

# protobuf extensions(extend)的消息定义及Java使用方法

时间 2014-07-16 11:22:53 CSDN博客

原文 [http://blog.csdn.net/xiao\\_\\_gui/article/details/37874081](http://blog.csdn.net/xiao__gui/article/details/37874081)

主题 Java Protobuf

---

在定义protobuf消息时，有时候需要用到extensions来对原有的消息类型进行扩展，有利于消息定义的复使用。

1、下面写一个最简单的例子，定义一个message BaseData，并对其进行扩展：

```
// 定义一个message BaseData，100~199之间的tag可供扩展
message BaseData {
    required int32 code = 1;
    extensions 100 to 199;
}

// 扩展BaseData，加上一个extend_data，tag为100
extend BaseData {
    required string extend_data = 100;
}
```

运行protoc.exe --java\_out=. Example.proto生成Example.java文件。

```
public static void main(String[] args) {

    Example.BaseData.Builder baseBuilder = Example.BaseData.newBuilder();
    baseBuilder.setCode(123);
    baseBuilder.setExtension(Example.extendData, "xxg");
    Example.BaseData baseData = baseBuilder.build();

    System.out.println(baseData.getCode());
    System.out.println(baseData.getExtension(Example.extendData));
}
```

这里需要调用setExtension方法来设置扩展的extendData。setExtension有两个参数，第一个定义扩展是哪一個字段，Example.extendData即表示Example类中的extendData，第二个就是设置扩展字段的值

2、另外还有一种常用的扩展定义方式：

```
// 定义一个message BaseData，100~199之间的tag可供扩展
message BaseData {
    required int32 code = 1;
    extensions 100 to 199;
}

// 定义一个message Data
message Data {
    required string msg = 1;
```

```

        required Data extend_data = 100;
    }
}

```

这种方式看起来有点奇怪，实际上可以理解成这样：

```

// 定义一个message BaseData, 100~199之间的tag可供扩展
message BaseData {
    required int32 code = 1;
    extensions 100 to 199;
}

// 定义一个message Data
message Data {
    required string msg = 1;
}

// 扩展BaseData, 加上一个Data类型的字段, tag为100
extend BaseData {
    required Data extend_data = 100;
}

```

Java代码：

```

public static void main(String[] args) {

    // 先构造一个message Data类型对象
    Example.Data.Builder dataBuilder = Example.Data.newBuilder();
    dataBuilder.setMsg("xxg");
    Example.Data data = dataBuilder.build();

    // 再构造message BaseData对象, 将Data对象通过setExtension设置到BaseData中
    Example.BaseData.Builder baseBuilder = Example.BaseData.newBuilder();
    baseBuilder.setCode(123);
    baseBuilder.setExtension(Example.Data.extendData, data);
    Example.BaseData baseData = baseBuilder.build();

    System.out.println(baseData.getCode());
    System.out.println(baseData.getExtension(Example.Data.extendData).getMsg());
}

```

注意上面的Java代码用的是第一种proto消息定义方式，如果用的是下面第二种消息定义的方式，setExtension的参数Example.Data.extendData就应该改成Example.extendData，注意嵌套结构。