# 不帕接口文档

**目录**

[不帕接口文档 1](#_Toc104235020)

[1. 连接测试接口 2](#_Toc104235021)

[1.1 接口信息 2](#_Toc104235022)

[1.2请求参数 2](#_Toc104235023)

[1.3示例返回 2](#_Toc104235024)

[2. 文件上传接口 4](#_Toc104235025)

[2.1 接口信息 4](#_Toc104235026)

[2.2请求参数 4](#_Toc104235027)

[2.3示例返回 4](#_Toc104235028)

[3. 语音诊断接口 6](#_Toc104235029)

[3.1 接口信息 6](#_Toc104235030)

[3.2请求参数 6](#_Toc104235031)

[3.3示例返回 6](#_Toc104235032)

[4. 视频诊断接口 8](#_Toc104235033)

[4.1 接口信息 8](#_Toc104235034)

[4.2请求参数 8](#_Toc104235035)

[4.3示例返回 8](#_Toc104235036)

# 连接测试接口

**1.1 接口信息**

请求路径： https://nopa.datahys.com:8000/connect

请求方法：POST

接口描述：用来快速的查询当前服务器是否可以连接，同时可以用来计算延迟。

**1.2请求参数**

通过 JSON 格式，共包含下面一个参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **说明** | **类型** | **是否必须** |
| send\_time | 使用机器时间  1970纪元后经过的浮点秒  例如Python中：  time.time() | Float或者 int | 是 |

**1.3示例返回**

**当状态码status\_code为200**

{'status': 'success',

 'TimeDelay': -37.486724615097046,

 'receive\_time': 1653274552.3377705}

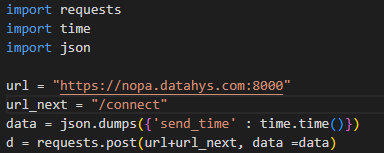
Status:状态，只会返回’success’

TimeDelay: 两台机器间的时间差（1970纪元后经过的浮点秒之差），可以计算客户端与服务器的时区差。

receive\_time: 服务器接收时间（1970纪元后经过的浮点秒之差）

**其余状态码为连接失败**

代码例子：



# 文件上传接口

**2.1 接口信息**

请求路径： https://nopa.datahys.com:8000/file\_save/<open\_id>

请求方法：POST

接口描述：用来给用户(对应的id为open\_id)上传文件。

注意:暂时open\_id用’test‘来代替

**2.2请求参数**

通过files格式，保护一个参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **说明** | **类型** | **是否必须** |
| file | 具体的文件  （音频最好是mp3格式的；视频最好是mp4格式的） | 文件 | 是 |

**2.3示例返回**

**当状态码status\_code为200且成功保存**

{'status': 'success',

 'message': '文件成功保存到./resources/received\_file/test/2022-05-23\_13-44-13.mp3',

 'file\_id': 142}

Status:状态，返回’success’时代表成功上传并保存文件

message: 会告诉我们将文件保存在服务器的路径。

file\_id: 本次上传文件的file\_id，（未来可以通过file\_id来对文件进行查询）

**当状态码status\_code为200但成功失败**

{'status': 'error',

 'message': '文件保存出错',

 'file\_id': 0}

Status:状态，返回’error’时代表上传文件成功但是保存文件失败

message: 会告诉我们文件保存出错。

file\_id: 出错的时候file\_id就是0

**其余状态码为上传失败**

代码示例

import requests

url = "https://nopa.datahys.com:8000"

url\_next = "/file\_save/test"

with open('./a.mp3', 'rb') as file:

    files = {'file': file}

    d = requests.post(url+url\_next,  files = files)

# 语音诊断接口

**3.1 接口信息**

请求路径： https://nopa.datahys.com:8000/diagnose\_speech

请求方法：POST

接口描述：用来给音频打分，判断是否有语音障碍（帕金森）。

**3.2请求参数**

通过json请求发送，共包含一个参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **说明** | **类型** | **是否必须** |
| file\_id | 文件的file\_id，在上传的时候已经返回给用户了 | String或者int | 是 |

**3.3示例返回**

**当状态码status\_code为200**

{'status': 'success',

 'PD': 0.2908493486041002,

 'feature': {'Jitter (local)': 3.614,

  'Jitter (local, absolute)': 0.000204779,

  'Jitter (rap)': 1.823,

  'Jitter (ppq5)': 1.828,

  'Jitter (ddp)': 5.468,

  'Shimmer (local)': 19.83,

  'Shimmer (local, dB)': 1.556,

  'Shimmer (apq3)': 8.227,

  'Shimmer (apq5)': 10.834,

  'Shimmer (apq11)': 15.374,

  'Shimmer (dda)': 24.681,

  'Mean autocorrelation': 0.787829,

  'Mean noise-to-harmonics ratio': 0.319497,

  'Mean harmonics-to-noise ratio': 6.965,

  'Median pitch': 118.849,

  'Mean pitch': 209.289,

  'Standard deviation': 178.245,

  'Minimum pitch': 87.105,

  'Maximum pitch': 607.533,

  'Number of pulses': 203.0,

  'Number of periods': 158.0,

  'Mean period': 0.005665839,

  'Standard deviation of period': 0.003360538,

  'Fraction of locally unvoiced frames': 79.147,

  'Number of voice breaks': 14.0,

  'Degree of voice breaks': 70.307}}

status:状态，返回’success’时代表诊断成功

PD: 告诉我们用户患有语音障碍的概率（小于0.5:”恭喜你很健康”，否则“有帕金森病的风险”）。

features: 提取出的音频特征

**当状态码不是200则诊断失败**

代码示例

import requests

import json

url = "https://nopa.datahys.com:8000"

url\_next = "/diagnose\_speech"

data = json.dumps({'file\_id' : 142})

d = requests.post(url+url\_next, data =data)

# 视频诊断接口

**4.1 接口信息**

请求路径： https://nopa.datahys.com:8000/diagnose\_face

请求方法：POST

接口描述：用来给视频打分，判断是否有表情障碍（帕金森）。

注意：检测时间较长（大约20s）

**4.2请求参数**

通过json请求发送，共包含一个参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **说明** | **类型** | **是否必须** |
| file\_id | 文件的file\_id，在上传的时候已经返回给用户了 | String或者int | 是 |

**4.3示例返回**

**当状态码status\_code为200，且status为success, error为False**

{'error': False,

 'landmark': [{'contour\_chin': {'x': 353, 'y': 859},

   'contour\_left1': {'x': 218, 'y': 560},

   'contour\_left10': {'x': 241, 'y': 755},

   'contour\_left11': {'x': 250, 'y': 775},

   'contour\_left12': {'x': 261, 'y': 795},

   'contour\_left13': {'x': 274, 'y': 813},

   'contour\_left14': {'x': 289, 'y': 830},

   'contour\_left15': {'x': 307, 'y': 845},

   'contour\_left16': {'x': 328, 'y': 855},

   'contour\_left2': {'x': 215, 'y': 581},

   'contour\_left3': {'x': 214, 'y': 603},

   'contour\_left4': {'x': 214, 'y': 625},

   'contour\_left5': {'x': 216, 'y': 646},

   'contour\_left6': {'x': 219, 'y': 668},

   'contour\_left7': {'x': 224, 'y': 690},

   'contour\_left8': {'x': 228, 'y': 712},

   'contour\_left9': {'x': 234, 'y': 734},

   'contour\_right1': {'x': 502, 'y': 560},

   'contour\_right10': {'x': 482, 'y': 763},

   'contour\_right11': {'x': 472, 'y': 784},

   'contour\_right12': {'x': 458, 'y': 803},

   'contour\_right13': {'x': 443, 'y': 821},

   'contour\_right14': {'x': 425, 'y': 837},

   'contour\_right15': {'x': 405, 'y': 850},

   ...],

 'no\_face\_time': 0.0,

 'PD': 0.3,

 'status': 'success'}

status:状态，返回’success’时代表请求成功

error: 诊断是否失败，返回“False”（诊断是否失败，失败的可能的原因有脸的时长过短，或者是网络波动等）

landmark: 提取的面部关键点位置。

no\_face\_time: 视频中没有检测到脸的时间长度

**当状态码status\_code为200，且status为success, error为True**

{'error': True,

 'error\_reason': '检测到面部出现的时长过短',

 'status': 'success'}

status:状态，返回’success’时代表请求成功

error: 诊断是否失败，返回“False”（诊断是否失败）

error\_reason: 诊断失败原因：

1. 录制时长太短 （视频小于4s）。
2. 检测接口有问题 （face++接口连通不畅）
3. 检测到面部出现的时长过短 （面部时长太短）

**当状态码不为200，检测失败**

代码示例

import requests

import json

url = "https://nopa.datahys.com:8000"

url\_next = "/diagnose\_face"

data = json.dumps({'file\_id' : 143})

d1 = requests.post(url+url\_next, data =data)

d = requests.post(url+url\_next, data =data)