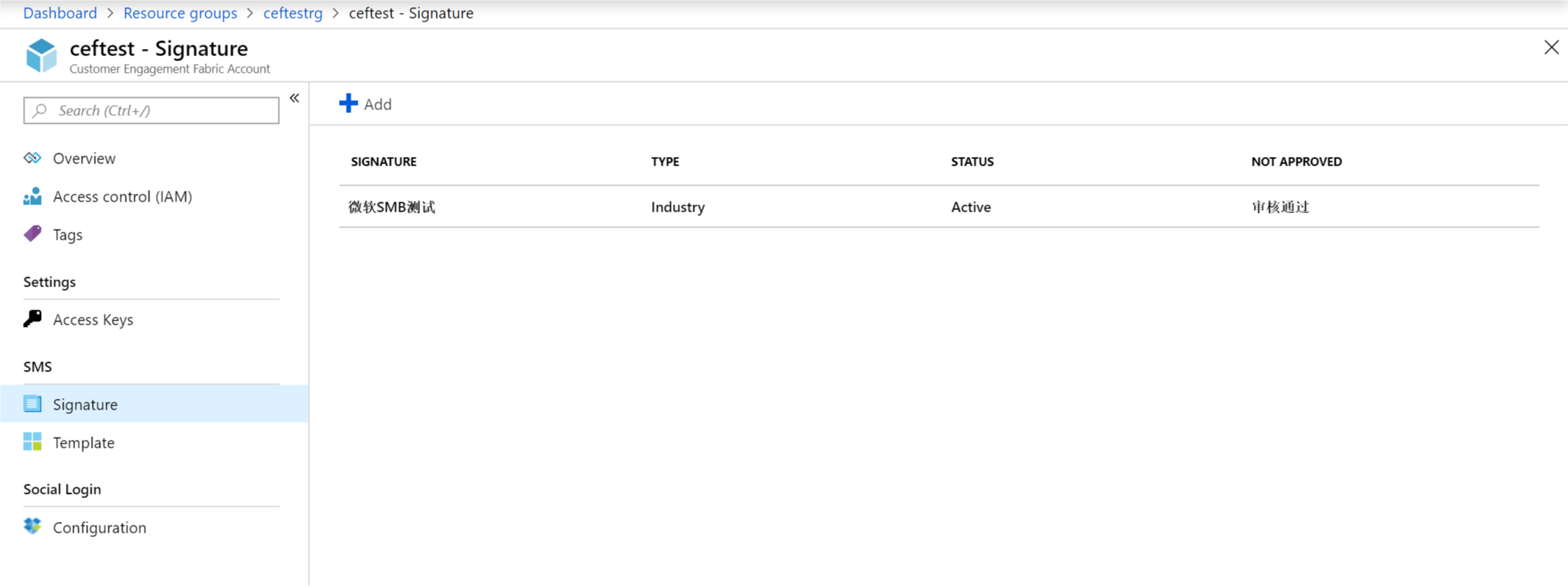
官方文档:

<https://docs.azure.cn/zh-cn/customer-engagement-fabric/customer-engagement-fabric-sms-notification-apply-signature>

申请一般两个工作日内通过, 下图为审核过程中, 状态是Pending.



短信签名审核通过后, 状态更新为Active, 审核通过, 如下图.

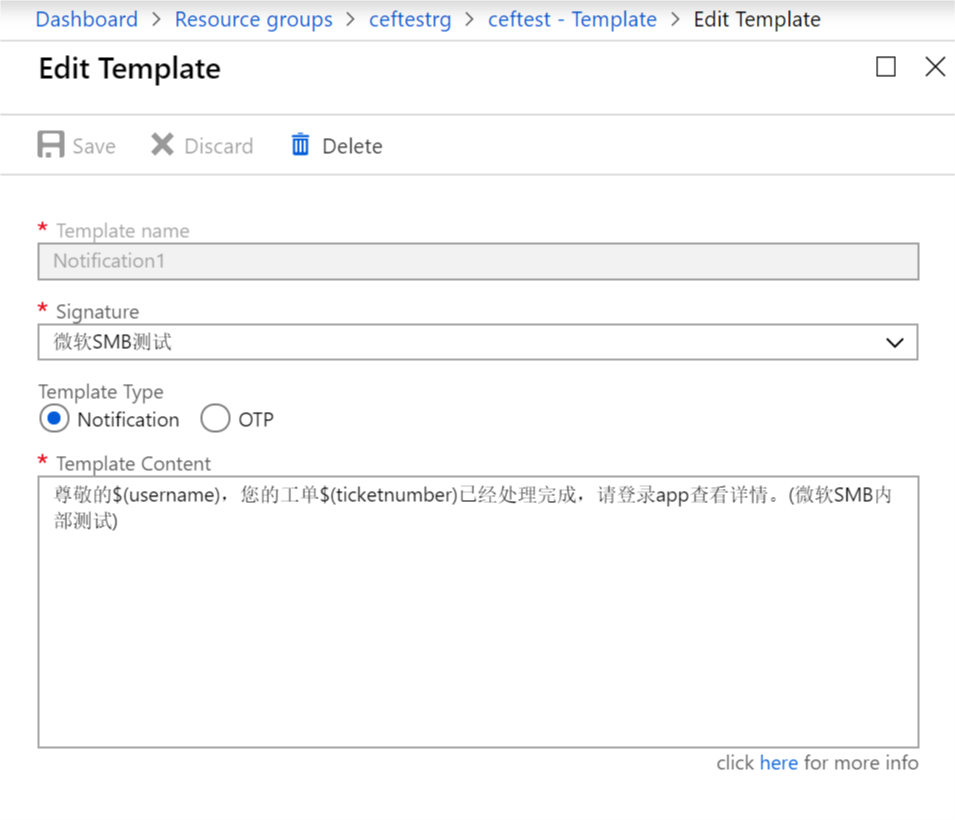


# 步骤三:申请短信模板

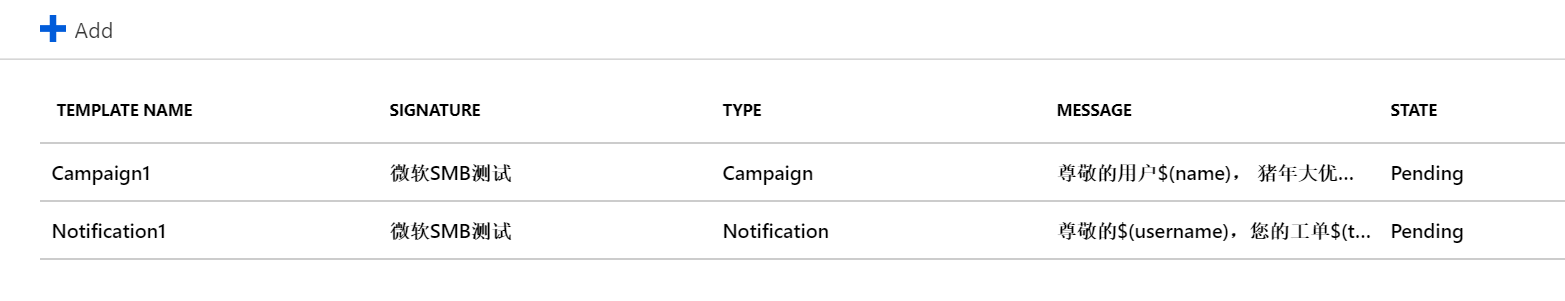
在Azure Portal 中, 选择Template, 增加一个新的短信模板. 需要根据短信的类型选择营销或者是通知短信. 如下图所示, 申请一个通知类短消息.

具体格式要求可参考官方文档:

<https://docs.azure.cn/zh-cn/customer-engagement-fabric/customer-engagement-fabric-sms-notification-apply-template>



短消息模板同样需要后台审核, 审核过程中的状态为Pending, 如下图.

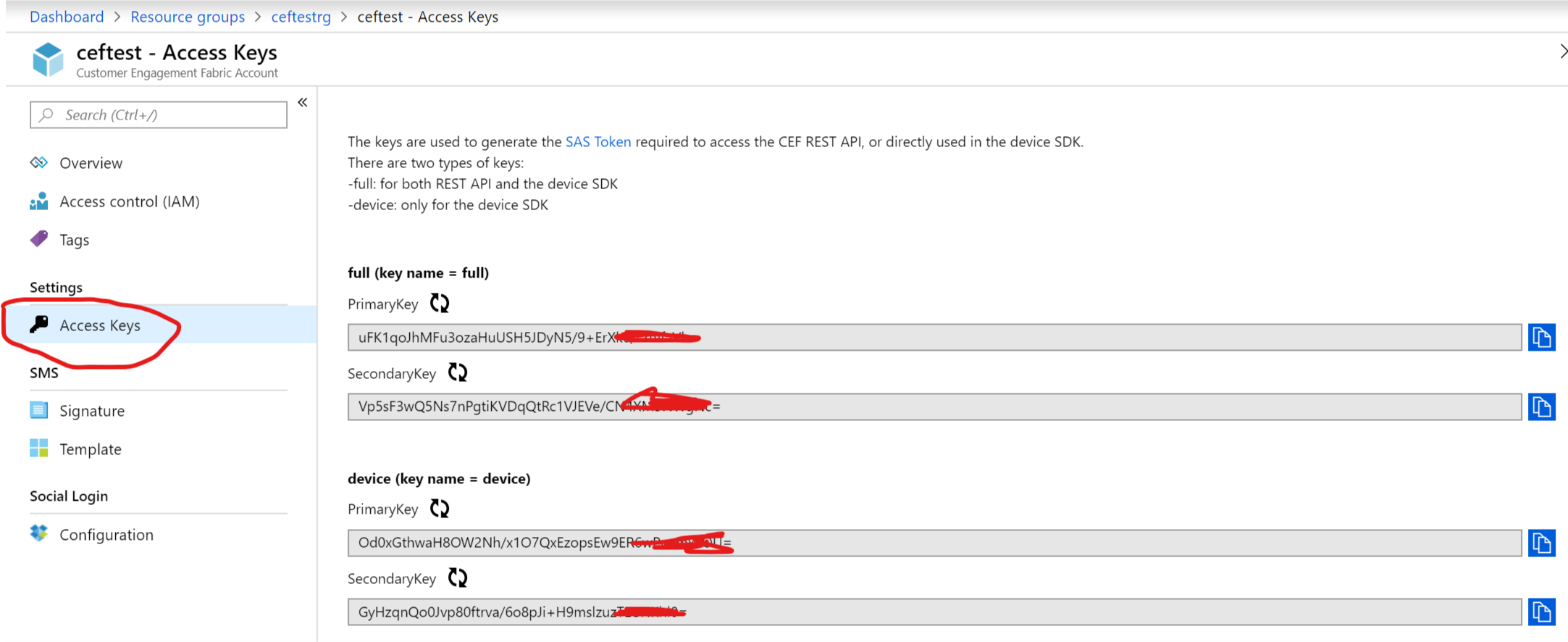


# 步骤四:发送短信

用户通过 REST API 接入短信服务。开发者可以使用任何编程语言与 REST API 进行交互。CEF REST API 的每个请求都需要通过 SAS Token 进行身份验证。身份验证信息以 Authorization 请求头发送。. 该Token是通过CEF 访问秘钥计算获得. 因此首先要获取CEF Key.

## 通过Portal获取 CEF Key

登录Azure Portal, 选择上面步骤创建的CEF账户, 选择Access Keys. 记录full秘钥的Primary Key内容.



## 共享访问签名令牌(SAS Token)

SAS Token的计算示例可参考官方文档:

<https://docs.azure.cn/zh-cn/customer-engagement-fabric/customer-engagement-fabric-generate-sas-token>

也可以通过下面Powershell命令获取SAS Token(Key部分替换为上述步骤获取的CEF Key)

[Reflection.Assembly]::LoadWithPartialName("System.Web")| out-null

$Exp=1

$Key="替换为上述步骤获取的CEF Key"

$Expires=([DateTimeOffset]::Now.ToUnixTimeSeconds())+$Exp\*3600

$SignatureString= "se=" + [string]$Expires +"&skn=full"

$HMAC = New-Object System.Security.Cryptography.HMACSHA256

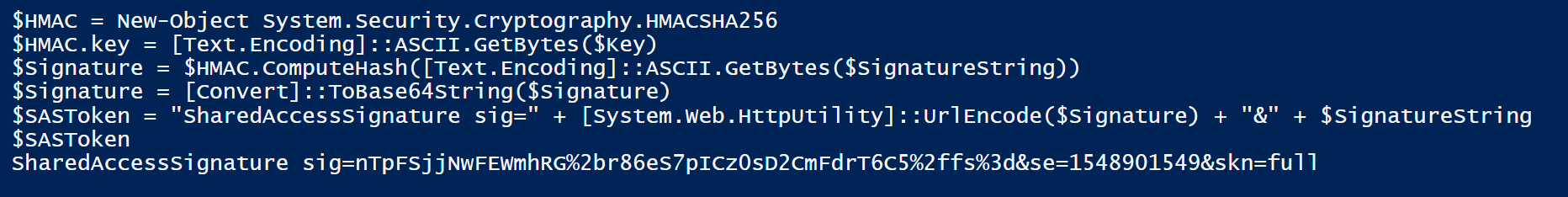
$HMAC.key = [Text.Encoding]::ASCII.GetBytes($Key)

$Signature = $HMAC.ComputeHash([Text.Encoding]::ASCII.GetBytes($SignatureString))

$Signature = [Convert]::ToBase64String($Signature)

$SASToken = "SharedAccessSignature sig=" + [System.Web.HttpUtility]::UrlEncode($Signature) + "&" + $SignatureString

$SASToken



复制获取到的SAS Token例如,

SharedAccessSignature sig=nTpFSjjNwFEWmhRG%2br86eS7pICz0sD2CmFdrT6C5%2ffs%3d&se=1548901549&skn=full

## 通过Postman 测试短消息功能

使用Postman 测试单条发送功能. CEF REST API的规范参考:

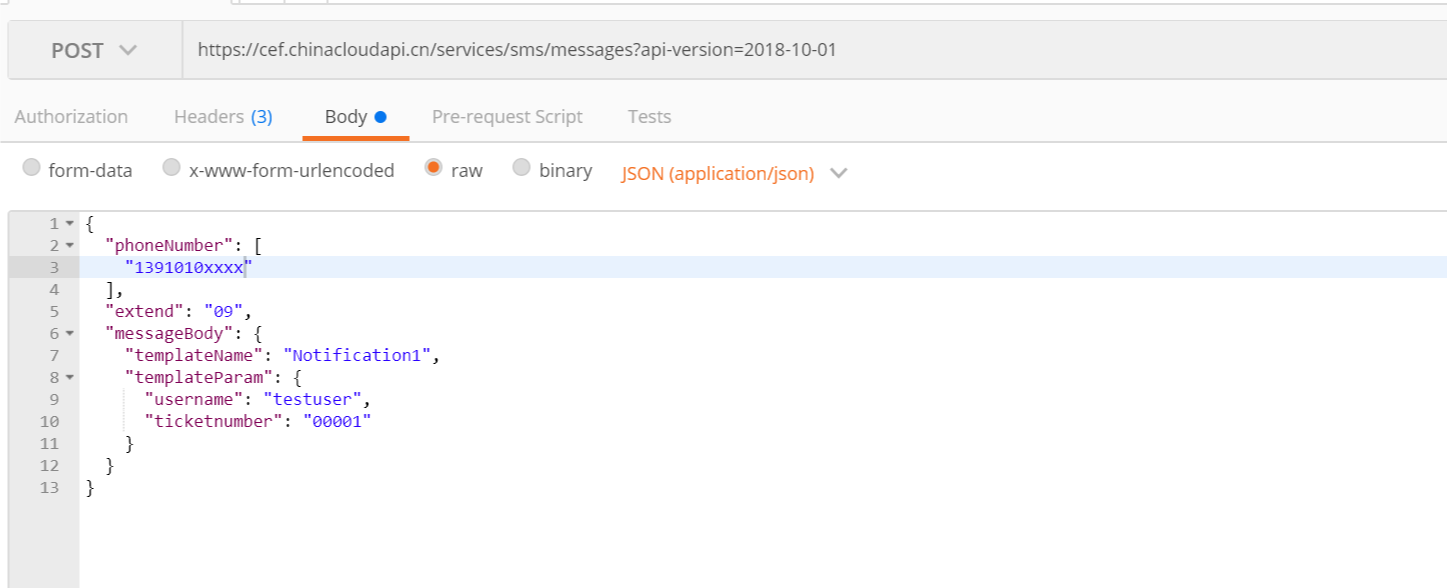
<https://docs.azure.cn/zh-cn/customer-engagement-fabric/customer-engagement-fabric-rest-api-send-one-message>

单条发送API使用POST方法, 需要三个Header, 分别是Account, Authentication和Content-Type

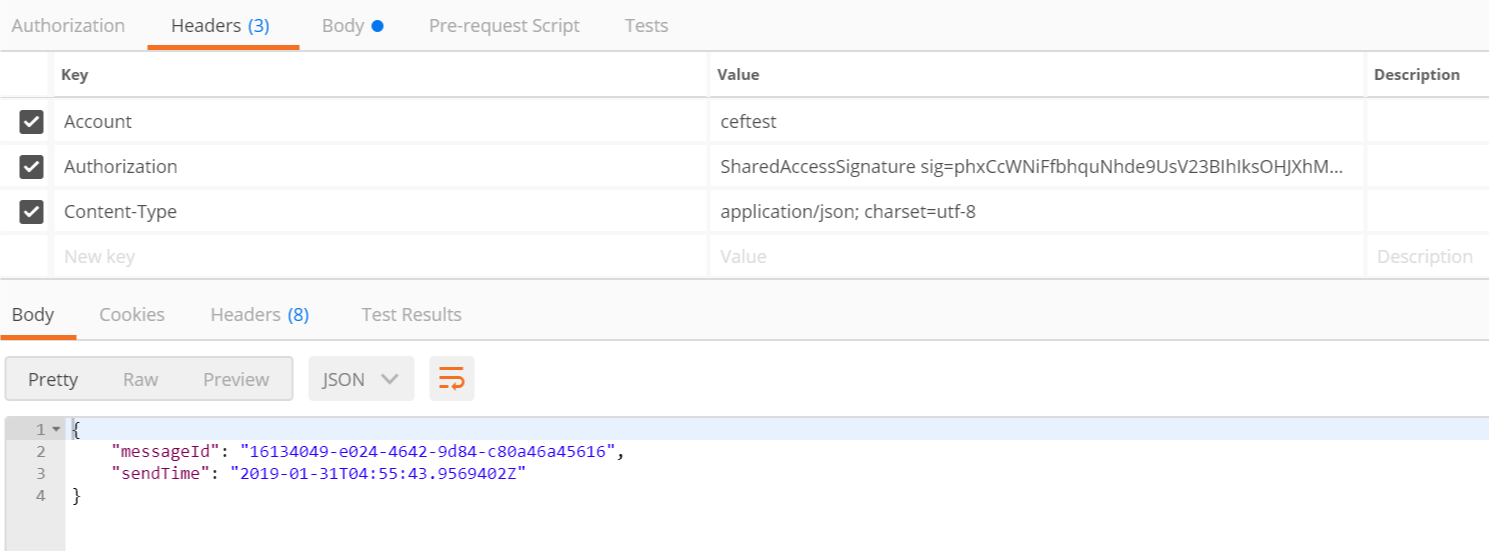


Account 就是CEF账号, Authentication填写SAS Token, Content-Type 填写application/json

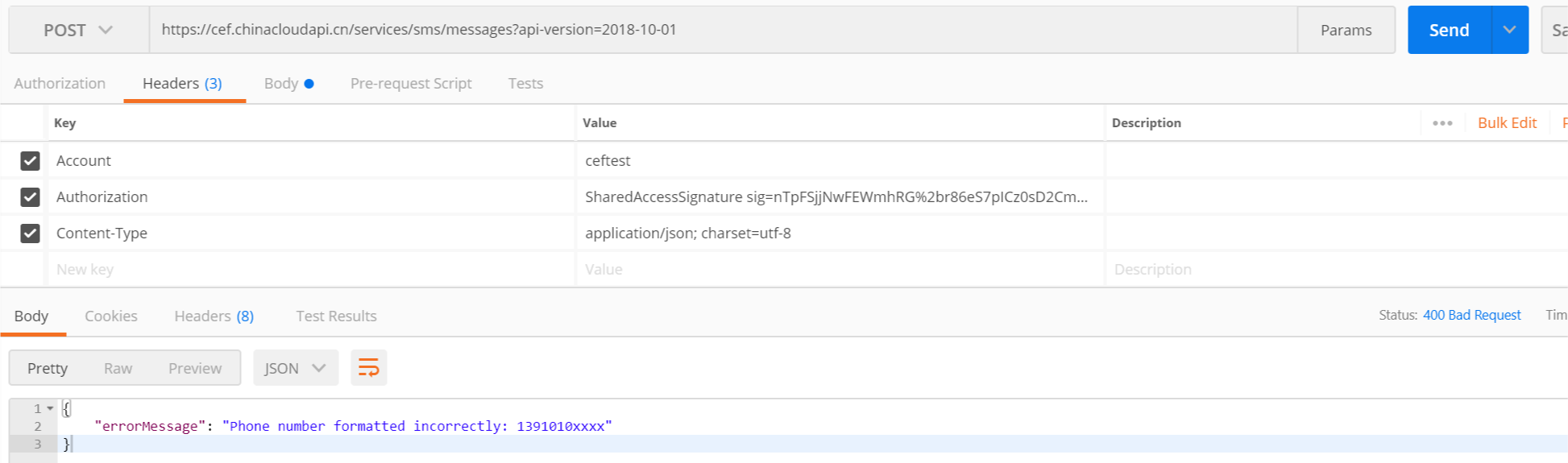
Body的内容需要与之前申请的短消息模板匹配, 填写一个中国大陆的手机号码即可.



点击发送, Postman将上述消息发送至Azure CEF服务. 收到短信成功发送消息, 或者错误信息,如下图.



短信成功会显示消息ID和发送时间.

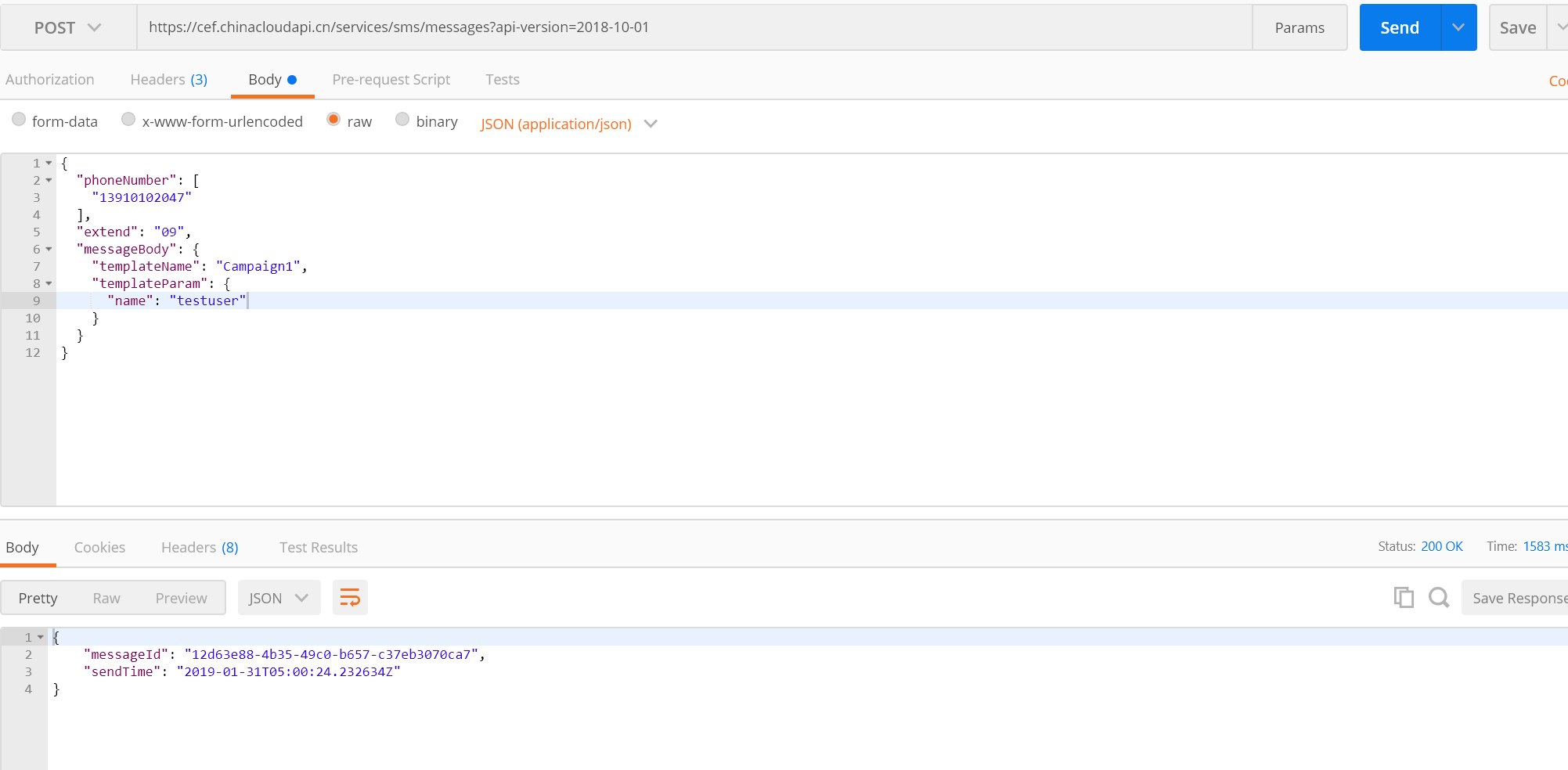


短信失败会给出错误原因.



收到的短信样例.

营销短信的样例



# .net样例程序

完成程序包通过下面链接下载. <https://github.com/haoxin/CEFDemo>

private void button1\_Click(object sender, EventArgs e)

{

TimeSpan ts = new TimeSpan(System.Convert.ToInt32(textExp.Text) \* 3600, 0, 0);

String SAS = CreateSASToken(textKey.Text, "full", ts);

textBox1.Text = SAS;

returnLabel.Text = sendSMS(SAS, textPhone.Text);

}

//根据Key, Key名字和过期时间计算SAS Token.

public static string CreateSASToken(string key, string keyName, TimeSpan timeout)

{

const string Schema = "SharedAccessSignature";

const string SignKey = "sig";

const string KeyNameKey = "skn";

const string ExpiryKey = "se";

var values = new Dictionary<string, string>

{

{ KeyNameKey, keyName },

{ ExpiryKey, (DateTimeOffset.UtcNow + timeout).ToUnixTimeSeconds().ToString() }

};

var signContent = string.Join("&", values

.Where(pair => pair.Key != SignKey)

.OrderBy(pair => pair.Key)

.Select(pair => $"{pair.Key}={WebUtility.UrlEncode(pair.Value)}"));

string sign;

using (var hmac = new HMACSHA256(Encoding.UTF8.GetBytes(key)))

{

sign = Convert.ToBase64String(hmac.ComputeHash(Encoding.UTF8.GetBytes(signContent)));

}

return $"{Schema} {SignKey}={WebUtility.UrlEncode(sign)}&{signContent}";

}

//根据短信模板内容, 构建Json消息体, 通过Httpclient 发送消息.

public static string sendSMS(string SASToken, String phoneNumber)

{

String url = "https://cef.chinacloudapi.cn/services/sms/messages?api-version=2018-10-01";

HttpClient client = new HttpClient();

client.DefaultRequestHeaders.Add("Account", "ceftest");

client.DefaultRequestHeaders.Add("Authorization", SASToken);

JObject message = new JObject(

new JProperty("phonenumber",

new JArray(phoneNumber)

),

new JProperty("extend", "09"),

new JProperty("messageBody",

new JObject(

new JProperty("templateName", "Notification1"),

new JProperty("templateParam",

new JObject(

new JProperty("username", "testuser"),

new JProperty("ticketnumber", "xxx0001")

)))));

Console.WriteLine(message.ToString());

StringContent content = new StringContent(message.ToString(), Encoding.UTF8, "application/json");

var response = client.PostAsync(url, content).Result;

var body = response.Content.ReadAsStringAsync().Result;

return body.ToString();

}