

Formulário - GrpcClient

```
rpc Op1 (Request) returns (Response);

rpc Op2 (Request) returns (stream Response);

rpc Op3 (stream Request) returns (Response);

rpc Op4 (stream Request) returns (stream Response);
```

```
// imports

public class App {
    private static String svcIP = "localhost";
    private static int svcPort = 8000;
    private static ManagedChannel channel;
    private static NameServiceGrpc.NameServiceBlockingStub blockingStub;
    private static NameServiceGrpc.NameServiceStub noBlockStub;

    public static void main(String[] args) {
        try {
            if (args.length == 2) {
                svcIP = args[0];
                svcPort = Integer.parseInt(args[1]);
            }
            System.out.println("connected to " + svcIP + ":" + svcPort);
            channel = ManagedChannelBuilder.forAddress(svcIP, svcPort)
                .usePlaintext()
                .build();

            blockingStub = NameServiceGrpc.newBlockingStub(channel);
            noBlockStub = NameServiceGrpc.newStub(channel);

            // ...

            Thread.sleep(10000); // 10 seconds

            channel.shutdown();
        } catch (Exception ex) {
            System.out.println("Unhandled exception");
            ex.printStackTrace();
        }
    }

    static void op1() {
        Request request = Request.newBuilder().build();
        Response response = blockingStub.op1(request);
    }

    static void op2() {
        Request request = Request.newBuilder().build();
        noBlockStub.op2(request, new StreamObserver<Response>() {
            @Override
            public void onNext(Response response) {
                // process response
            }
        })
    }
}
```

```

        @Override
        public void onError(Throwable throwable) {
            // handle error
        }

        @Override
        public void onCompleted() {
            // handle completion
        }
    });
}

static void op3() {
    StreamObserver<Response> responseObserver = new StreamObserver<Response>() {
        @Override
        public void onNext(Response response) {
            // process response
        }

        @Override
        public void onError(Throwable throwable) {
            // handle error
        }

        @Override
        public void onCompleted() {
            // handle completion
        }
    };

    StreamObserver<Request> requestObserver = noBlockStub.op3(responseObserver);
    requestObserver.onNext(Request.newBuilder().build());
    requestObserver.onCompleted();
}

static void op4() {
    StreamObserver<Response> responseObserver = new StreamObserver<Response>() {
        @Override
        public void onNext(Response response) {
            // process response
        }

        @Override
        public void onError(Throwable throwable) {
            // handle error
        }

        @Override
        public void onCompleted() {
            // handle completion
        }
    };

    StreamObserver<Request> requestObserver = noBlockStub.op4(responseObserver);
    requestObserver.onNext(Request.newBuilder().build());
    requestObserver.onCompleted();
}
}

```

