**Spring boot+grpc demo**

**Grpc服务端**

Maven支持：

<**parent**>  
 <**groupId**>net.devh</**groupId**>  
 <**artifactId**>grpc-server-spring-boot-starter</**artifactId**>  
 <**version**>1.2.0.RELEASE</**version**>  
 <**relativePath**>./</**relativePath**>  
</**parent**>  
  
<**artifactId**>grpc-server-spring-boot-starter</**artifactId**>  
<**version**>1.4.2.RELEASE</**version**>  
  
<**dependencies**>  
 <**dependency**>  
 <**groupId**>net.devh</**groupId**>  
 <**artifactId**>grpc-server-spring-boot-autoconfigure</**artifactId**>  
 <**version**>1.2.0.RELEASE</**version**>  
 </**dependency**>  
</**dependencies**>

以单独的项目编译后发布，供grpc-server使用。

在grpc-server中添加maven依赖：

<**dependency**>  
 <**groupId**>com.grpctest</**groupId**>  
 <**artifactId**>grpcapi</**artifactId**>  
 <**version**>1.1-SNAPSHOT</**version**>  
 <**scope**>compile</**scope**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>net.devh</**groupId**>  
 <**artifactId**>grpc-server-spring-boot-starter</**artifactId**>  
 <**version**>1.4.2.RELEASE</**version**>  
</**dependency**>

实现GRPC生成的接口，并使用@GrpcService注解：

@GrpcService(RPCDateServiceGrpc.**class**)  
**public class** RPCDateService **extends** RPCDateServiceGrpc.RPCDateServiceImplBase{  
  
 @Override  
 **public void** getDate(RPCDateRequest request, StreamObserver<RPCDateResponse> responseObserver) {  
 }

设置属性文件：

**spring.application.name**: **grpc-server  
grpc.server.port**: **9898**

**grpc客户端：**

maven支持：

<**parent**>  
 <**groupId**>net.devh</**groupId**>  
 <**artifactId**>grpc-client-spring-boot-starter</**artifactId**>  
 <**version**>1.2.0.RELEASE</**version**>  
 <**relativePath**>./</**relativePath**>  
</**parent**>  
  
<**artifactId**>grpc-client-spring-boot-starter</**artifactId**>  
<**version**>1.4.1.RELEASE</**version**>  
  
<**dependencies**>  
 <**dependency**>  
 <**groupId**>net.devh</**groupId**>  
 <**artifactId**>grpc-client-spring-boot-autoconfigure</**artifactId**>  
 <**version**>1.2.0.RELEASE</**version**>  
 </**dependency**>  
</**dependencies**>

以单独的项目编译后发布，供grpc-client使用。

在grpc-client中添加maven依赖：

<**dependency**>  
 <**groupId**>com.grpctest</**groupId**>  
 <**artifactId**>grpcapi</**artifactId**>  
 <**version**>1.1-SNAPSHOT</**version**>  
 <**scope**>compile</**scope**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>net.devh</**groupId**>  
 <**artifactId**>grpc-client-spring-boot-starter</**artifactId**>  
 <**version**>1.4.1.RELEASE</**version**>  
</**dependency**>

编写grpc客户端实现接口，使用注解@GrpcClient指定具体关联服务：

@Service  
**public class** GrpcClientService {  
  
 @GrpcClient(**"grpc-server"**)  
 **private** Channel **serverChannel**;  
  
 **public** String sendServerData(String name) {  
 RPCDateServiceGrpc.RPCDateServiceBlockingStub rpcDateService = RPCDateServiceGrpc.*newBlockingStub*(**serverChannel**);  
 *//构造请求对象* RPCDateRequest rpcDateRequest = RPCDateRequest  
 .*newBuilder*()  
 .setUserName(name)  
 .build();  
 RPCDateResponse rpcDateResponse = rpcDateService.getDate(rpcDateRequest);  
 **return** rpcDateResponse.getServerDate();  
 }  
}

在属性文件中作属性指定：

**server.port**: **8081  
spring.application.name**: **grpc-client  
  
grpc.client.grpc-server.host**: **127.0.0.1  
grpc.client.grpc-server.port**: **9898  
grpc.client.grpc-server.enableKeepAlive**: **true  
grpc.client.grpc-server.keepAliveWithoutCalls**: **true**

封装成http rest接口测试

@RestController  
**public class** GrpcClientController {  
 @Autowired  
 **private** GrpcClientService **grpcClientService**;  
  
 @RequestMapping(**"/"**)  
 **public** String printServerData(@RequestParam(defaultValue = **"ChaiShuai"**) String name) {  
 **return grpcClientService**.sendServerData(name);  
 }  
}

运行起来，在localhost：8081看到效果。

参考资料：

<https://blog.csdn.net/qq_28423433/article/details/79108976>

<https://www.v2ex.com/t/343538>

<https://github.com/yidongnan/grpc-spring-boot-starter>

文件grpcapi/grpc-client-spring-boot-starter/grpc-spring-boot-start都是为了作为maven库发布后供grpc-client和grpc-server服务而存在的

FAQ:

1. 添加maven依赖时遇到麻烦，在grpc-server直接加依赖grpc-server-spring-boot-autoconfigure和grpc-server-spring-boot-starter，并不管用，在grpc-server以外，以单独的项目编译发布（在pom文件中设置grpc-server-spring-boot-starter和grpc-server-spring-boot-autoconfigure为子父级关系）供maven使用才可以。？？？？
2. 注意版本冲突。Grpc-all与grpc-server-spring-boot-starte下的子jar包会有版本冲突的问题。