从零开始入门开发

（MVC 模式，前后端分离）

1 后端开发——以 Django 为例

官方文档：https://www.djangoproject.com/start/overview/

菜鸟教程链接：https://www.runoob.com/django/django-tutorial.html

1.1 环境

Python 3.6+（仅 3.6 版本支持 celery，便于后台语音识别批处理）

Django 3.0.6

1.2 开发步骤

1.2.1 基本环境创建

创建虚拟环境：conda create -n SpeechRecDemo　python=3.6

切换虚拟环境：conda activate SpeechRecDemo

依赖环境安装：pip install -r requirements.txt

新建 Django 项目：django-admin startproject SpeechRecDemo

1.2.2 项目配置

SpeechRecDemo/settings.py

（1）开放所有主机（本项目为小程序端）访问权限

```
ALLOWED_HOSTS = ['*']
```

（2）服务模块注册（后续说明相关服务作用）

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'service',
    'upload',
    'recognize',
]
```

（3）设置媒体文件夹，存放图片、音频等用户上传的静态文件

```
STATIC_URL = '/static/'
STATIC_LOCAL = os.path.join(BASE_DIR,'static')

MEDIA_URL = '/media/'
MEDIA_LOCAL = os.path.join(BASE_DIR,'media')

SPEECH_URL = MEDIA_URL + 'speech/'
SPEECH_LOCAL = os.path.join(MEDIA_LOCAL, "speech")
```

PS：在项目根目录下新建 static 和 media 文件夹

（4）时区、日期设置（可选）

```
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Asia/Shanghai'

USE_I18N = True

USE_TZ = False

USE_L10N = False

DATETIME_FORMAT = 'Y-m-d H:i:s'

DATE_FORMAT = 'Y-m-d'
```

1.2.3 服务模块创建

（1）创建 API 入口

django-admin startapp service

（2）创建子服务

文件上传模块：django-admin startapp upload

语音识别模块：django-admin startapp recognize

1.2.4 API 编写（以 upload 模块为例）

upload/views.py

新建 myUploadTest 函数，利用 request 对象获取 HTTP 请求的相关参数，返回 JSON 格式的数据交给前端处理。

```python
import json
from django.shortcuts import render
from django.http import HttpResponse
from django.views.decorators.csrf import csrf_exempt

@csrf_exempt
def myUploadTest(request):
    id = request.GET['id']
    res = {
        'code': 0,
        'msg': 'success',
        'data': {
            'testKey': 'Hello Upload, id: ' + id
        }
    }
    return HttpResponse(json.dumps(res))
```

1.2.5 路由配置

（1）在 SpeechRecDemo/urls.py，设置 API 服务的根 URL 和静态文件访问的 URL

```python
from django.conf.urls import url
from django.contrib import admin
from django.urls import path, include
from django.views.static import serve

from SpeechRecDemo import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^api/', include('service.urls')),
    url(r'^static/(?P<path>.*)$',serve, {'document_root': settings.STATIC_LOCAL}),
    url(r'^media/(?P<path>.*)$',serve, {'document_root': settings.MEDIA_LOCAL}),
]
```

（2）新建 service/urls.py，设置子服务 upload 和 recognize 的 URL 入口

```python
from django.urls import include, re_path

urlpatterns = [
    re_path(r'^upload/', include('upload.urls')),
    re_path(r'^recognize/', include('recognize.urls')),
]
```

（3）新建 upload/urls.py，链接到 views.py 对应函数（接口）

```python
from django.urls import re_path
from upload import views

urlpatterns = [
    re_path(r'^uploadTest$', views.myUploadTest),
]
```

1.2.6 recognize 模块

接口编写操作与 upload 模块类似

1.2.7 接口访问

启动 Web 服务：python manage.py runserver 0.0.0.0:8765

浏览器输入 localhost:8765/api/upload/uploadTest?id=2022，查看接口返回结果（浏览器访问属于 GET 请求，request 对象获取时需要区别 METHOD 为 POST、GET、PUT……）

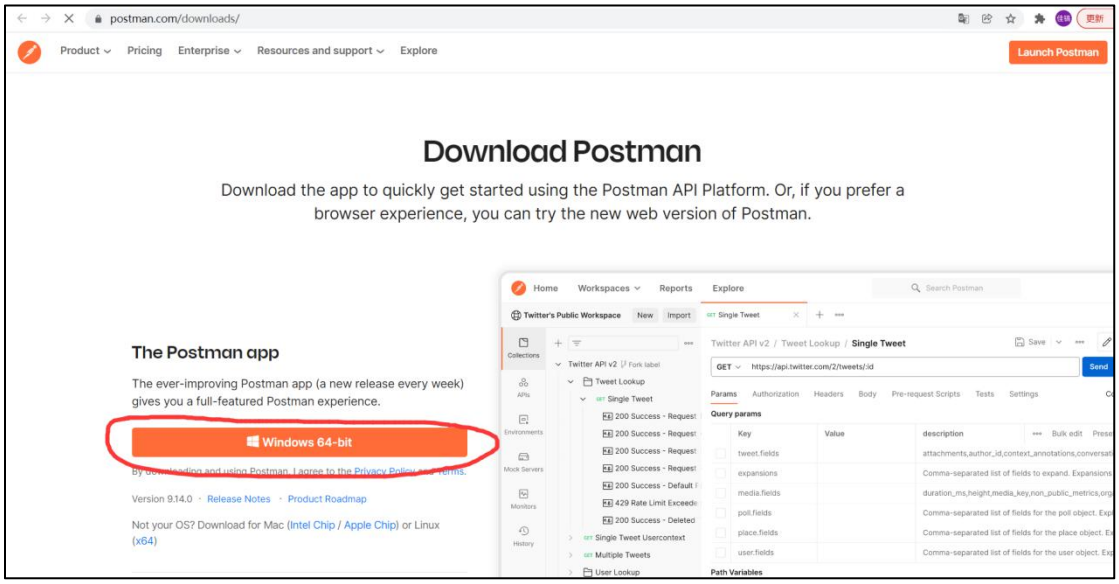
1.2.8 其他

后端常用数据库、缓存等技术，网站前后端分离可能需要注意跨域问题。

1.3 接口调试——以 Postman 工具为例

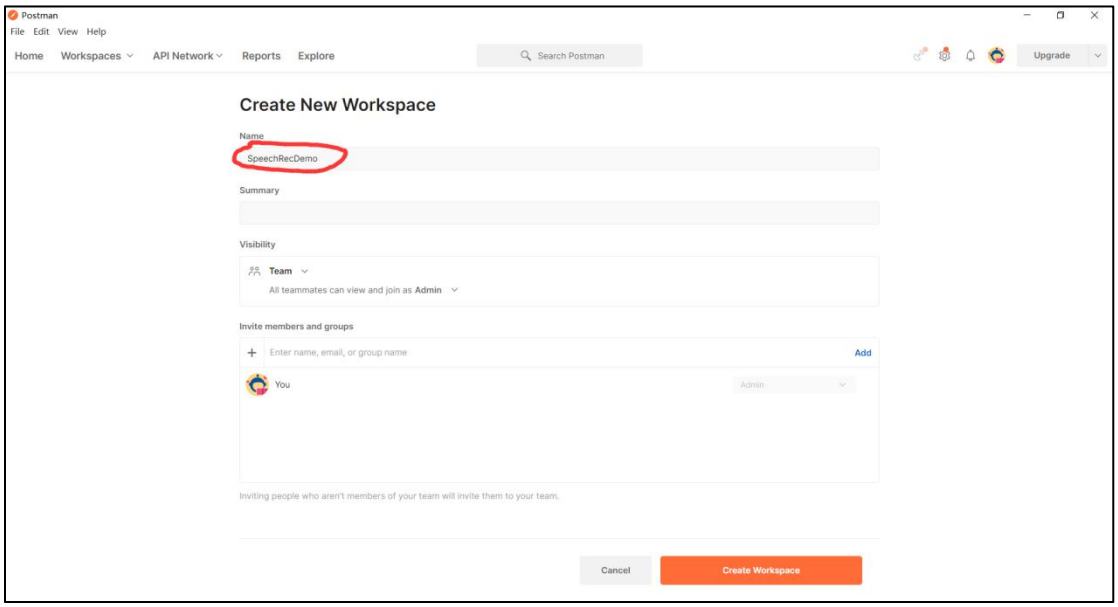
可以在线使用或安装客户端

1.3.1 客户端版安装

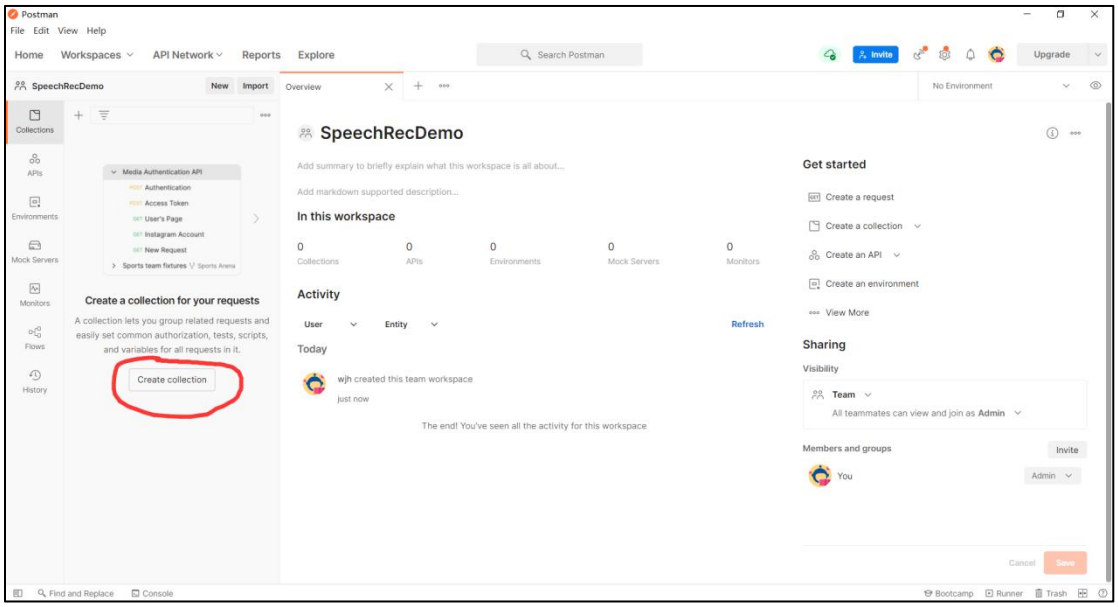


1.3.2 使用步骤

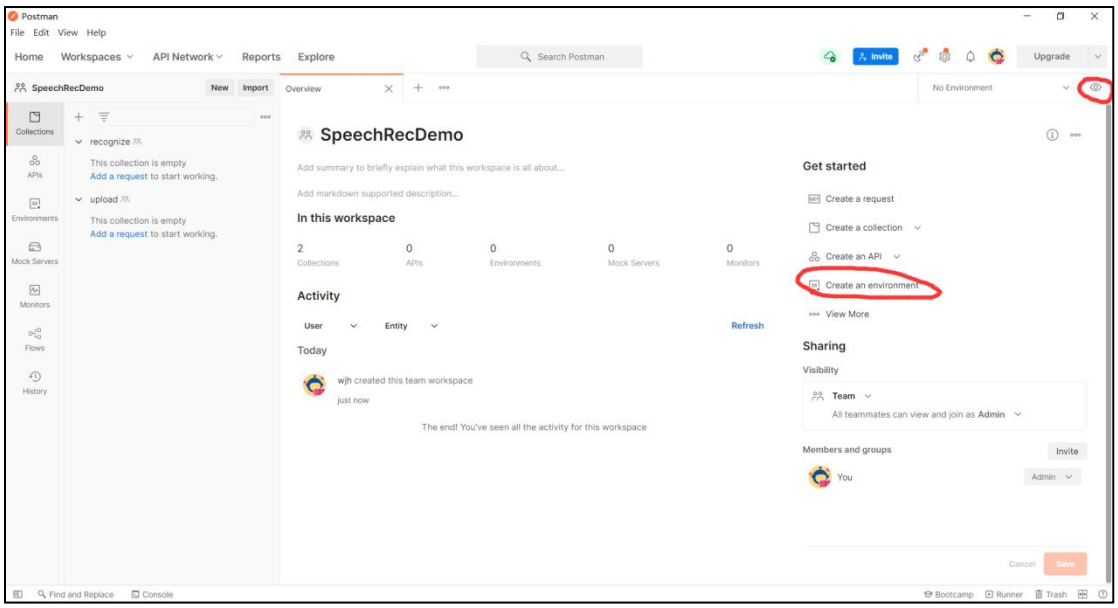
（1）创建工作区

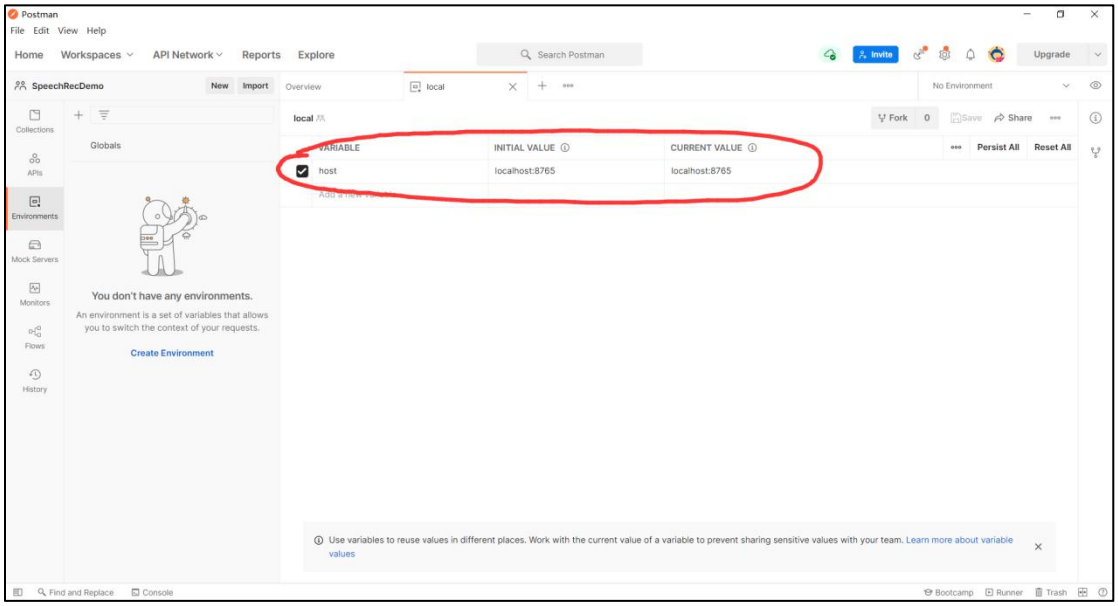


（2）创建测试集



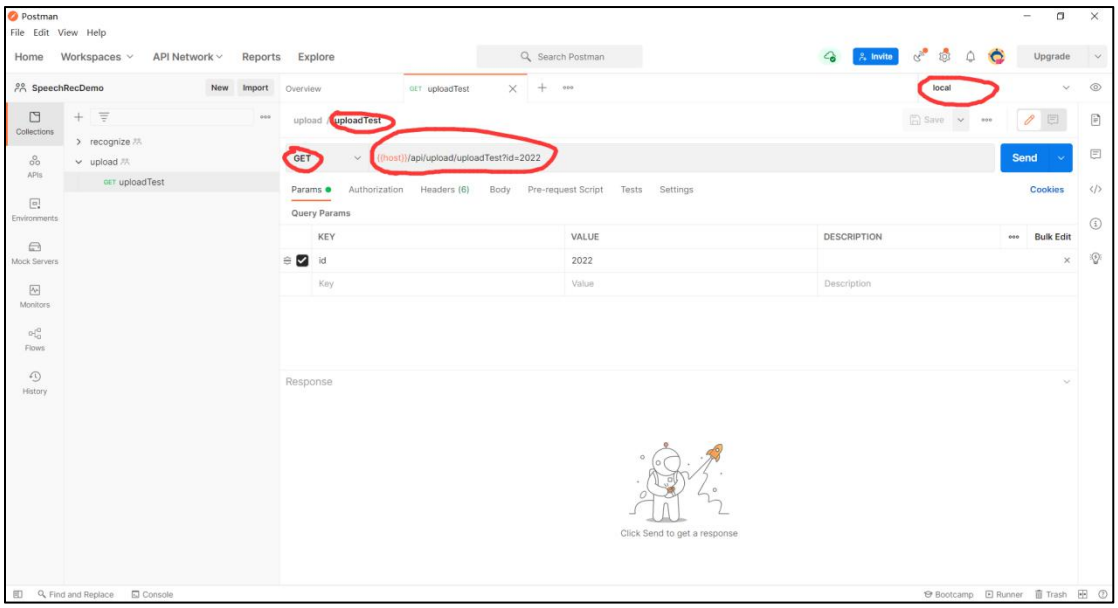
（3）创建测试环境（服务器地址）





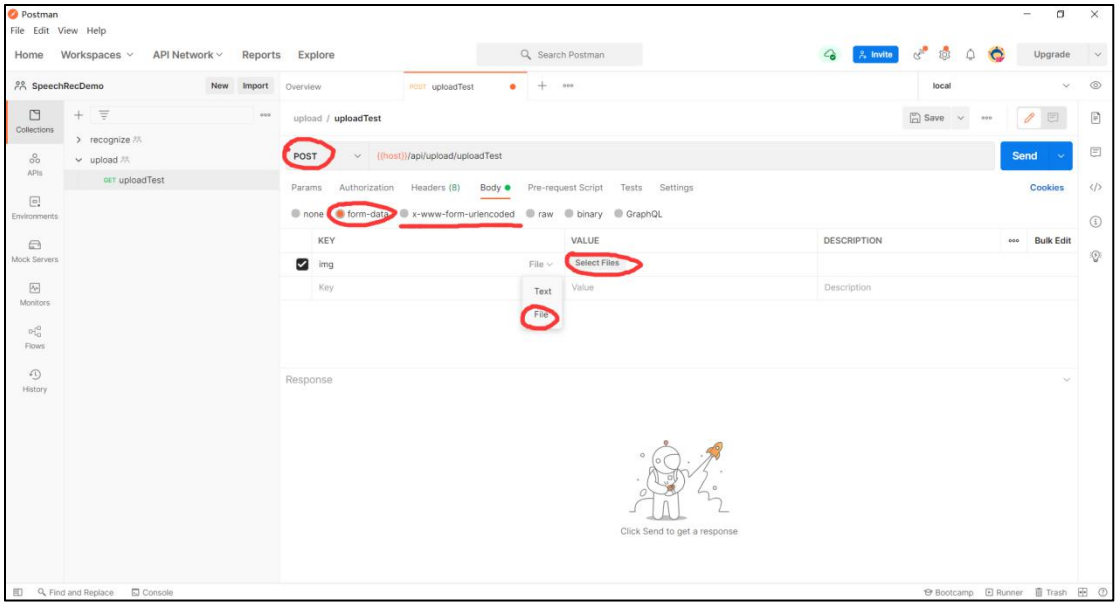
PS：新增线上环境时操作一致，使用时根据需要在本地和线上环境之间切换。

（4）创建请求



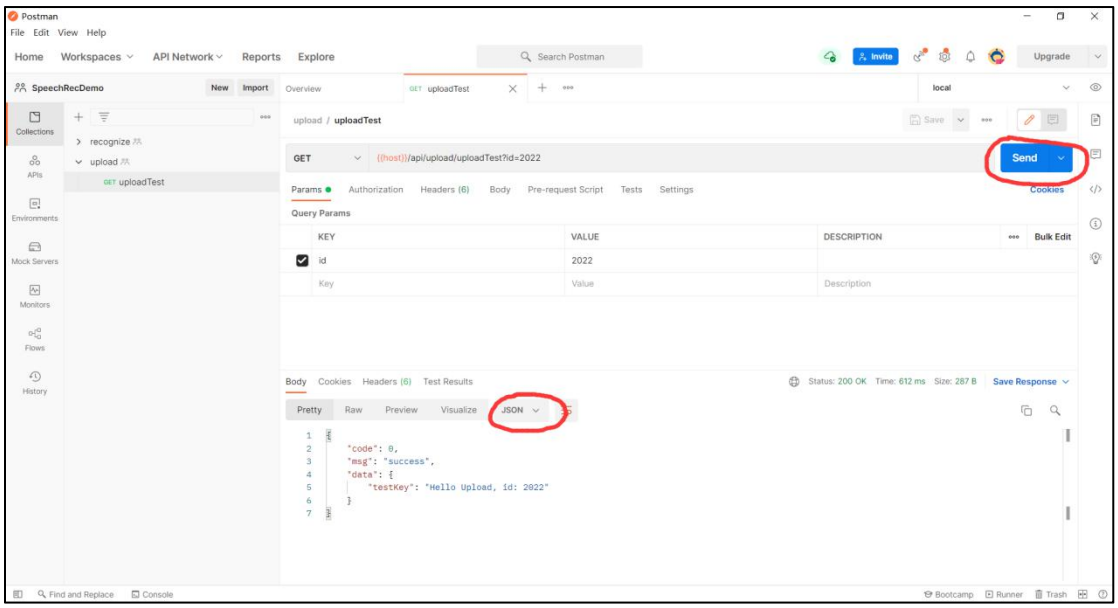
要素：环境选择，URL 设置，METHOD 设置

若请求为 POST 方式，且需要文件上传，请求设置如下：



PS：无需上传文件则选择 POST + x-www-form-urlencoded

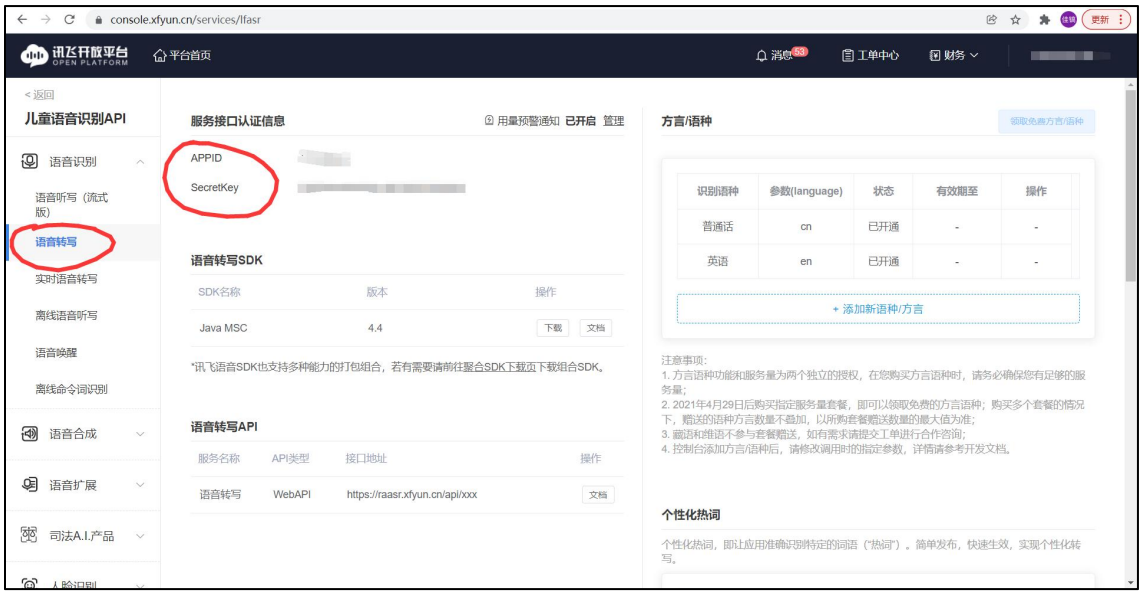
（5）发送请求



1.4 第三方服务——讯飞语音转写

https://console.xfyun.cn/services/lfasr

下载并修改官方给定 Demo，主要是配置 APPID 和 SecretKey



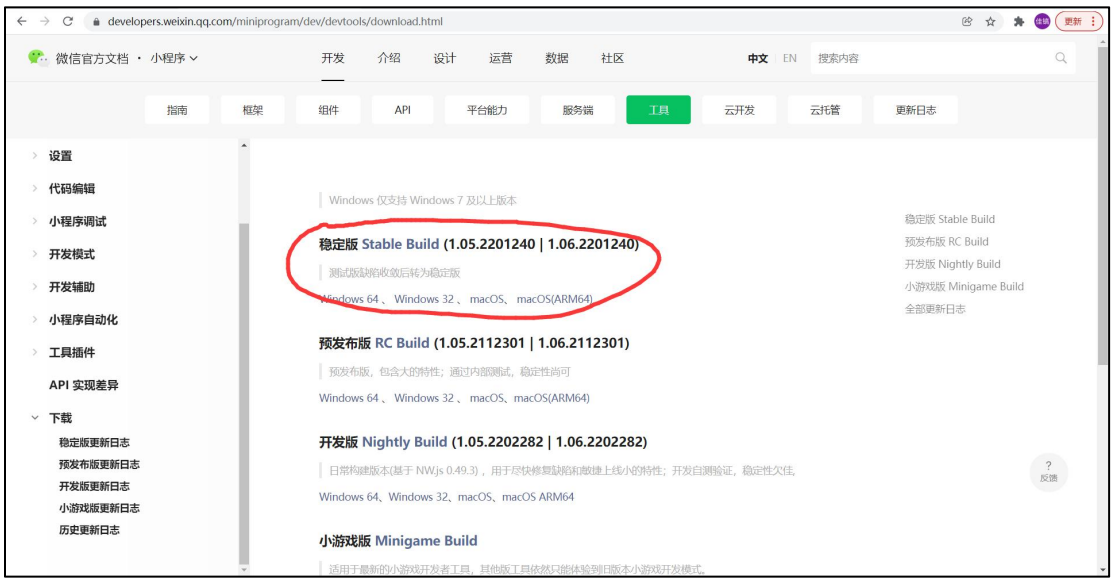
2 前端开发——以微信小程序为例

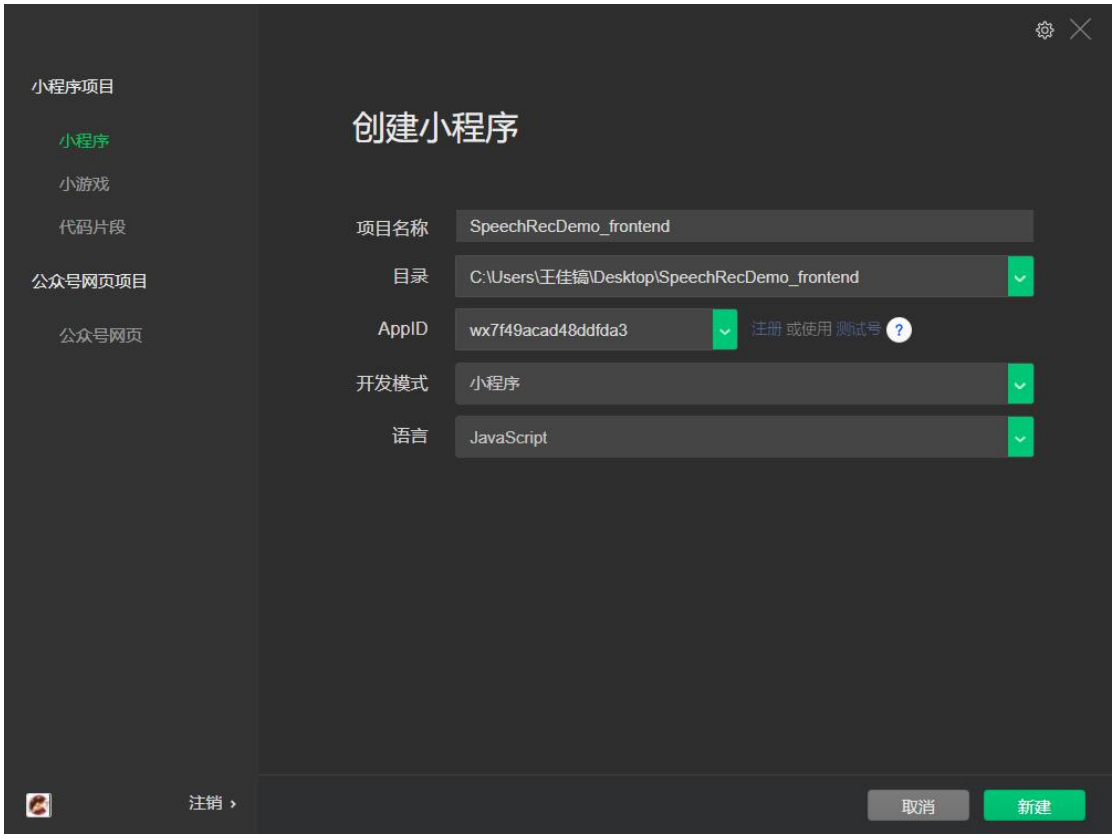
官方文档：https://developers.weixin.qq.com/miniprogram/dev/framework/

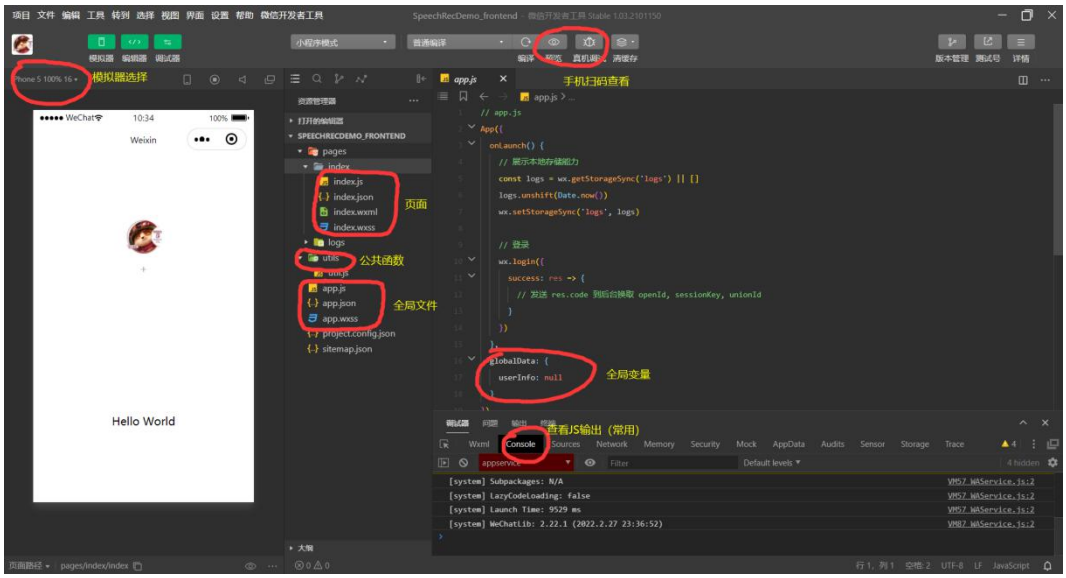
2.1 开发步骤

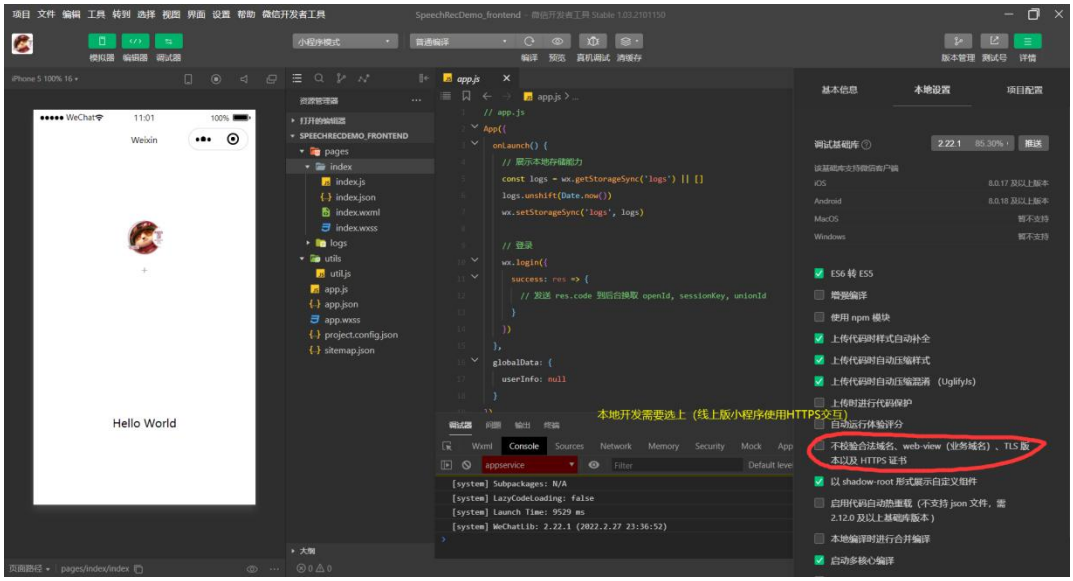
2.1.1 微信开发者工具相关说明

https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html

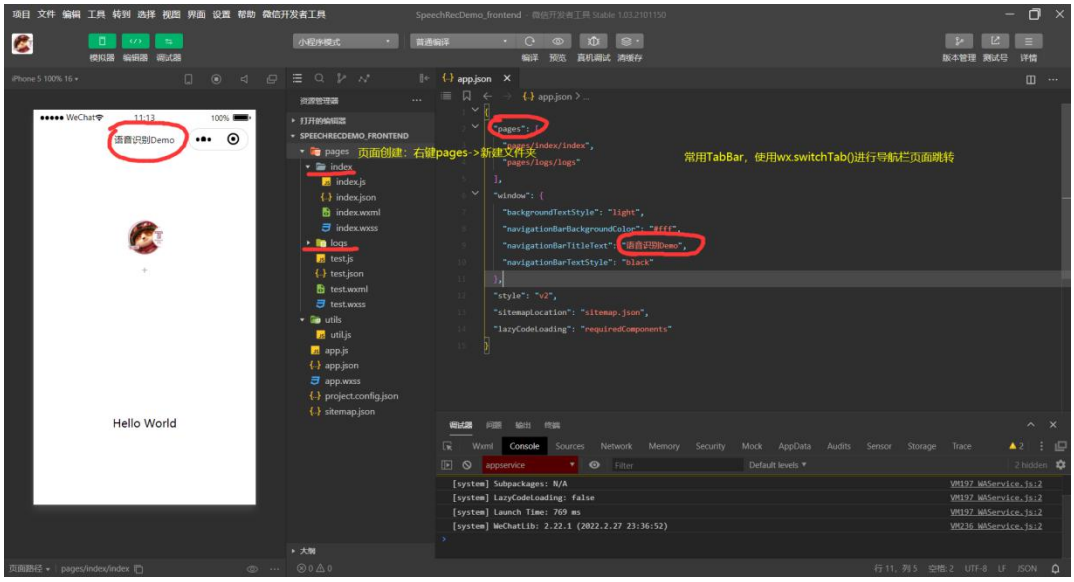





模拟器选择
手机扫码查看
页面
公共函数
全局文件
全局变量
查看JS输出（常用）
Hello World

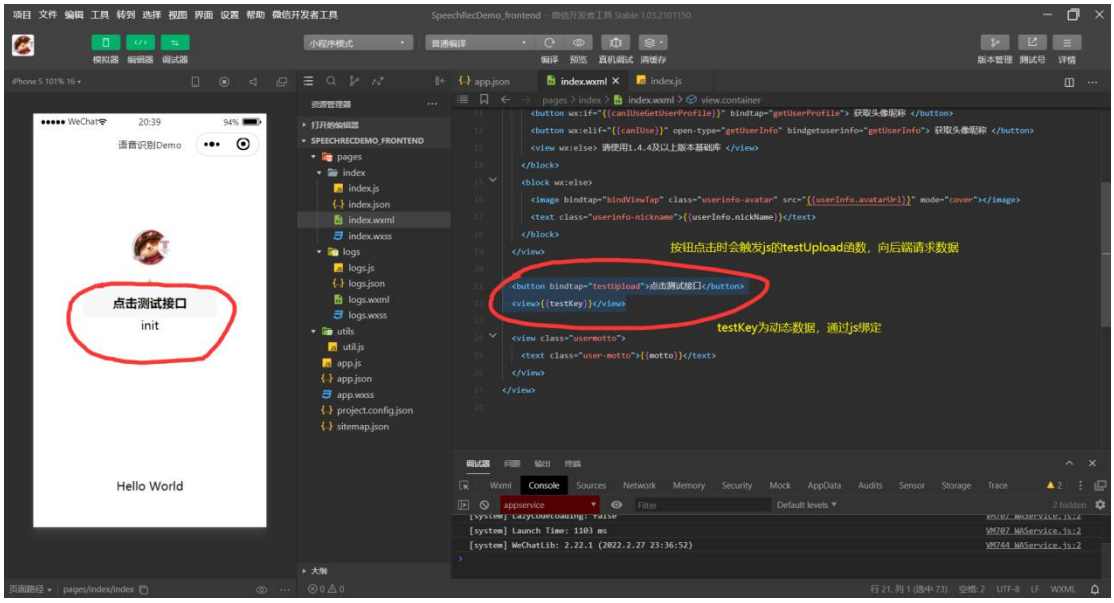

本地开发需要选上（线上版小程序使用HTTPS交互）
Hello World

2.1.2 页面整体配置


页面创建：右键pages->新建文件夹
常用TabBar，使用wx.switchTab()进行导航栏页面跳转
Hello World

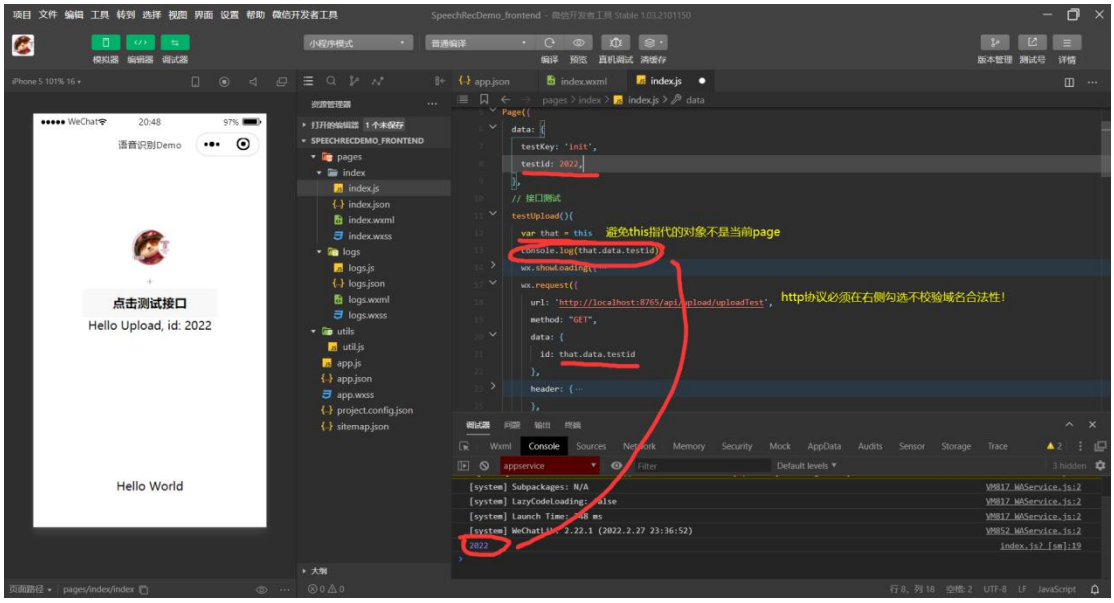
2.1.3 网络请求与数据绑定

wxml+wxss 文件设计界面，类似于 HTML+CSS



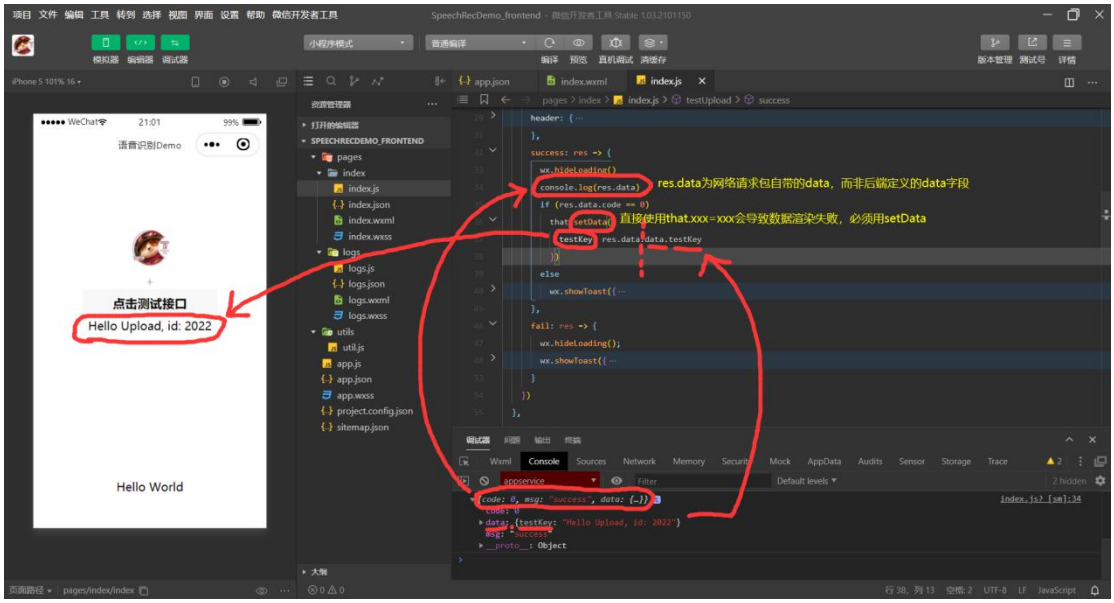
JS 文件处理前端业务逻辑，可认为是连接视图界面和后端的桥梁。

网络请求发送部分设置如下：



注意：真机调试时如果使用 localhost 会请求失败，请配置为真实的 IP 地址，cmd 输入 ipconfig 查看对应 IPV4 地址。真机和 PC 需要在同一个局域网中。

网络请求响应部分处理如下：



2.1.4 其他

界面设计：一般使用 iPhone 6 模拟器设计界面，复杂系统需要组件化编程（类似于函数式编程思想），官方组件推荐 WeUI 组件库（https://weui.io/）；常使用 flex 布局方式，在 wxss 中调整样式，宽度设计建议用 rpx 而非 px，适配不同大小的屏幕。推荐阿里巴巴矢量图库：https://www.iconfont.cn/

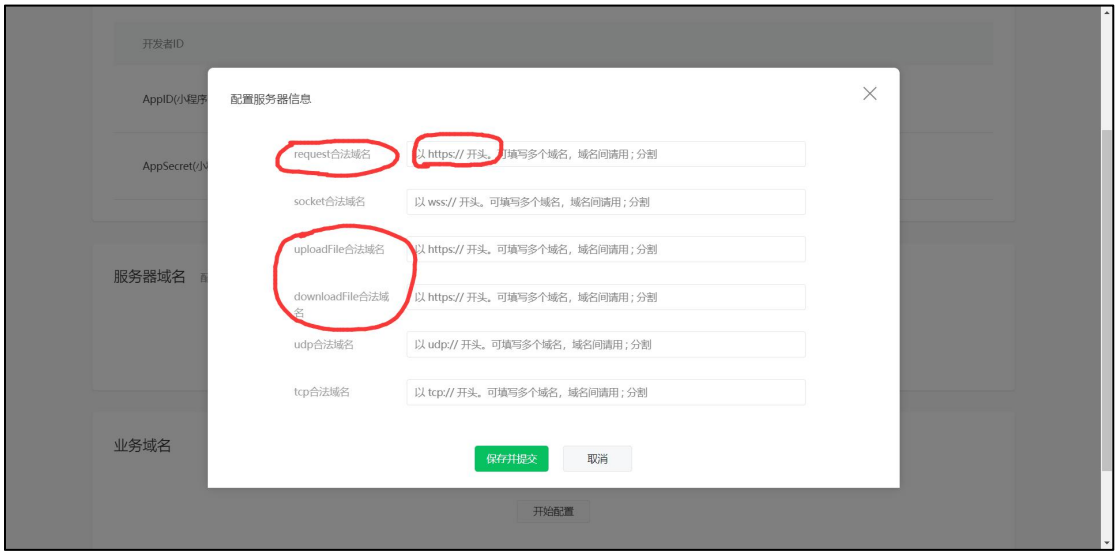
业务逻辑：JS 需要关注页面的生命周期函数，如 onLoad()、onShow()，提示框可使用 wx.showModal()、wx.showToast()，为了不让页面跳转过快等需求可以使用 JS 定时器技术。

真机调试：开发者工具的模拟器和真机有区别，最终需要在真机上运行，一切以真机调试结果为准。如模拟器录音文件上传后会出现问题，真机没有问题。

2.2 小程序后台

微信小程序后台管理：https://mp.weixin.qq.com/

域名配置一般用于线上环境，服务端必须支持 HTTPS 协议。



PS：小程序上线需要审核，涉及评论等业务需要企业开发主体（营业执照）。