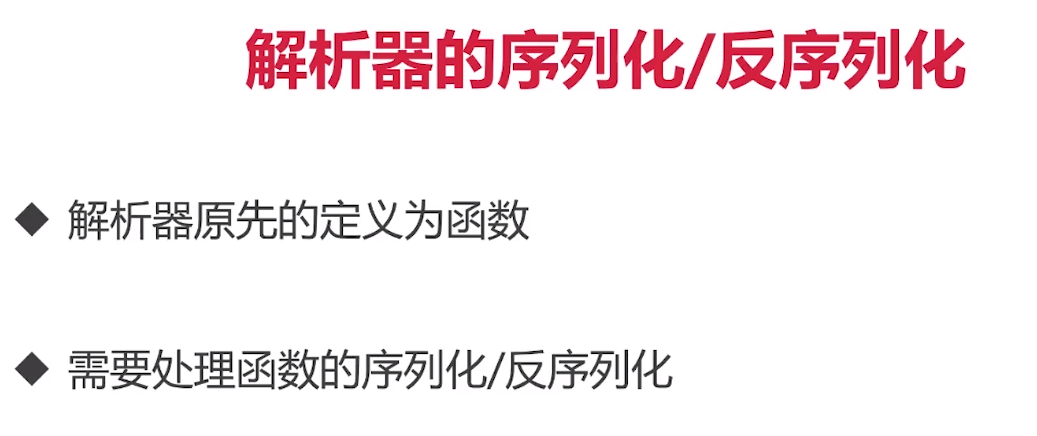
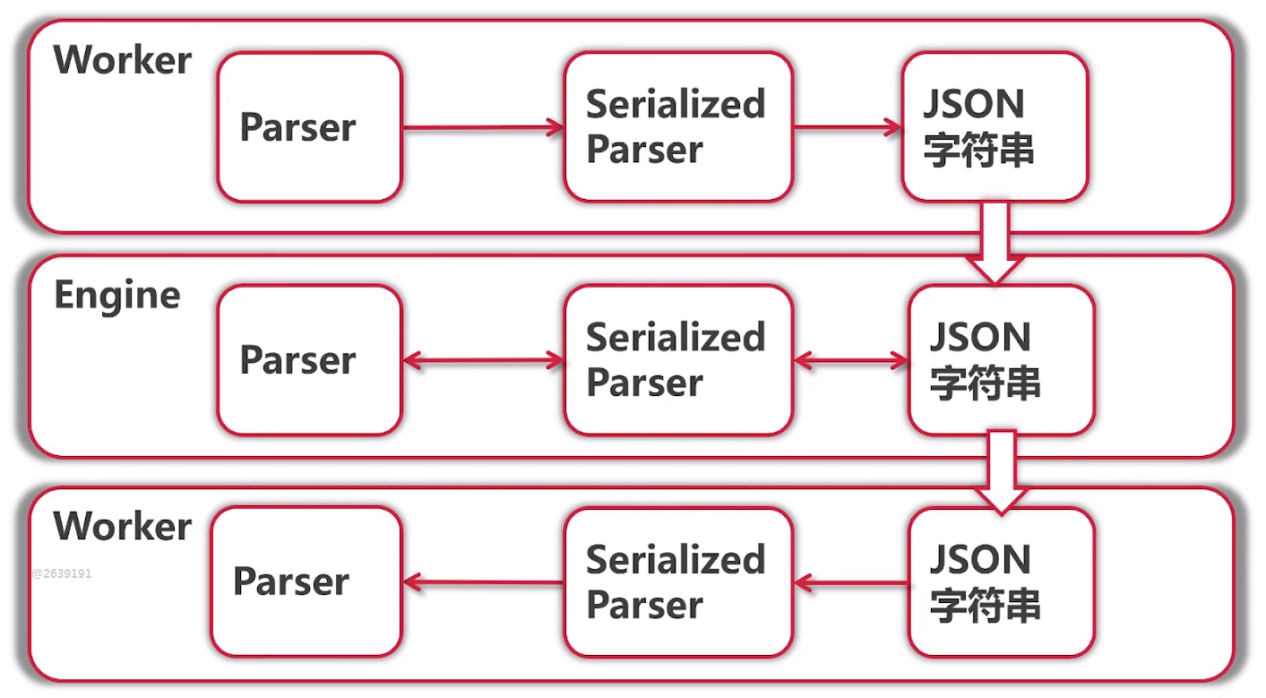
# 解析器的序列化

要想把worker改成RPC，必须把解析器（函数）序列化成json传输，因为解析器是包含在worker的输入、输出里面的。





把函数名字存在一个结构体里面，对结构体进行序列化/反序列化

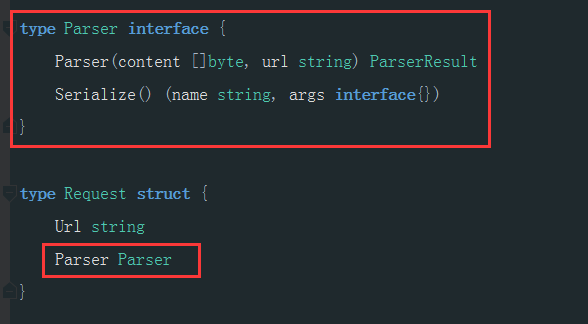


# 把ParserFunc改成接口

把Request的第二个属性ParserFunc方法类型改成了Parser接口，里面的第一个方法是ParserFunc，第二个方法是Serialize。

即原来是传一个方法，现在是利用接口传2个方法。而**第二个方法返回的参数是用来在网上传递给RPC的。**

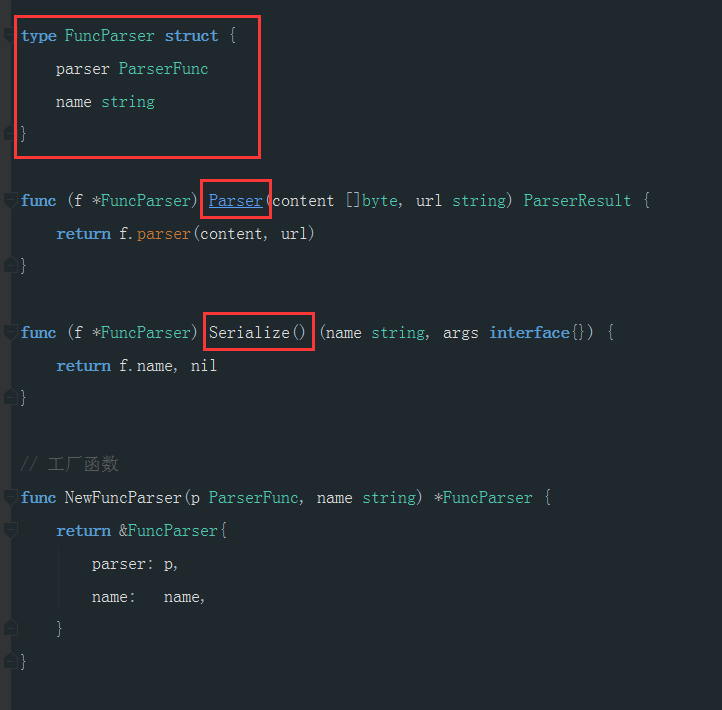
**现在，Parser属性的值是实现Parser接口的结构体对象。**



## 城市列表和城市解析器的改造

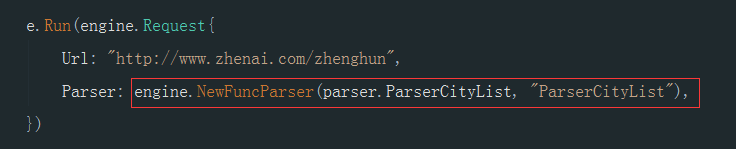
创建FuncParser结构体实现Parser接口

使用工厂函数来创建FuncParser对象，第一个参数是原来的ParserFunc，第二个参数是方法名



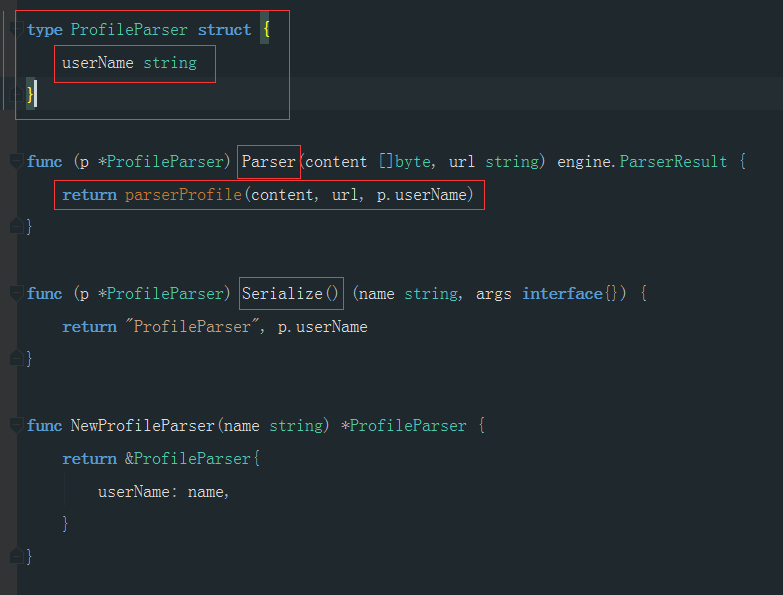
这里用工厂方法创建实现了Parser接口的结构体对象

第一个参数是原来的ParserCityList方法，第二个参数是该方法的名字



## 用户解析器改造

parserProfile多了一个name参数，用结构体属性和Parser方法包装一层，userName会在创建结构体的时候传进来。



创建实现了Parser接口的结构体，并传递userName给结构体

