

consul 跨主机 micro通信

2018-11-26 8:57

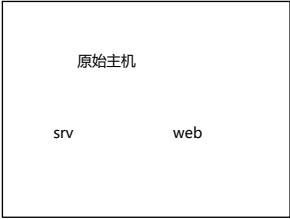
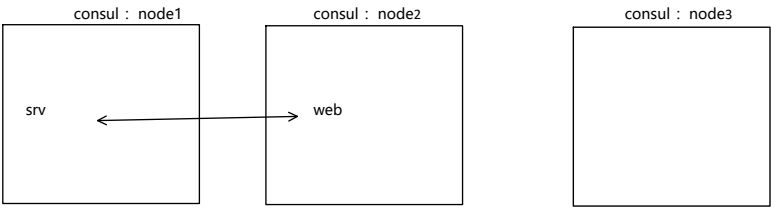
将 srv 和web服务编译好（以防在新的环境下 环境不兼容 但是二进制的可执行程序就没有问题）

将srv 放到我们的consul 的node1 节点

将 web 放到我们的consul的node2节点

我们分别启动node1里面的srv 与node2里面的web

此时我们发现可以进行通信



创建web服务

2018-11-26 14:56

创建命令

```
micro new --type "web" sss/lhomeWeb
```

Creating service go.micro.web.lhomeWeb in /home/itcast/go/src/sss/lhomeWeb

```
.
├─ main.go
├─ plugin.go
├─ handler
│   └─ handler.go
├─ html
│   └─ index.html
├─ Dockerfile
├─ Makefile
└─ README.md
```

在 /home/itcast/go/src/sss/lhomeWeb目录下创建 一个 名为utils文件夹

```
cd /home/itcast/go/src/sss/lhomeWeb
```

mkdir utils 将三个文件考入

创建数据库文件

```
cd /home/itcast/go/src/sss/lhomeWeb
```

```
mkdir models
```

创建1个models.go将 数据库文件导入

运行项目创建数据库

导入数据

```
mkdir conf
```

vim data.sql

将数据考入data.sql

#登录mysql

\$ mysql -uroot -p

#输入root密码

Mysql> use go1micro

#数据的导入

mysql> source ./conf/data.sql

#数据检查

mysql> select * from area;

mysql> select * from facility;

创建conf文件

在conf文件内容创建app.conf

将配置信息写入app.conf

导入前端页面

将hrml.zip

导入我们IhomeWeb的html文件夹中

关于micro

2018-11-26 10:43

micro 是1个微服务的架构的一个框架

protobuf 无关语言无关平台 序列化传输格式

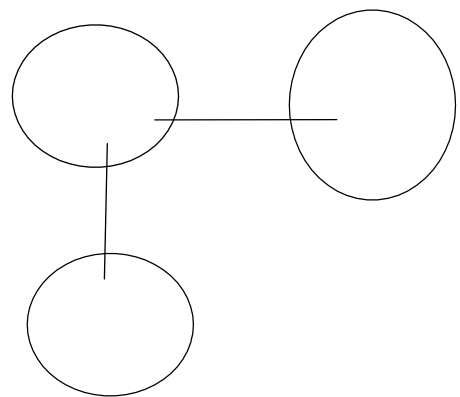
rpc为了了解grpc

grpc是rpc的一个升级版， grpc与protobuf结合使用， 也可使用json 进行数据交互

consul 服务发现和服务管理

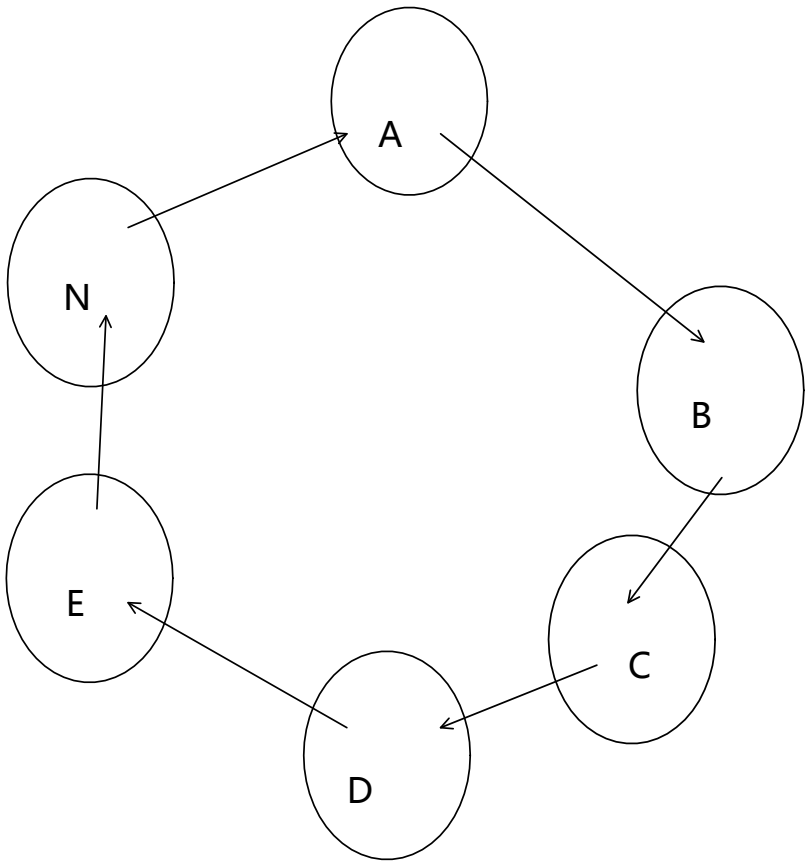
micro 实际上是一个插件化的框架 默认支持的就是 protobuf grpc consul

默认就帮我们进行服务注册的操作将我们的数据发送到consul 进行注册

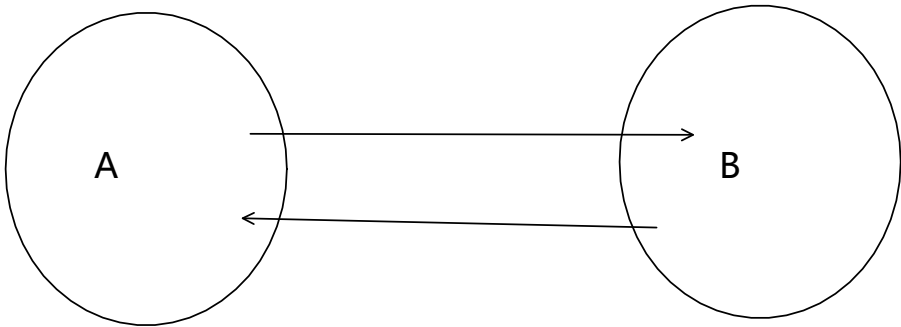


避免环形依赖和双向依赖

环形依赖



双向依赖



开发

2018-11-26 10:52

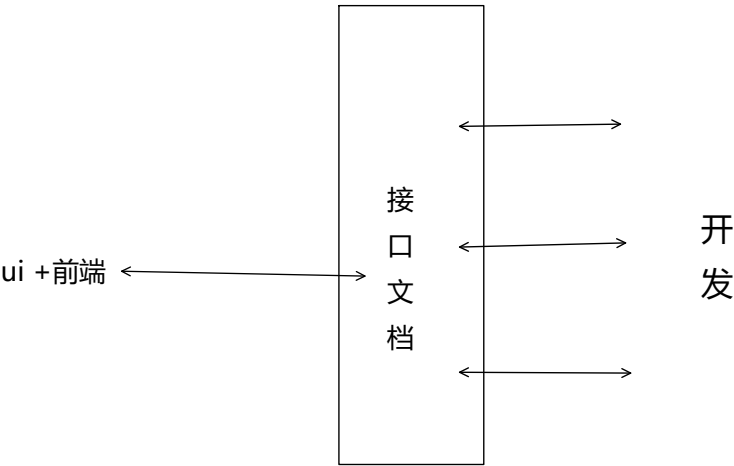
老板 接了一个活

项目经理和对方谈生意 需求
ui +前端：1-2
开发：3-5
测试：可能就是兼职

根据需求大家讨论

如果在开发的时候发现接口有问题向领导提出
集体进行修改

6个月
1-2月 ui+前端
3-6月 才开始开发



项目中 mysql 的使用

2018-11-26 11:29

登录到数据库 `mysql -uroot -p`

输入密码

登录进来

`create database if not exists go1micro default charset utf8 collate utf8_general_ci;`

`go1micro` 表示我们库的名称 我要保存在使用数据库的时候需要作为参数填写

HTTP 进行浏览器 与web服务器的通信

