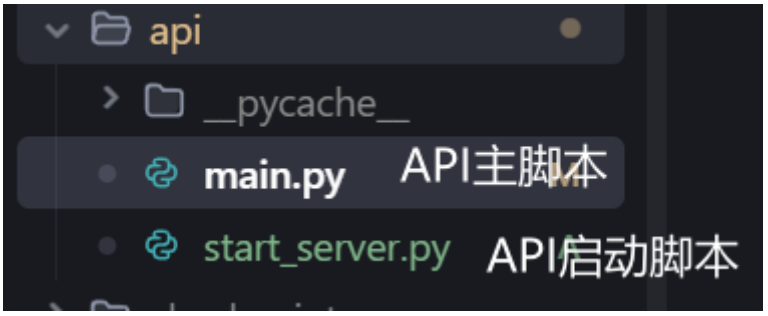
API说明书



API启动脚本运行后终端如下显示

```
INFO:     Will watch for changes in these directories: ['D:\\ZILIAO\\programstudy\\EE
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [16116] using WatchFiles
INFO:     Started server process [9316]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

```python
class TrainingConfig(BaseModel):
    method: str = 'LEAD'  # 训练方法
    task_name: str = 'pretrain_lead'  # 任务名称
    model: str = 'LEAD'  # 模型类型
    model_id: str = 'P-11-Base'  # 模型ID
    is_training: int = 0  # 训练标志0表示测试，1表示训练
    data: str = 'MultiDatasets'  # 数据类型
    root_path: str = './dataset/'  # 数据集根路径
    pretraining_datasets: str = 'ADSZ,APAVA-19,ADFSU,AD-Auditory,REEG-PD-19,PEARL-Neuro-19,Depression-19,REEG-SRM-19'  # 预训练数据集
    training_datasets: str = 'ADFTD'  # 训练数据集
    testing_datasets: str = 'ADFTD'  # 测试数据集
    checkpoints_path: str = './checkpoints/LEAD/pretrain_lead/LEAD/P-11-Base/'  # 检查点路径
    e_layers: int = 12  # 编码器层数
    batch_size: int = 512  # 批量大小
    n_heads: int = 8  # 注意力头数
    d_model: int = 128  # 模型维度
    d_ff: int = 256  # 全连接层维度
    swa: bool = True  # 是否使用SWA
    des: str = 'Exp'  # 实验描述
    itr: int = 5  # 实验次数
    learning_rate: float = 0.0002  # 学习率
    train_epochs: int = 60  # 训练轮次
    top_k: int = 5
    num_kernels: int = 6
    seq_len: int = 96
    enc_in: int = 7
    dec_in: int = 7
    c_out: int = 7
    moving_avg: int = 25
    factor: int = 1
    distil: bool = True
    dropout: float = 0.1
    embed: str = 'timeF'
    activation: str = 'gelu'
    output_attention: bool = False
    patch_len: int = 32
    stride: int = 8
    patch_len_list: str = '4'
    up_dim_list: str = '76'
    augmentations: str = 'flip,frequency,jitter,mask,channel,drop'
    no_inter_attn: bool = False
    no_temporal_block: bool = False
    no_channel_block: bool = False
    K: int = 65536
    momentum: float = 0.999
    temperature: float = 0.07
    mask_ratio: float = 0.5
    contrastive_loss: str = 'all'
    num_workers: int = 0
    patience: int = 3
    loss: str = 'MSE'
    lradj: str = 'type1'
    use_amp: bool = False
    no_normalize: bool = False
    sampling_rate: int = 128
    low_cut: float = 0.5
    high_cut: float = 45
    cross_val: str = 'fixed'
    use_gpu: bool = True
    gpu: int = 0
    use_multi_gpu: bool = True
    devices: str = '0'
    p_hidden_dims: List[int] = [128, 128]
    p_hidden_layers: int = 2
```

API默认的模型参数，最好别在这修改模型参数

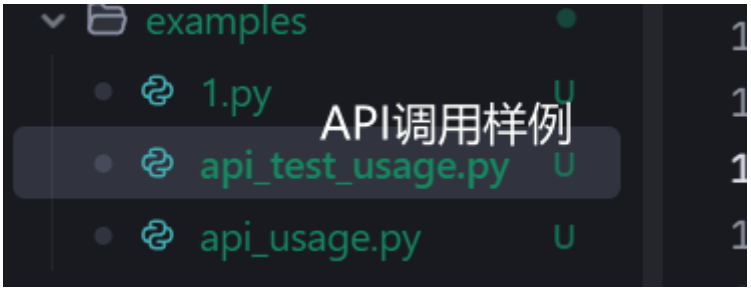
```python
root_path: str = './dataset/'  # 数据集根路径
pretraining_datasets: str = 'ADSZ,APAVA-19,ADFSU,AD-Auditory,REEG-PD-19,PEARL-Neuro-19,Depres
training_datasets: str = 'ADFTD'  # 训练数据集
testing_datasets: str = 'ADFTD'  # 测试数据集
checkpoints_path: str = './checkpoints/LEAD/pretrain_lead/LEAD/P-11-Base/'  # 模型路径
```

数据集存放的位置

数据集名称(文件夹名称)

模型的位置(最后一个是模型的文件夹名称)

以下是**api**调用示例




```python
import requests
import json

BASE_URL = "http://localhost:8000"

# 测试模型
def test_model():
    test_config = {
        "checkpoints_path": "./checkpoints/LEAD/pretrain_lead/LEAD/P-11-Base/",
        "testing_datasets": "ADFTD",
        "root_path": './dataset/'
    }
    response = requests.post(f"{BASE_URL}/test", json=test_config)
    return response.json()

if __name__ == "__main__":
    # 测试模型
    print("开始测试模型: ")
    test_results = test_model()
    print(json.dumps(test_results, indent=2, ensure_ascii=False))
```

如果需要改调用模型的参数，在这里改

示例的三个分别是模型路径、测验数据集名称、测验数据集存放位置

注意有关训练的具体参数不要随意修改，请和有骐沟通，参数改变，导入的模型也要动

以下是返回的信息

格式是一个字典，需要的信息包含在**results**键中，具体需要哪些信息和有骐讨论一下


```
{
 "status": "success",
 "message": "模型测试完成",
 "": {
  "sample_val": {
   "Accuracy": [
    74.46093749999999,
    1.3508143476642038
   ],
```

```
    "Precision": [
      74.6002146598087,
      1.4587996308977929
    ],
    "Recall": [
      74.50633955227624,
      1.37127188599804
    ],
    "F1": [
      74.44597158388045,
      1.3375822755599136
    ],
    "AUROC": [
      79.68440099546635,
      1.7794912679658756
    ],
    "AUPRC": [
      76.98751554611849,
      2.0037910106382237
    ]
  },
  "subject_val": {
    "Accuracy": [
      76.92307692307693,
      0.0
    ],
    "Precision": [
      77.38095238095238,
      0.0
    ],
    "Recall": [
      77.38095238095238,
      0.0
    ],
    "F1": [
      76.92307692307692,
      0.0
    ],
    "AUROC": [
      77.3809523809524,
      0.0
    ],
    "AUPRC": [
```

```
    74.9084249084249,
    0.0
   ]
  },
  "sample_test": {
   "Accuracy": [
    77.28236607142858,
    1.2792778578291302
   ],
   "Precision": [
    78.36492856472924,
    1.316615659746954
   ],
   "Recall": [
    76.17082512861221,
    1.3311002829916407
   ],
   "F1": [
    76.41378022091182,
    1.3677348922289712
   ],
   "AUROC": [
    86.78910190007194,
    1.2691163706335358
   ],
   "AUPRC": [
    86.11024744824867,
    1.2290158613985034
   ]
  },
  "subject_test": {
   "Accuracy": [
    85.71428571428571,
    0.0
   ],
   "Precision": [
    90.0,
    0.0
   ],
   "Recall": [
    83.33333333333333,
    0.0
   ],
```

```
      "F1": [
        84.44444444444447,
        1.1102230246251565e-14
      ],
      "AUROC": [
        83.33333333333334,
        0.0
        0.0
      ],
      "AUPRC": [
        80.0,
        0.0
      ]
    }
  }
}
```