

```

public void afterReopen() throws IOException {

}

@Override
public void write(Event event) throws IOException {
    UsingFlumeEvent.Event.Builder
        builder = UsingFlumeEvent.Event.newBuilder();
    for (Map.Entry<String, String> entry : event.getHeaders()
        .entrySet()) {
        builder.addHeader(UsingFlumeEvent.Header.newBuilder()
            .setKey(entry.getKey())
            .setVal(entry.getValue()).build());
    }
    builder.setBody(ByteString.copyFrom(event.getBody()));
    UsingFlumeEvent.Event e = builder.build();
    stream.write(ByteBuffer.allocate(Integer.SIZE / 8).putInt(e
        .getSerializedSize()).array());
    e.writeTo(stream);
}

@Override
public void flush() throws IOException {
    stream.flush();
}

@Override
public void beforeClose() throws IOException {
    if (writeHeaderAndFooter) {
        stream.write(footer);
    }
}

@Override
public boolean supportsReopen() {
    return false;
}

public static class Builder implements EventSerializer.Builder {

    @Override
    public EventSerializer build(Context context,
        OutputStream outputStream) {
        return new ProtobufSerializer(context, outputStream);
    }
}

```