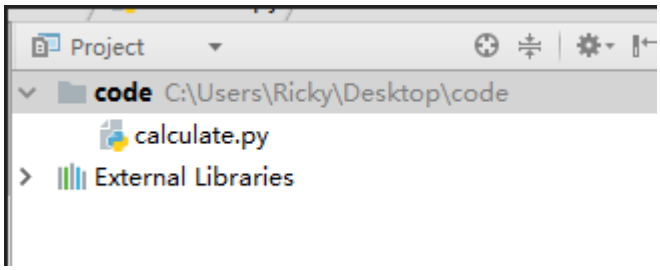
快速入门

定义功能函数

创建一个文件例如求2个数的和 calculate.py

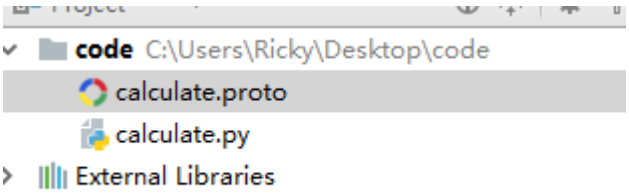
```
def sumnum(num1,num2):
    return num1+num2
```



创建.proto文件

创建calculate.proto文件

```
syntax = "proto3";

package rpc_package;

// 定义服务
service SumService {
    // 定义接口和类型
    rpc Sum (SumRequest) returns (SumReply) {}
}

// 定义请求
message SumRequest {
    int32 num1 = 1;
    int32 num2 = 2;
}

// 定义响应
message SumReply {
    int32 total = 1;
}
```



生成gRPC类

我们将使用特殊工具自动生成类。

安装protobuf编译器和grpc库

```
pip install grpcio-tools
```

编译生成代码

```
python -m grpc_tools.protoc -I. --python_out=. --grpc_python_out=. calculate.proto
```

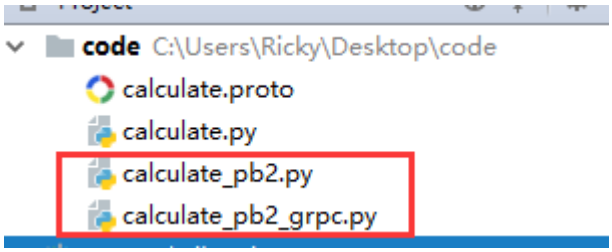
- `-I` 表示搜索proto文件中被导入文件的目录
- `--python_out` 表示保存生成Python文件的目录，生成的文件中包含接口定义中的数据类型
- `--grpc_python_out` 表示保存生成Python文件的目录，生成的文件中包含接口定义中的服务类型

执行编译

```
(toutiao) [root@toutiao-web code]# python -m grpc_tools.protoc -I. --python_out=. --
grpc_python_out=. calculate.proto
```

编译之后会自动生成如下两个rpc调用辅助代码模块：

- calculate_pb2.py 保存根据接口定义文件中的数据类型生成的python类（包含message）
- calculate_pb2_grpc.py 保存根据接口定义文件中的服务方法类型生成的python调用RPC方法（包含server）

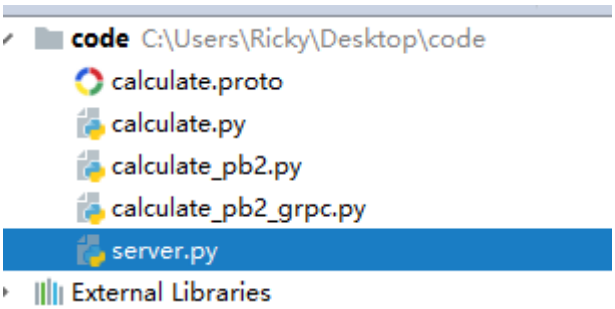


创建gRPC服务器

创建server.py文件

```
#!/usr/bin/env python
# -*-coding: utf-8 -*-
from concurrent import futures
import grpc
import time
from calculate import sumnum

from calculate_pb2_grpc import add_SumServiceServicer_to_server,SumServiceServicer
```

```
from calculate_pb2 import SumReply
class Service(SumServiceServicer):
    # 这里实现我们定义的接口
    def Sum(self, request, context):
        #设置响应
        response=SumReply()
        response.total=sumnum(request.num1,request.num2)
        return response


def serve():
    # 这里通过thread pool来并发处理server的任务
    server = grpc.server(futures.ThreadPoolExecutor(max_workers=10))

    # 将对应的任务处理函数添加到rpc server中
    add_SumServiceServicer_to_server(Service(), server)

    # 这里使用的非安全接口，世界gRPC支持TLS/SSL安全连接，以及各种鉴权机制
    server.add_insecure_port('[::]:50000')
    server.start()
    try:
        while True:
            time.sleep(60 * 60 * 24)
    except KeyboardInterrupt:
        server.stop(0)

if __name__ == "__main__":

    serve()
```



创建gRPC客户端

创建client.py文件

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import grpc
from calculate_pb2 import SumRequest
from calculate_pb2_grpc import SumServiceStub

def run():
    num1=10

    num2=20
```

```
    # 使用with语法保证channel自动close
    with grpc.insecure_channel('localhost:50000') as channel:
        # 客户端通过stub来实现rpc通信
        stub = SumServiceStub(channel)

        # 客户端必须使用定义好的类型，这里是HelloRequest类型
        response = stub.Sum(SumRequest(num1=num1,num2=num2))
    print ("num1:{} + num2:{} = {} ".format(num1,num2,response.total))

if __name__ == "__main__":
    run()
```