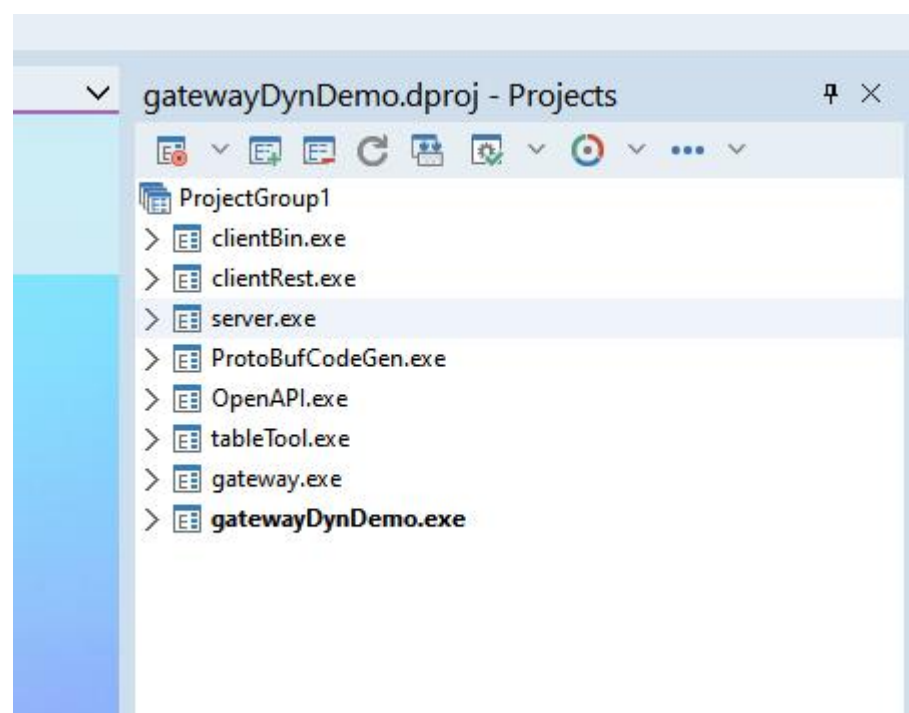


DELPHI 微服务架构

目录

工程组组成	1
工程组说明	1
IDL	2
IDL 发展史	2
RESTFUL API	2
JSON	3
PROTOBUF	3
PROTO 模板文件	4
微服务网关	4
数据表或 SQL 生成 RESTFUL CRUD PASCAL 代码工具	4
proto 模板文件翻译为 pascal 代码工具	5
自动生成 SWAGGER 在线接口文档工具	5
服务自动发现演示	6
微服务例子	6

工程组组成



工程组说明

clientBin.exe (传统二进制客户端)

clientRest.exe (rest 客户端)

Server.exe (微服务例子)

ProtoBufCodeGen.exe (.proto 模板文件翻译为 pascal 代码工具)

OpenAPI.exe (自动生成 SWAGGER 在线接口文档工具)

tableTool.exe (数据表或 SQL 生成 RESTFUL CRUD PASCAL 代码)

Gateway.exe (微服务网关)

gatewayDynDemo.exe（服务自动发现演示）

微服务架构，数据和接口使用 IDL 标准。

IDL

IDL 是 Interface description language 的缩写，指接口描述语言，是跨平台开发的基础。

IDL 是用来描述软件组件接口的一种计算机语言。IDL 通过一种中立的方式来描述接口和数据，使得在不同平台上运行的对象和用不同语言编写的程序可以相互通信交流；比如，一个组件用 C++ 写成，另一个组件用 Java 写成。

IDL 文件描述数据和接口，而服务对象则为一个指定的对象实现提供上述数据和接口。

IDL 文件描述了服务器提供的服务功能，客户机可以根据该接口文件描述的方法向服务器提出业务请求。

IDL 发展史

历史上曾经出现过的 IDL（排名不分先后）：CORBA、COM、COM+、WEBSERVICE、RESTFUL、gRPC、THRIFT 等。

RESTFUL API

REST 全称：REpresentational State Transfer，英文翻译过来就是“表现层状态转化”。

RESTful：用 URL 定位资源、用 HTTP 动词（GET、POST、PUT、DELETE）描述操作。

Resource：资源，即数据。

Representational：某种表现形式，比如用 JSON，XML，JPEG 等；

State Transfer：状态变化。通过 HTTP 动词实现。

URI（统一资源标识符）：可以唯一标识一个资源。

RESTful API 就是 REST 风格的 API，即 rest 是一种架构风格，跟编程语言无关，跟平台无关，采用 HTTP 做传输协议。

RESTful API 就是一套协议来规范多种形式的前端和同一个后台的交互方式。

RESTful API 由后台也就是 SERVER 来提供前端来调用。前端调用 API 向后台发起 HTTP 请求，后台响应请求将处理结果反馈给前端。也就是说 RESTful 是典型的基于 HTTP 的协议。那么 RESTful API 有哪些设计原则和规范呢？

1、资源。首先是弄清楚资源的概念。资源就是网络上的一个实体，一段文本，一张图片或者一首歌曲。资源总是要通过一种载体来反应它的内容。文本可以用 TXT，也可以用 HTML 或者 XML、图片可以用 JPG 格式或者 PNG 格式，JSON 是现在最常用的资源表现形式。

2、统一接口。RESTful 风格的数据元操 CRUD（create, read, update, delete）分别对应 HTTP 方法：GET 用来获取资源，POST 用来新建资源（也可以用于更新资源），PUT 用来

更新资源，DELETE 用来删除资源，这样就统一了数据操作的接口。

3、URI。可以用一个 URI（统一资源定位符）指向资源，即每个 URI 都对应一个特定的资源。要获取这个资源访问它的 URI 就可以，因此 URI 就成了每一个资源的地址或识别符。一般的，每个资源至少有一个 URI 与之对应，最典型的 URI 就是 URL。

4、无状态。所谓无状态即所有的资源都可以 URI 定位，而且这个定位与其他资源无关，也不会因为其他资源的变化而变化。有状态和无状态的区别，举个例子说明一下，例如要查询员工工资的步骤为第一步：登录系统。第二步：进入查询工资的页面。第三步：搜索该员工。第四步：点击姓名查看工资。这样的操作流程就是有状态的，查询工资的每一个步骤都依赖于前一个步骤，只要前置操作不成功，后续操作就无法执行。如果输入一个 URL 就可以得到指定员工的工资，则这种情况就是无状态的，因为获取工资不依赖于其他资源或状态，且这种情况下，员工工资是一个资源，由一个 URL 与之对应可以通过 HTTP 中的 GET 方法得到资源，这就是典型的 RESTful 风格。

在 rest 中会通过向服务器提交的请求类型来表示增删改查这些操作

GET (SELECT)：从服务器取出资源。

POST (CREATE)：在服务器新建一个资源。

PUT (UPDATE)：在服务器更新资源。

DELETE (DELETE)：从服务器删除资源。

RESTful API 还有其他一些规范。

①、应该将 API 的版本号放入 URL。GET: <http://www.xxx.com/v1/friend/123>。或者将版本号放在 HTTP 头信息中。我个人觉得要不要版本号取决于自己开发团队的习惯和业务的需要，不是强制的。

②、URL 中只能有名词而不能有动词，操作的表达是使用 HTTP 的动词 GET,POST,PUT,DELETE。URL 只标识资源的地址，既然是资源那就是名词了。

③、如果记录数量很多，服务器不可能都将它们返回给用户。API 应该提供参数，过滤返回结果。?limit=10：指定返回记录的数量、?page=2&per_page=100：指定第几页，以及每页的记录数。

JSON

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式。易于人阅读和编写，可以在多种语言之间进行数据交换 [5]。同时也易于机器解析和生成。它基于 JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 的一个子集。

JSON 是 Douglas Crockford 在 2001 年开始推广使用的数据格式，在 2005 年-2006 年正式成为主流的数据格式，雅虎和谷歌就在那时候开始广泛地使用 JSON 格式。

PROTOBUF

Google 的数据序列化协议：ProtoBuf (Protocol Buffers)。

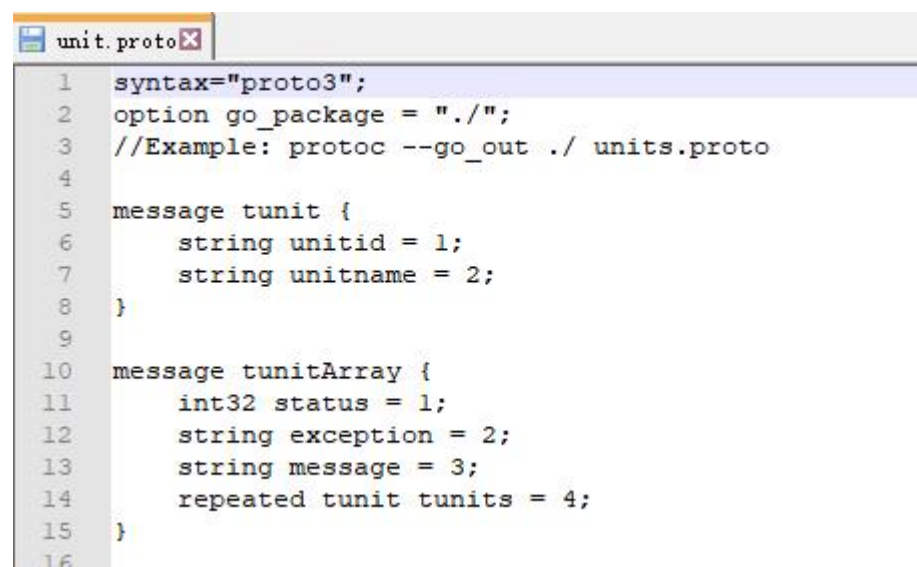
数据序列化协议用于将数据结构序列化成为方便存储、传输的格式，ProtoBuf 是一个被广泛使用的序列化协议，它拥有下面一些优点：

1. 二进制协议，相比 json 或 xml 而言体积小
2. 使用方便，使用 proto 工具直接生成可使用的代码

使用 ProtoBuf 的步骤可以分为这么几步：

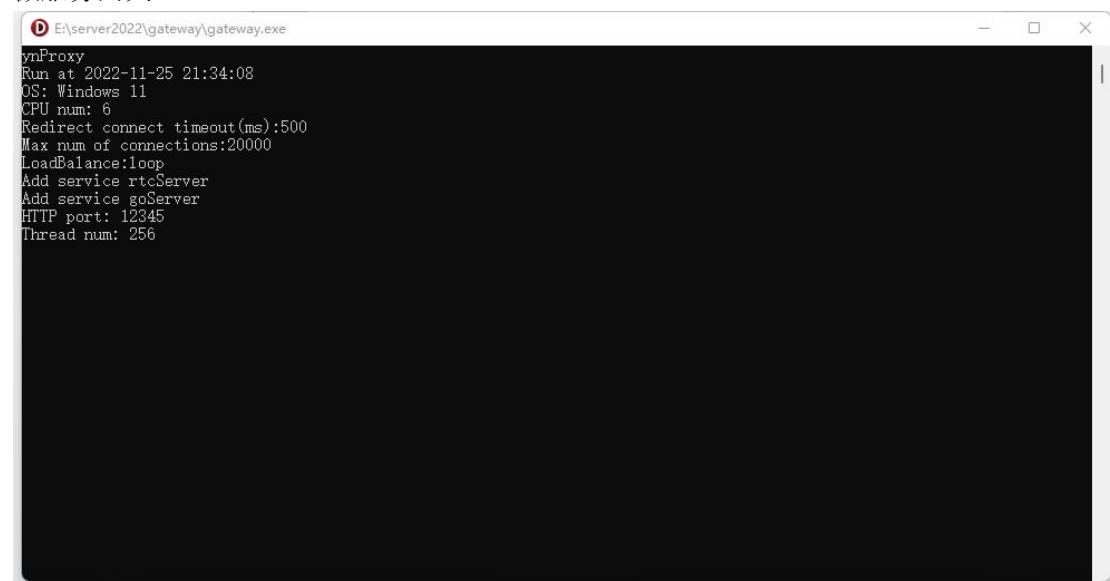
1. 定义描述文件 *.proto
2. 生成用于操作数据的代码
3. 调用生成的代码进行序列化或者反序列化

PROTO 模板文件



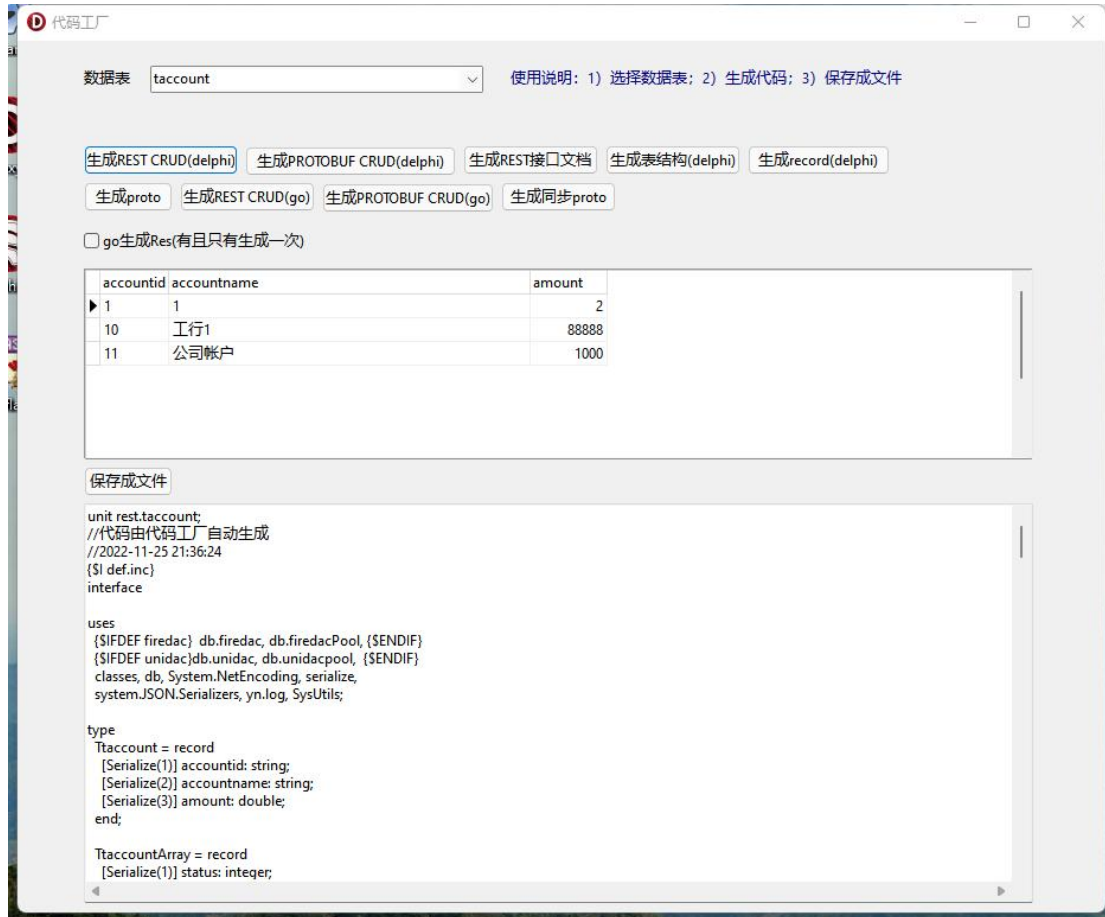
```
1 syntax="proto3";
2 option go_package = "./";
3 //Example: protoc --go_out ./ units.proto
4
5 message tunit {
6     string unitid = 1;
7     string unitname = 2;
8 }
9
10 message tunitArray {
11     int32 status = 1;
12     string exception = 2;
13     string message = 3;
14     repeated tunit tunits = 4;
15 }
16
```

微服务网关

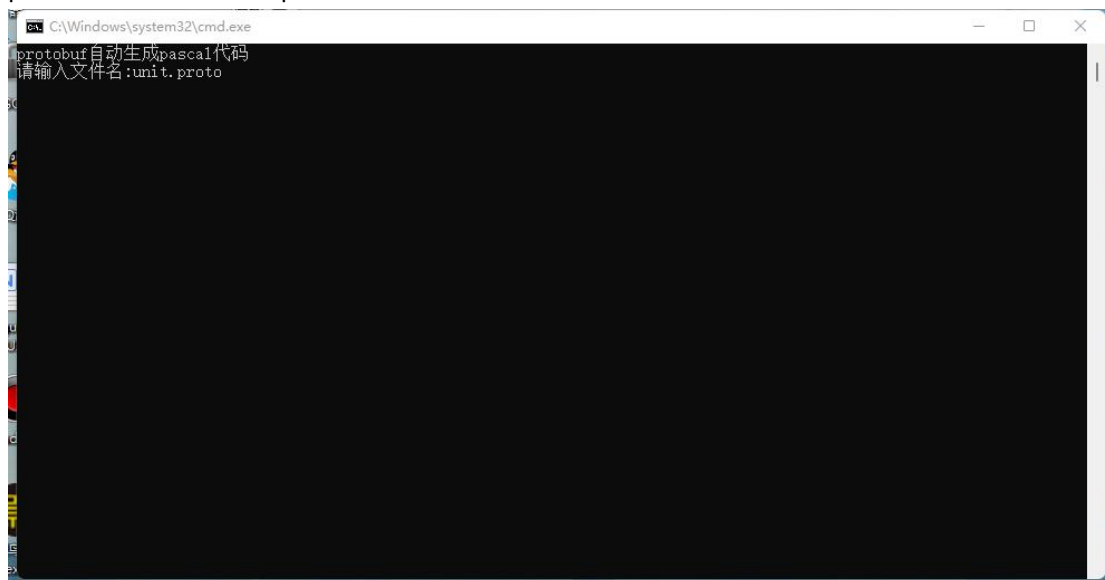


```
E:\server2022\gateway\gateway.exe
ynProxy
Run at 2022-11-25 21:34:08
OS: Windows 11
CPU num: 6
Redirect connect timeout(ms):500
Max num of connections:20000
LoadBalance:loop
Add service rtcServer
Add service goServer
HTTP port: 12345
Thread num: 256
```

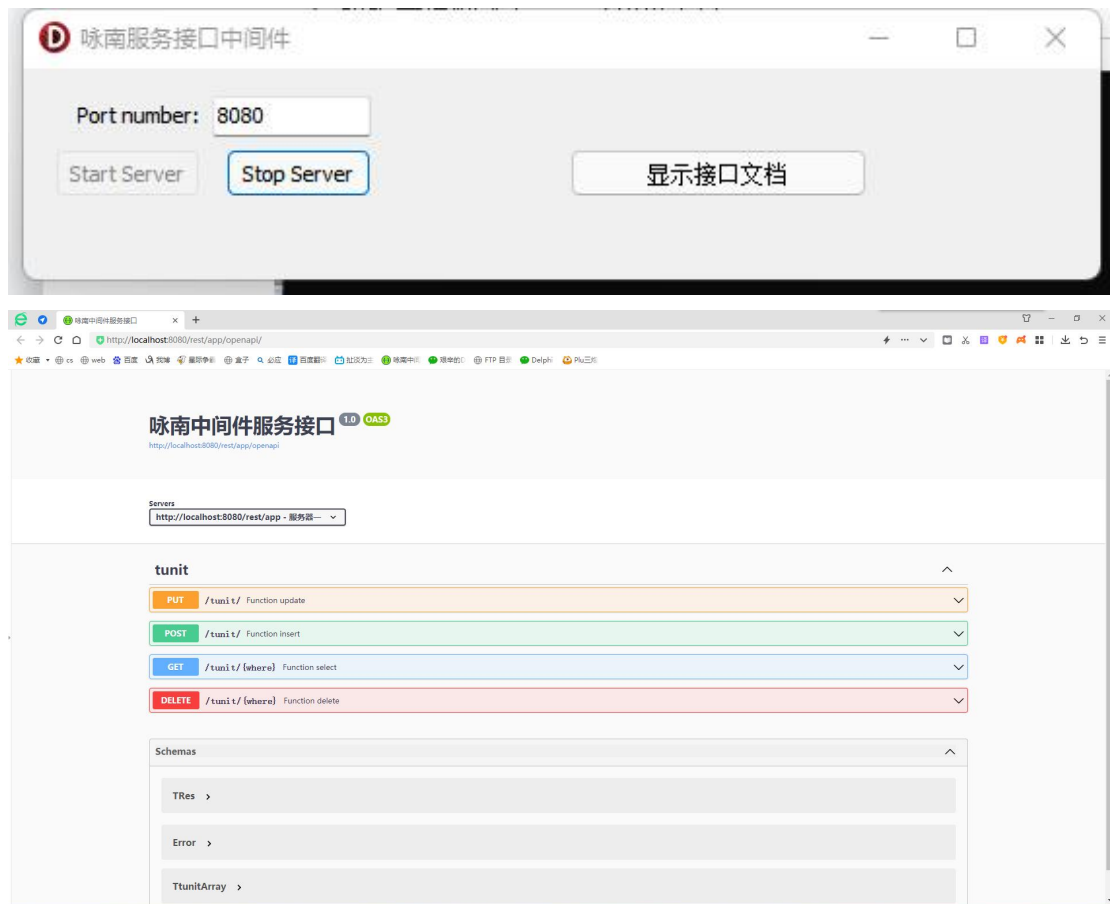
数据表或 SQL 生成 RESTFUL CRUD PASCAL 代码工具



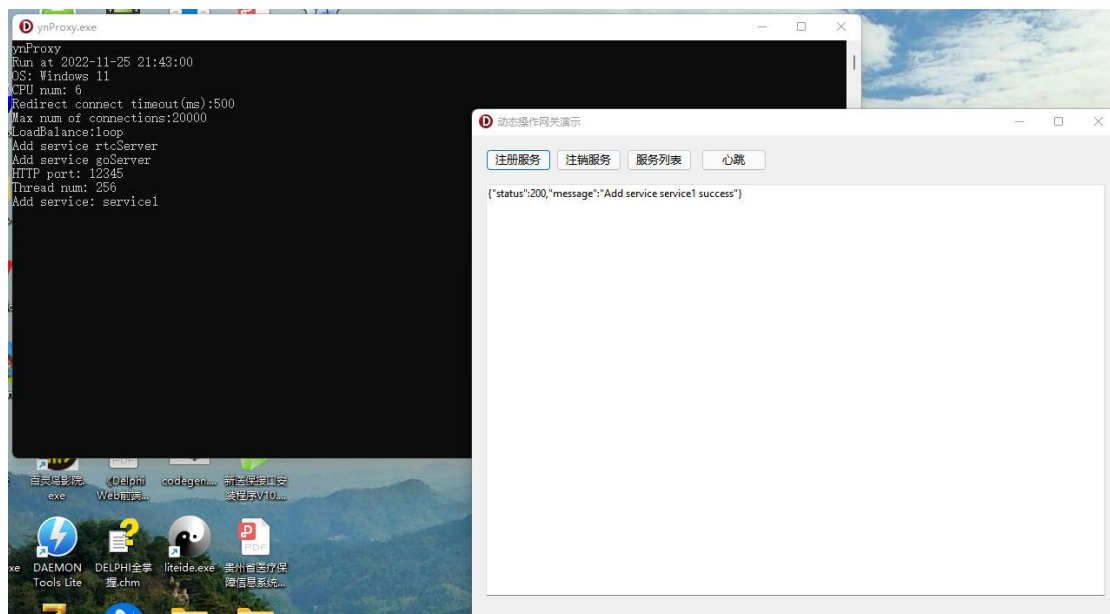
proto 模板文件翻译为 pascal 代码工具



自动生成 SWAGGER 在线接口文档工具



服务自动发现演示



微服务例子

