API有三种类型的可配置请求处理程序：

API Handler: /[service]/[method]

请求/响应：api.Request / api.Response

该路径用于解析服务和方法。

请求通过API服务处理，该服务采用请求api.Request和响应api.Response类型。

请求/响应的定义可以在go-api / proto找到

请求/响应主体的内容类型可以是任何内容。

路由不可用的默认回退处理程序。

设置通过--handler = api

RPC Handler: /[service]/[method]

请求/响应：json / protobuf

默认处理程序的替代方法，它使用转微客户端将请求主体转发为RPC请求。

允许使用具体的Go类型定义API处理程序。

有用的地方你不需要完全控制标题或请求/响应。

可用于运行单层后端服务，而不是附加的API服务。

支持的内容类型application / json和application / protobuf。

通过--handler = rpc设置

Reverse Proxy: /[service]

请求/响应：http

该请求将被反向代理由路径中第一个元素解析的服务

这样可以在API后面实现REST

通过--handler = proxy设置。

或者使用/ rpc端点直接与任何服务通话 - 期望参数：服务，方法，请求，可选地接受地址来定位特定主机