

在 IDEA 中使用 Maven 编译 proto 文件

Google Protocol Buffer(简称 Protobuf) 是 Google 公司内部的混合语言数据标准, 目前已经正在使用的有超过 48,162 种报文格式定义和超过 12,183 个 .proto 文件。他们用于 RPC 系统和持续数据存储系统。

Protocol Buffers 是一种序列化数据结构的方法。对于通过管线(pipeline)或存储数据进行通信的程序开发上是很有用的。这个方法包含一个接口描述语言, 描述一些数据结构, 并提供程序工具根据这些描述产生代码, 用于将这些数据结构产生或解析数据流。

proto2 提供一个程序产生器, 支持C++、Java和Python, 第三方类库支持JavaScript; 而到了 proto3, 支持C++、Java (包含JavaNano)、Python、Go、Ruby、Objective-C和C#, 并且从 3.0.0 Beta 2 版开始支持JavaScript, 第三方类库支持Perl、PHP、Scala和Julia。

我们可以在 Maven 中配置相关插件, 然后直接在 IDEA 编译 proto 文件。我们的 iteblog.proto 文件如下:

```
option java_package = "com.iteblog.data.coprocessor.generated";
option java_outer_classname = "IteblogProtos";
option java_generic_services = true;
option java_generate_equals_and_hash = true;
option optimize_for = SPEED;
```

```
message CountRequest {
}
```

```
message CountResponse {
  required int64 count = 1 [default = 0];
}
```

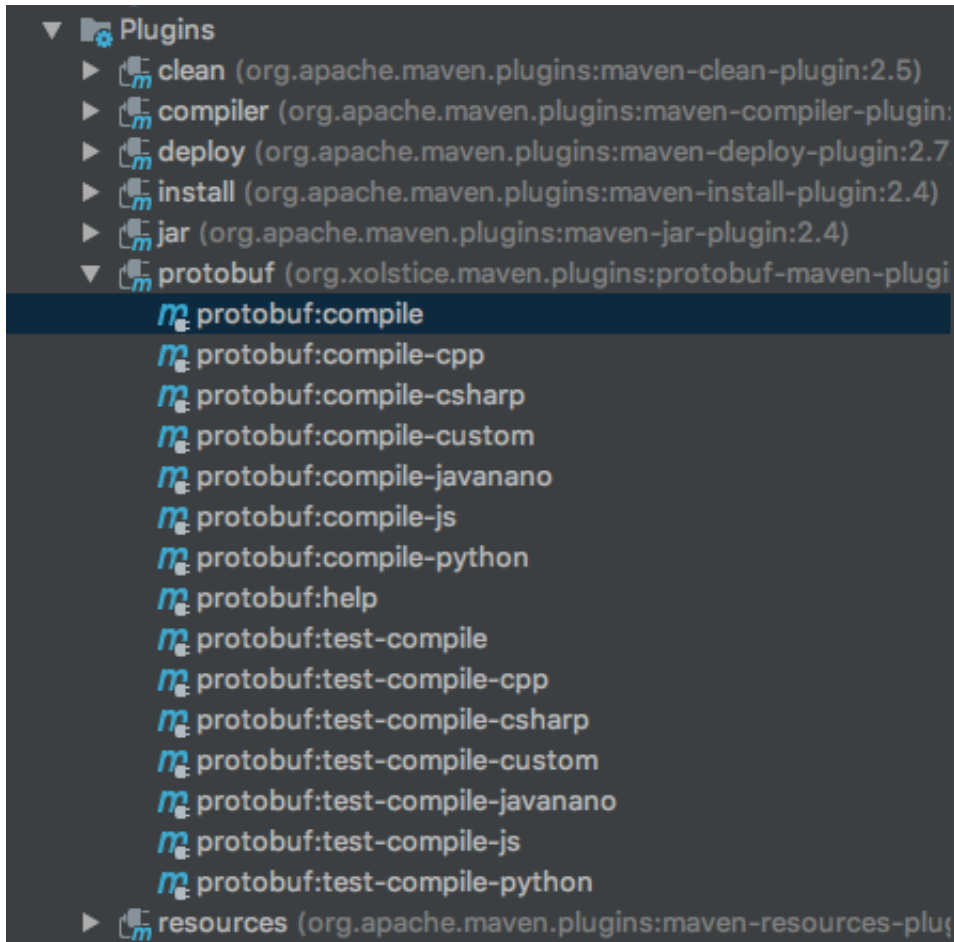
```
service RowCountService {
  rpc getRowCount(CountRequest)
    returns (CountResponse);
  rpc getKeyValueCollection(CountRequest)
    returns (CountResponse);
}
```

我们在 pom.xml 文件里面加入如下插件:

```
<build>
```

```
<plugins>
  <plugin>
    <groupId>org.xolstice.maven.plugins</groupId>
    <artifactId>protobuf-maven-plugin</artifactId>
    <version>0.6.1</version>
    <configuration>
      <protocExecutable>/usr/local/bin/protoc</protocExecutable>
    </configuration>
    <executions>
      <execution>
        <goals>
          <goal>compile</goal>
        </goals>
      </execution>
    </executions>
  </plugin>
</plugins>
</build>
```

注意看上面配置的 `<protocExecutable>/usr/local/bin/protoc</protocExecutable>`，其中 `/usr/local/bin/protoc` 是我们 `protoc` 安装目录，具体如何安装可以参见 [《在CentOS下编译安装Protobuf类库》](#) 文件介绍，MAC 平台安装和里面介绍一致。我们在 maven 里面运行 `protobuf` 插件会调用这个命令，所以需要实现安装好。当 Maven 下载完这个插件之后，你可以在 idea 的 Maven 面板里面看到如下的插件，注意看 `protobuf` 插件，如下：



如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

我们把上面的 iteblog.proto 文件放到 maven 工程的 src/main/proto/ 目录下，如下：

```
./  
+- pom.xml  
+- src/  
  +- main/  
    +- proto/  
      +- iteblog.proto  
  +- test/  
    +- proto/  
      +- test_iteblog.proto
```

然后我们双击上图中的的 protobuf:compile 选项，这时候 maven 开始为我们将 iteblog.proto 文件编译成 java 类：

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.iteblog:hbase >-----
[INFO] Building iteblog 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- protobuf-maven-plugin:0.6.1:compile (default-cli) @ hbase ---
[INFO] Compiling 2 proto file(s) to /Users/iteblog/Desktop/iteblog/target/generated-
sources/protobuf/java
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.591 s
[INFO] Finished at: 2019-02-01T22:32:00+08:00
[INFO] -----
```

Process finished with exit code 0

从上面的输出可以看出，放到当前工程的 target/generated-sources/protobuf/java 目录下面，其中生成的 com.iteblog.data.coprocessor.generated.IteblogProtos.java 就是我们上面定义的。当然，我们还可以通过这个插件生成 cpp、c#、js 以及 python 的文件，这里就不介绍了。

如果你不想用 maven 来编译，你也可以直接使用 protoc 命令来编译：

```
protoc --java_out=/Users/iteblog/ iteblog.proto
```

上面命令会在 /Users/iteblog/ 目录下面生成我们要的 java 类。如果你是 windows，你也可以使用 protoc.exe 来编译，使用和 protoc 一样，这里就不介绍了。

关于 protobuf-maven-plugin 插件的更多用法可以参见其官网 <https://www.xolstice.org/protobuf-maven-plugin/usage.html>，这里就不再介绍了。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: [【】（）](#)