

语音识别与合成





1. 探索新知

1.1

准备工作

想要实现今天的效果，将用到【百度智能云】语音转文字和语音合成的相关接口



1. 注册百度智能云账号



给大家介绍准备工作的详细过程

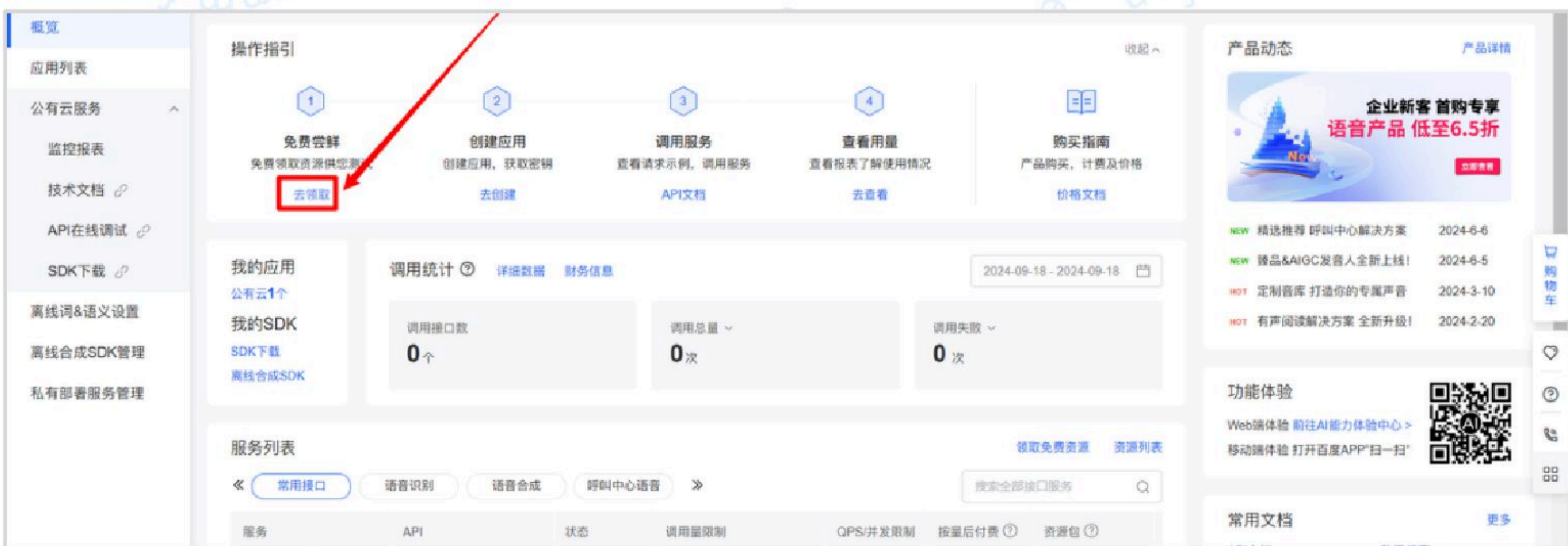
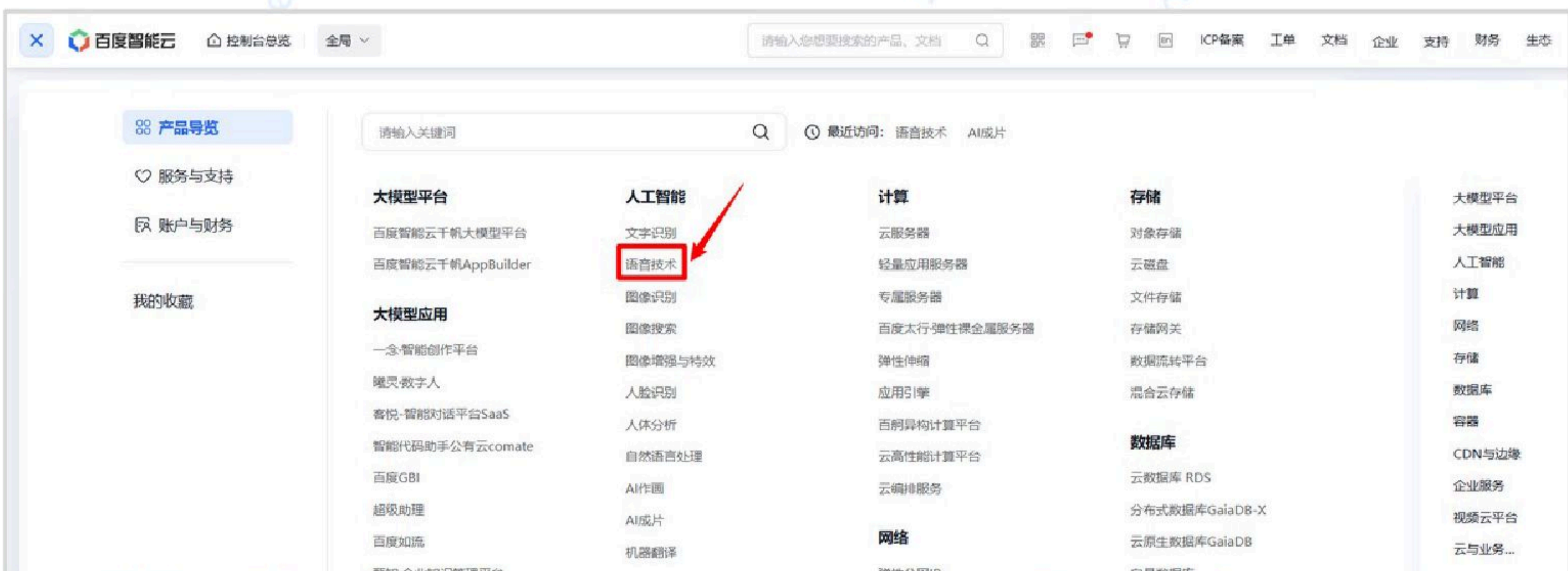
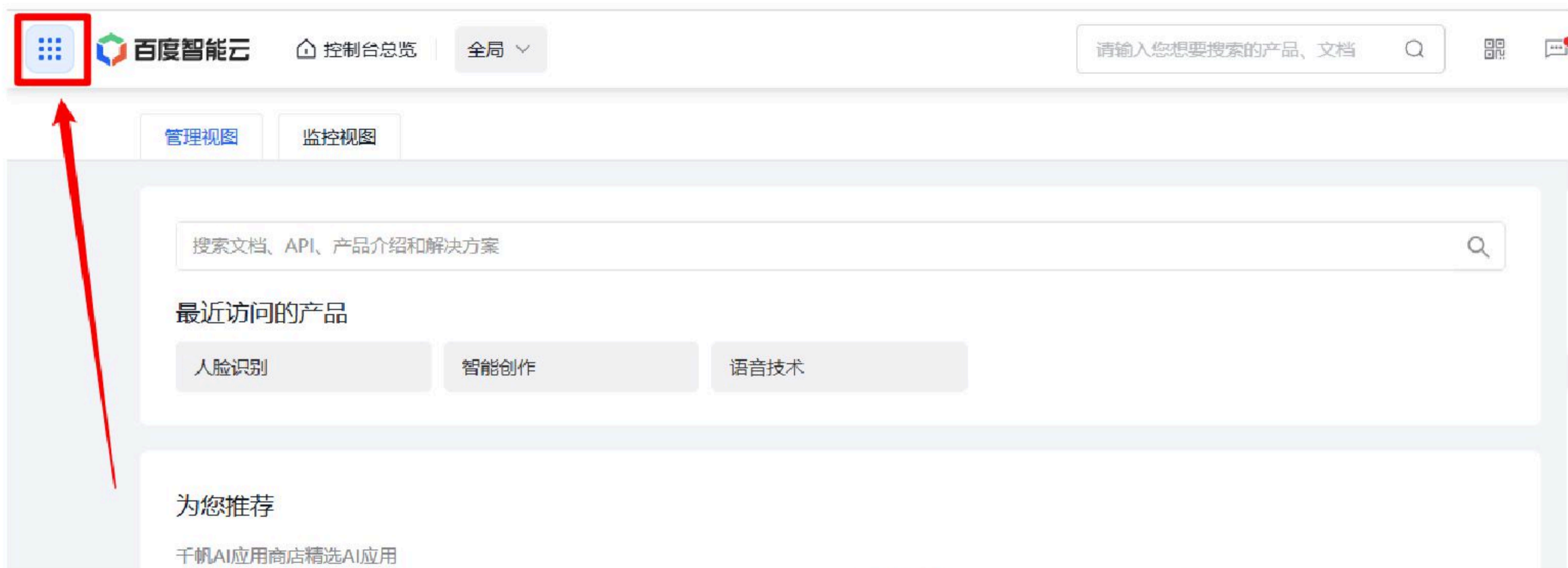


2. 打开产品—【语音技术】

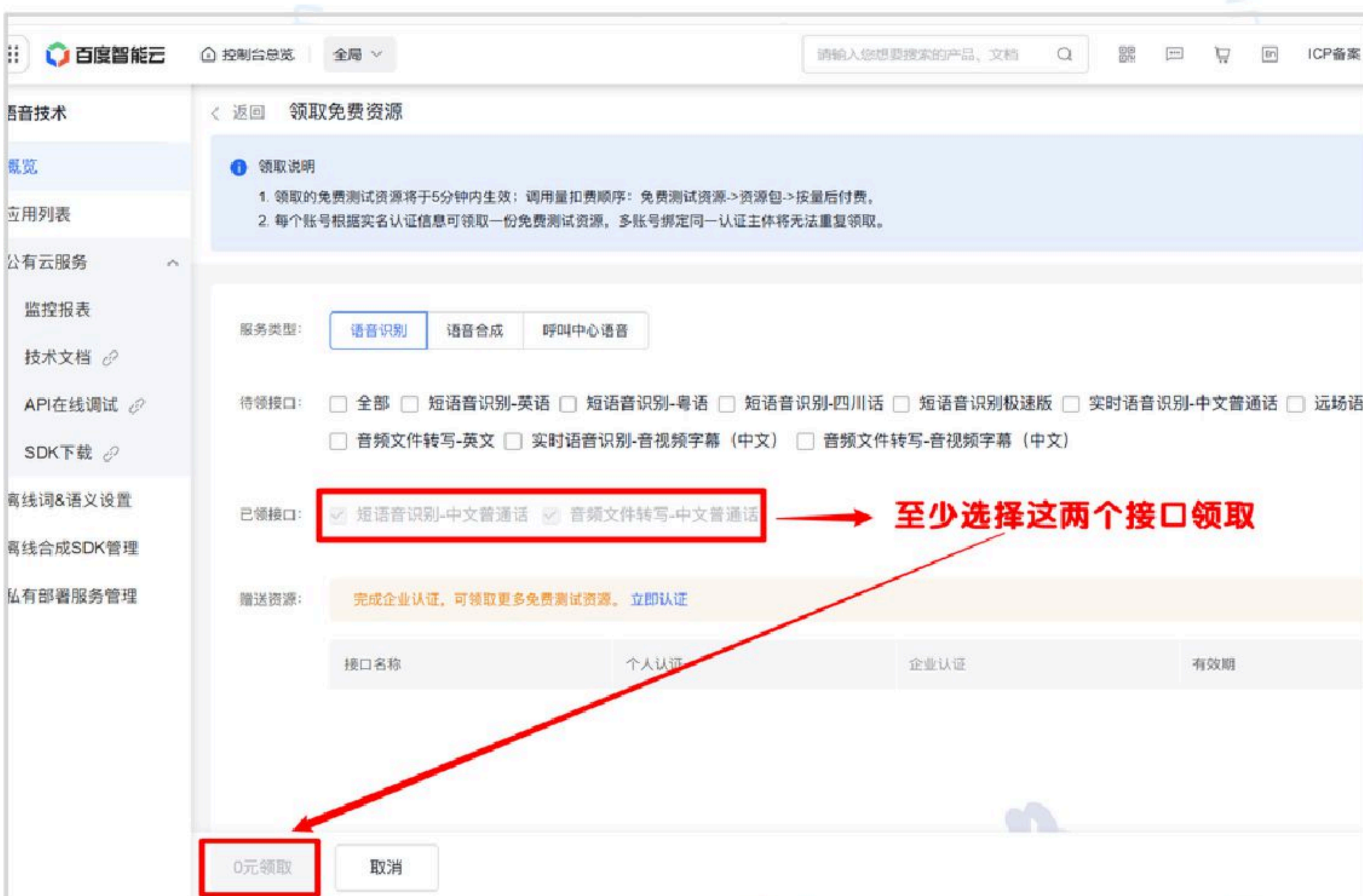


1.1

准备工作



3. 领取免费接口



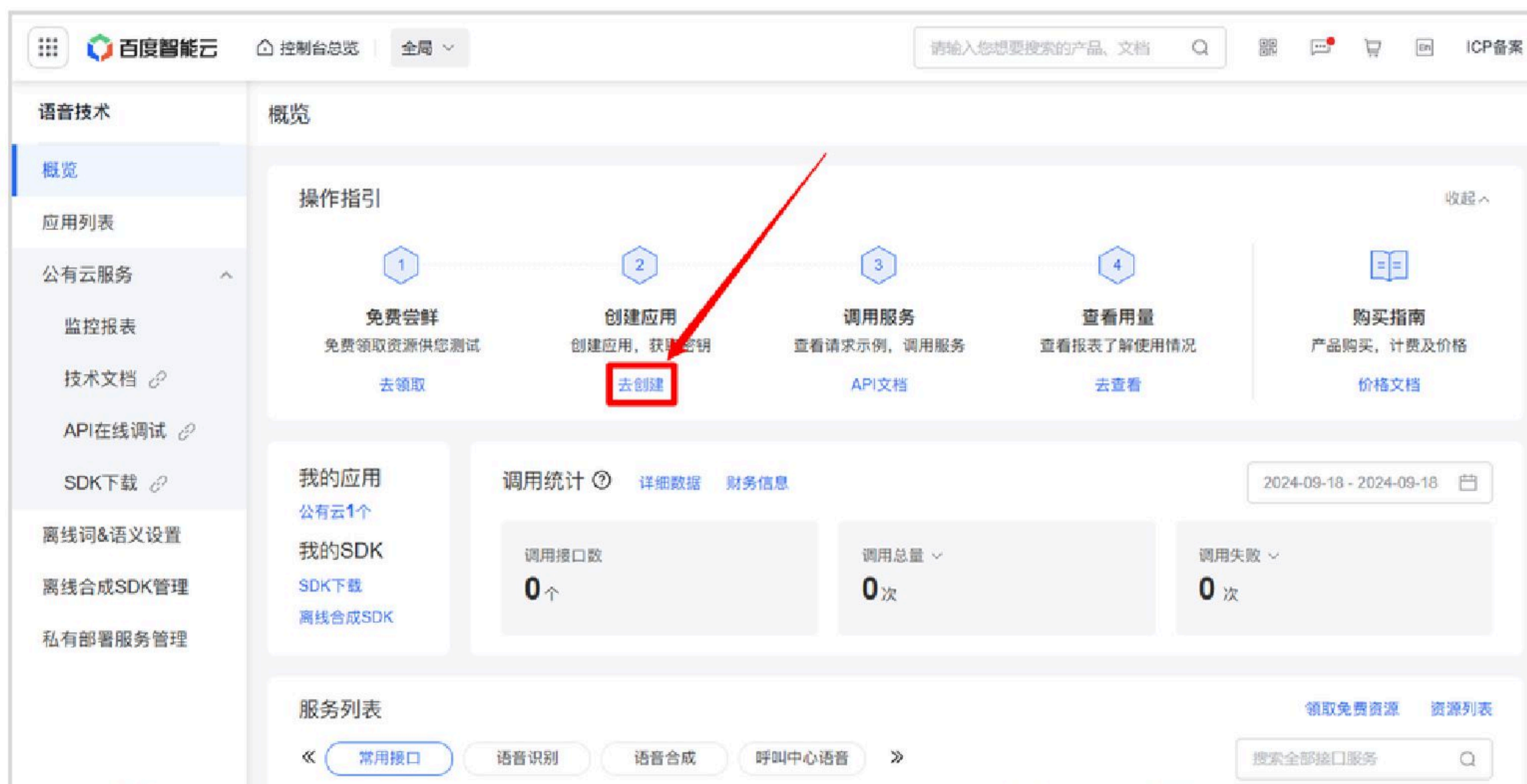
我已经领取过接口，所以两接口会在【已领接口】中。

大家在操作时，至少选择这俩接口，也可以直接领取【全部】接口，点击左下方的【0元领取】，即可领取试用（试用时长180天）。

1.1

准备工作

4. 创建应用



注意:

在创建应用之前，请确认自己已成功领取免费测试资源



根据业务需求，填写如下应用信息：

【应用名称】 必填项，所创建的应用名称，支持中英文、数字、下划线及中横线

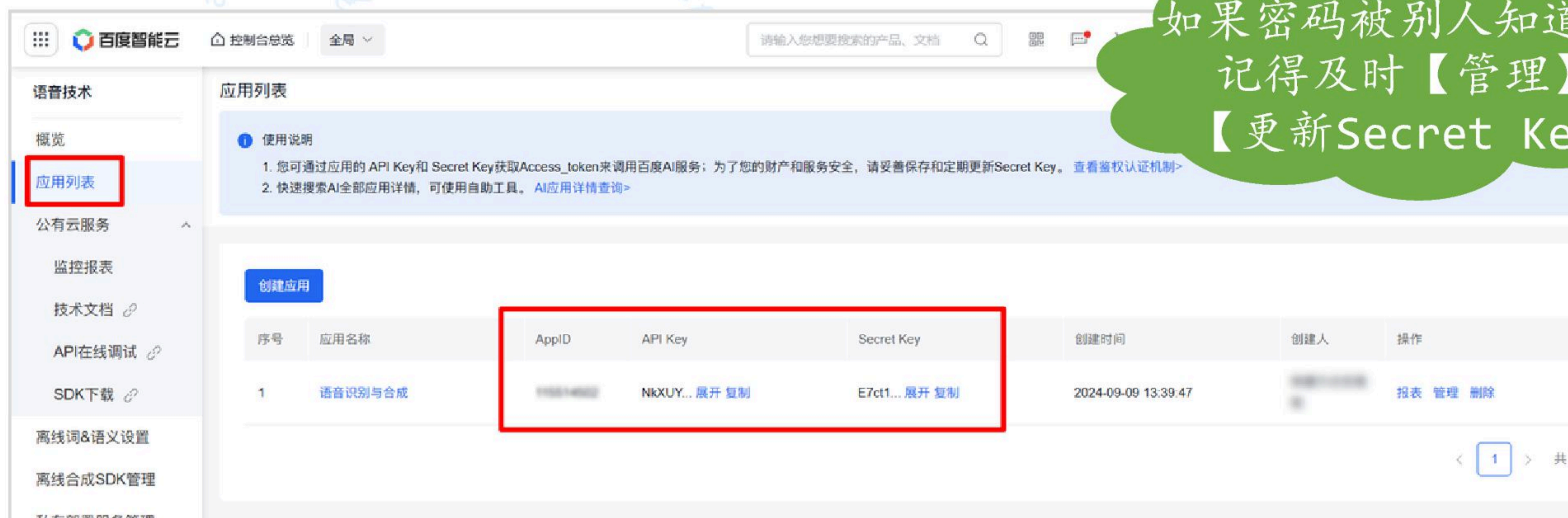
【接口选择】 勾选该应用需要用到的接口

【语音包名】 勾选【不需要】即可

【应用归属】 必选项，选择【个人】即可

【应用描述】 必填项，对此应用的业务场景进行简单描述

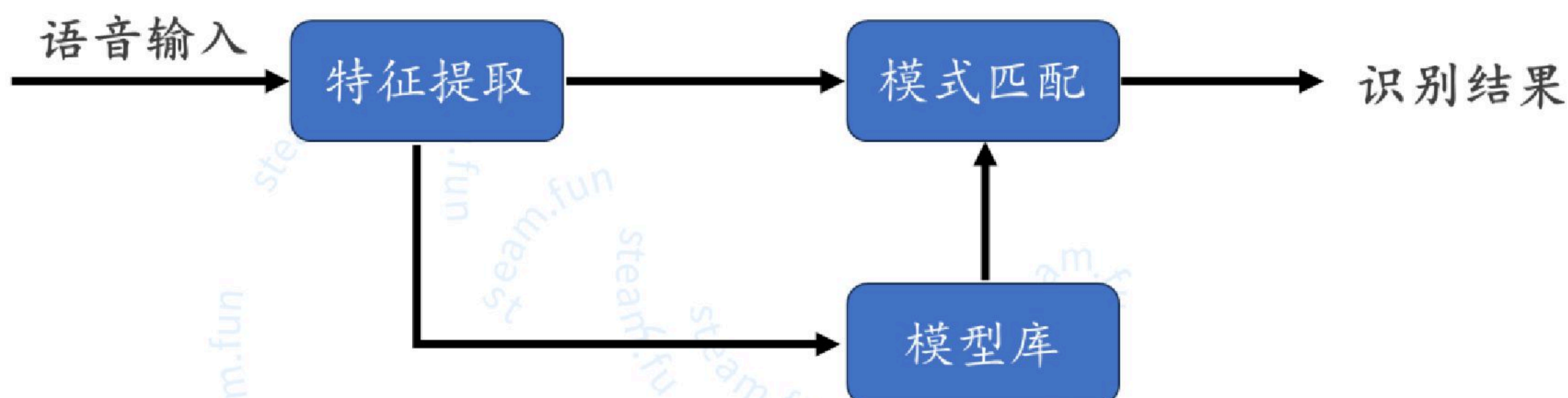
5. 获得应用的AK/SK



1.2

语音识别

语音识别的实现步骤：



了解了基本原理，试着实现效果。录音的代码在【语音识别.py】文件中预置好了，我们需要实现的是**语音识别**功能。

```
1 import pyaudio
2 import wave
3 import keyboard
4
5 # 从麦克风录制音频并保存到指定文件
6
7 # 创建 PyAudio 对象
8 p = pyaudio.PyAudio()
9 # 打开音频流，设置格式、声道数、采样率等参数
10 stream = p.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True, frames_per_buffer=1024)
11 # 创建一个空列表用于存储录制的音频数据
12 frames = []
13 # 打印提示信息，表示开始录制
14 print("Press 's' to start recording...")
15 # 等待用户按下 's' 键开始录制
16 keyboard.wait('s')
17 # 打印提示信息，表示开始录制
18 print("Recording...")
19 # 开始录制
20 while True:
21     # 从音频流中读取 1024 个数据样本
22     data = stream.read(1024)
23     # 将数据添加到 frames 列表中
24     frames.append(data)
25     # 检查是否按下 'q' 键结束录制
26     if keyboard.is_pressed('q'):
```

按下“s”键【开始录音】
按下“q”键【停止录音】

• 安装需要的第三方库

```
1 # 点击运行【安装第三方库.py】，即可安装本课用到的第三方库
2
3 import pip
4 pip.main(['install', 'baidu-aip'])
5 pip.main(['install', 'chardet'])
```

1. 【baidu-aip】百度人工智能开放平台提供的Python SDK（软件开发工具包），用于连接并调用百度AI开放平台的各种API接口
2. 【chardet】用于检测文本文件或字符串的字符编码的Python库

1.2

语音识别

1. 引入所需要的库

```
from aip import AipSpeech
```



这行代码告诉电脑：“请帮我把 AipSpeech 这个工具拿过来，我要用。”

AipSpeech



2. 定义常量

```
APP_ID = '你的 AppID '  
API_KEY = '你的 API Key '  
SECRET_KEY = '你的 Secret Key '
```

这里，我们告诉电脑一些重要的信息（称为常量），这些信息对使用 AipSpeech 工具非常重要

（这三个信息是应用开发中的重要凭证，请谨慎保管）

3. 准备【语音识别器】

```
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
```

这行代码我们准备好一个叫 client 的工具，这个工具就像一台机器，它通过前面设好的信息（APP_ID, API_KEY, SECRET_KEY）来工作，我们可以用它来与百度的图像识别服务进行沟通。

• 为什么要起名为 client ？

client 是一个常见的命名方式，表示某种服务接口的客户端对象。在这儿我们就称呼 client 为【语音识别器】吧。

1.2

语音识别

4. 定义要识别的语音文件

```
file = 'pyaudio.wav' # 这里写要识别的语音文件名
```

```
29 print("Recording stopped.")
30 # 停止音频流
31 stream.stop_stream()
32 # 关闭音频流
33 stream.close()
34 # 终止 PyAudio 对象
35 p.terminate()
36 # 打开文件, 以写入模式打开
37 wf = wave.open("pyaudio.wav", 'wb')
38 # 设置音频的声道数、采样宽度和采样率
39 wf.setnchannels(1)
40 wf.setsampwidth(p.get_sample_size(pyaudio.paInt16))
41 wf.setframerate(16000)
42 # 将 frames 列表中的所有数据连接成一个字节串, 并写入文件
43 wf.writeframes(b''.join(frames))
44 # 关闭文件
45 wf.close()
46 print("Done")
```

生成的录音文件为
【pyaudio.wav】

这里我们告诉电脑：“你要去找一个叫'pyaudio.wav'的文件，这个文件里有我要你识别的声音。”

5. 调用【语音识别器】

```
res = client.asr(read_audio(file), 'wav', 16000, {'dev_pid': 1537})
```

这行代码让我们的工具client来识别音频文件file，并返回结果。

read_audio(file): 自定义函数，读取音频文件的内容

'wav', 16000: 告诉机器文件格式是wav，音频采样率是16000

{'dev_pid': 1537}: 告诉机器我们要识别的是普通话

6. 读取音频文件

```
def read_audio(file):
    with open(file, 'rb') as f:
        return f.read()
```

with open(file, 'rb') as f:

告诉电脑“用rb模式（读二进制）的方式打开这个文件，一会儿我要用它。”

return f.read()

这行代码相当于读这个文件的全部内容并返回。

思考：为什么使用 'rb' 模式打开读取？

在Python中，使用 rb 模式打开文件通常是为了读取二进制文件，这对于读取可能会在文件开始部分包含不可打印字符的文件非常重要，例如图片、音频或者其他的二进制数据。这种模式下，Python不会做任何解释，按照原始的字节形式读取文件内容。

7. 打印返回结果

如果识别成功，识别结果放在 “result” 字段中

```
if 'result' in res:
    print("识别结果: ", res['result'][0])
else:
    print("识别失败: ", res)
```

- 识别成功返回例子：

```
{'corpus_no': '7416174414138101560', 'err_msg': 'success.',
'err_no': 0, 'result': ['你好。'], 'sn': '333613074851726712662'}
```

- 识别错误返回例子：

```
{'err_msg': 'recognition error.', 'err_no': 3307, 'sn':
'454775077861726713456'}
```

任务二新增代码：

```
from aip import AipSpeech

# 定义常量
APP_ID = '你的 AppID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'
# 准备【语音识别器】
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)

# 定义要识别的语音文件
file = 'pyaudio.wav'

# 读取音频文件
def read_audio(file):
    with open(file, 'rb') as f:
        return f.read()

# 调用【语音识别器】
res = client.asr(read_audio(file), 'wav', 16000, {
    'dev_pid': 1537 # 普通话识别
})

# 打印返回结果
if 'result' in res:
    print("识别结果：", res ['result'][0])
else:
    print("识别失败：", res)
```

1.3

语音合成



同样先完成准备工作，获取应用的AK/SK

- 领取【语音合成】的免费资源
- 创建【语音合成】的应用
- 获取应用的AK/SK



1. 准备【语音合成器】

```
from aip import AipSpeech

# 定义常量
APP_ID = '你的 AppID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 准备【语音合成器】
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
```

1.3

语音合成

2. 定义要合成的文本

```
text = "你好，我是P仔，很高兴认识你"
```



这里 text 是想让电脑说的内容

3. 调用【语音合成器】

```
# 调用【语音合成器】
res = client.synthesis(text, 'zh', 1, {
    'vol': 5, # 音量
    'per': 0, # 发音人
    'spd': 5 # 语速
})
print(res)
```

这段代码调用语音合成接口，让电脑把text中的文字变成语音

client.synthesis: 我们用 client 做“合成语音”的工作

参数:

text: 必填。建议每次请求文本不超过120字节，约为60个汉字或者字母数字。

lang: 必填。语言选择，目前只有中英文混合模式，填写固定值'zh'。

ctp: 必填。客户端类型选择，web端填写固定值1。

{ } 里面的参数:

'vol': 5 选填。音量，基础音库取值0-9，默认为5中音量。

'per': 0 选填。发音人是默认的一个人。

度小宇=1，度小美=0，度逍遥（基础）=3，度丫丫=4

'spd': 5 选填。语速，取值0-15，默认为5中语速。

1.3

语音合成

4. 判断调用是否成功

```
# 判断调用是否成功
if not isinstance(res,dict):
    # 获取合成音频结果
    with open('save.mp3', 'wb') as f:
        file.write(res)
    print('音频文件保存成功')
else:
    print('调用失败, 错误码: %s, 错误信息: %s' %
          (res['err_no'], res['err_msg']))
```

这段代码检查电脑有没有成功地把文字变成语音。

isinstance(object, classinfo):

作用: 判断 object 是否为 classinfo 类型

if not isinstance(res, dict):

如果result不是字典类型, 说明合成成功。

任务三新增代码：

```
from aip import AipSpeech
# 定义常量
APP_ID = '你的 AppID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'
# 准备【语音合成器】
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
# 要合成的文本
text = "你好，我是P仔，很高兴认识你"
# 调用【语音合成器】
res = client.synthesis(text, 'zh', 1, {
    'vol': 5, # 音量
    'per': 0, # 发音人
    'spd': 5 # 语速
})
print(res)

# 判断调用是否成功
if not isinstance(res, dict):
    # 获取合成音频结果
    with open('save.mp3', 'wb') as f:
        f.write(res)
    print('音频文件保存成功')
else:
    print('调用失败，错误码： %s， 错误信息： %s' % (res['err_no'], res['err_msg']))
```



2. 强化练习

1. 为了使用百度处理语音相关的任务，需要导入下面哪个类？（ ）
A. AipImageClassify
B. AipFace
C. AipSpeech
D. AipImageProcess
2. 调用 `client.asr()` 函数时，参数 `16000` 代表什么？（ ）
A. 音频文件的大小
B. 音频文件的持续时间
C. 音频文件的采样率
D. 音频文件的音量
3. 以下哪一个函数用于调用百度的语音合成服务？（ ）
A. recognition
B. synthesis
C. speak
D. convert
4. 在进行语音合成时，参数 `vol` 是用来控制什么的？（ ）
A. 音量
B. 语速
C. 发音人
D. 语种
5. 语音合成返回结果的正确保存方法是将其保存为哪个文件格式？（ ）
A. .txt
B. .jpg
C. .mp3
D. .pdf

3.术语箱

client (计算机) 客户端

4.课后挑战

柯南变声领结

任务描述:

变身名侦探柯南，制作一个神奇的变声领结！你将录制一段音频，通过语音识别转换成文字，再通过语音合成器将文字变成机器人音调！快来挑战吧，看谁的变声音效最酷！

作业要求:

1. 将课堂代码【语音识别】和【语音合成】整合到一个文件中
2. 文字合成语音，选择喜欢的“语音人”

