protobuf是Google开发的一种数据描述语言语言，能够将结构化的数据序列化，可用于数据存储，通信协议等方面。

值得注意的是，protobuf是以二进制来存储数据的。相对于JSON和XML具有以下优点：

1. 简洁
2. 体积小:消息大小只需要XML的1/10 ~ 1/3
3. 速度快:解析速度比XML快20 ~ 100倍
4. 使用Protobuf的编译器,可以生成更容易在编程中使用的数据访问代码
5. 更好的兼容性,Protobuf设计的一个原则就是要能够很好的支持向下或向上兼容

# 实例代码：

package example; //包名

option java\_package = "com.example"; // 指定类存放包

option java\_outer\_classname = "Student"; //指定编译得到的类名

message stu {

required string name = 1;

required int32 id = 2;

optional string email = 3;

}

程序下载/安装 请Google：Protocol Buffers可执行文件。

<https://developers.google.com/protocol-buffers>

# proto 编译命令

protoc -I=$SRC\_DIR --java\_out=$DST\_DIR $SRC\_DIR/student.proto

# 消息类/Builder类

通过Builder类来得到消息类的对象，消息类即是protocol文件中定义的消息，编译最之后得到对应的.java类文件。

* Builder对象有get、set方法，消息类有get方法。
* 消息对象有get方法。消息对象一旦被创建之后不能被修改。

Student.Builder builder = Student.newBuilder(); //得到Builder类

builder.setName(“Xiaoming”);

builder.setId(10086);

builder.setEmail(“xiaoming@huwwei.com”);

Stu xiaoming = builder.build(); //生成消息对象

String name = xiaoming.getName(); // 查询数据

int id = xiaoming.getId();

String email = xiaoming.getEmail();

# 标准的消息方法

每个消息和builder类还包含大量的其它方法，来让你检查或管理整个消息，包括：

* isInitialized() : 检查是否所有的required字段都已经被设置了。
* toString() : 返回一个人类可读的消息表示，对调试特别有用。
* mergeFrom(Message other): (只有builder可用) 将 **other** 的内容合并到这个消息中，覆写单数的字段，附接重复的。
* clear(): (只有builder可用) 清空所有的元素为空状态。

# 解析和序列化

最后，每个protocol buffer类都有使用protocol buffer二进制格式写和读你所选择类型的消息的方法。这些方法包括：

* byte[] toByteArray();: 序列化消息并返回一个包含它的原始字节的字节数组。
* static Person parseFrom(byte[] data);: 从给定的字节数组解析一个消息。
* void writeTo(OutputStream output);: 序列化消息并将消息写入**OutputStream**。
* static Person parseFrom(InputStream input);: 从一个**InputStream** 读取并解析消息。