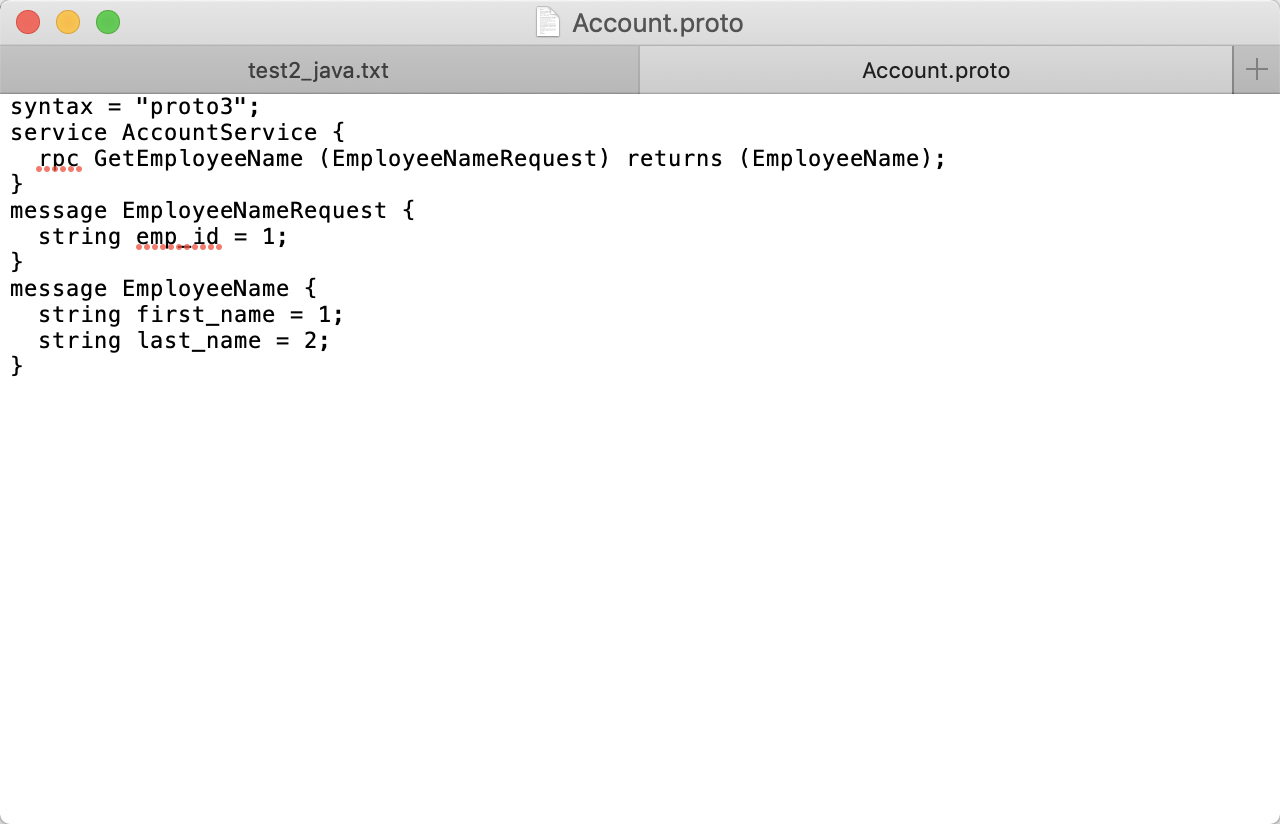
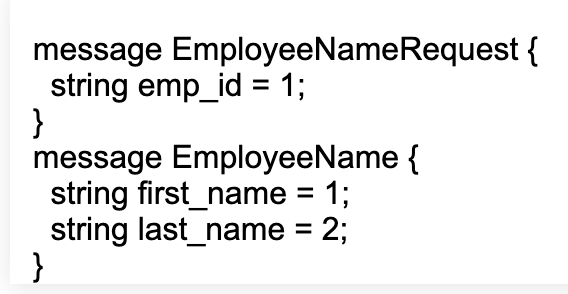
**2020年05月19日测试**

1. 测试说明
2. 测试网址：<http://7ke.top/show>
3. 测试文件组：共100份文件组，1份文件包括
   1. xxx.proto，proto schema文件
   2. xxx.pb，二进制文件
   3. xxx\_pb2.py，proto schema文件经protoc生成的py文件
   4. xxx\_example.py，生成proto实例的py文件
   5. xxx.json，由Google的MessageToJson函数对xxx.proto + xxx.pb生成的JSON文件
4. 特殊测试文件组：
   1. OrderInfo、protodemo、student、demo4、book：中文实例，但OrderInfo.json文件中保存的为\u编码
   2. GroundData：proto文件嵌套定义SensorData和GpsLocation，无法单独解析
   3. hello、Login：二进制文件、json文件内容超长
   4. test6：pb和JSON文件均为空内容
   5. PersonTest：PersonTest.proto文件中有特殊字符注释
5. 成功率总结
6. 纯proto文件转换，转换成功定义为“转换结果与原proto schema文件基本无差别”。当前成功率99%。
7. Java转换，转换成功定义为“上传对应proto和pb后，网页所转换结果与对应的JSON文件基本一致”。当前成功率70%。
8. Python转换，转换成功定义同Java转换。当前成功率53%。
9. 问题：
10. 前台页面在仅转换proto时，所有的转换都是从第一个字符串“message”开始的，“message”之前的所有信息均丢失。

以Account.proto为例，其内容为：



转换后



这就导致**所有的**proto解析中均没有syntax信息、package信息（若有）、定义在message外的option（若有）、在message外定义的enum（若有）。

**造成信息丢失。例如test6.proto、test4.proto、demo6.proto。特殊的，addressbook.proto文件可认为转换失败。**

|  |  |
| --- | --- |
| addressbook.proto文件内容 | addressbook.proto经过网页解析的内容 |
|  |  |

1. 不支持多个文件上传，直接问题是无法完成特殊的类型嵌套proto + pb的转换，嵌套有GroundData.proto（嵌套SensorData.proto和GroundData.proto）

Java转换失败情况：

情况①：未知错误10003、{result:"loading..."}

情况②：转换错误，内容与对应的.json文件显然不同

情况③：{errormessage:"未知错误，错误码501"}

情况④：{errormessage:"未知错误，错误码502"}

|  |  |  |  |
| --- | --- | --- | --- |
| Java转换失败文件（xxx.proto + xxx.pb） | | | |
| 情况① | 情况② | 情况③ | 情况④ |
| http | complex | test4 | product |
| testP | anthor | demo6 | transaction |
| demo | demo7 | yages-schema | MessageData |
| employee | student | calendar | CompanyProto |
| cover\_helloworld | msg | addressbook | spirits |
| person4 | msginfo | my-message | innerMap |
|  |  | Messages | Message |
|  |  | message3 |  |
|  |  | GroundData |  |
|  |  | user2 |  |
|  |  | Common |  |

以complex举例说明，在上传complex.proto和complex.pb后，对比complex.json

|  |  |
| --- | --- |
| complex.json内容 | complex.proto + comple.pb由网页解析得到的json内容 |
|  |  |

Python转换失败情况：

情况⑤：不能使用，上传proto和对应pb后，无法单击“后台Python解析”按钮

情况⑥：{errormessage:"属性中出现错误修饰词:proto2支持<required>,<optional>和<repeated>,必须显式声明;proto3支持<singular>和<repeated>,默认为<singular>"}

情况⑦：转换错误，内容与对应的.json文件显然不同

情况⑧：{errormessage:"未知错误，错误码523"}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Python转换失败对应文件（xxx.proto + xxx.pb） | | | | | |
| 情况⑤ | | | 情况⑥ | 情况⑦ | 情况⑧ |
| data2 | cover\_helloworld | PersonTest | storage\_messages | player | person2 |
| tallMessage | KVCommand | msginfo | oneoftest | yages-schema |  |
| test6 | game\_service | Login | demo | Messages |  |
| test4 | my-message | ProtocolModule | anthor | GroundData |  |
| TimeSchema | data | customer | demo6 |  |  |
| people | addressbook3 | example | demo3 |  |  |
| demo4 | message3 | hello | person3 |  |  |
| calendar | attendee | geolocation | Message |  |  |
| demo7 | example2 | common2 |  |  |  |
| student | test | message2 |  |  |  |
| protocolbufTest | lona | addressbook2 |  |  |  |
| protodemo |  |  |  |  |  |

以player举例说明，在上传player.proto和player.pb后，对比player.json

|  |  |
| --- | --- |
| player.json内容 | player.proto + player.pb由网页解析得到的json内容 |
|  |  |

1. 其他

对于上传不相关的proto schema和二进制文件依然能解析成功的问题，做如下讨论：

1. Java解析

共测试121种文件组合，出现这类情况的文件包括但不限于：data2.proto + http.pb / test4.pb / HotMessage.pb / anthor.pb / test2.pb / demo3.pb / UDP.pb / cover\_helloworld.pb / dog.pb / Message.pb / data.pb / example2.pb / msginfo.pb / GroundData.pb / person.pb / message2.pb / protodemo.pb / user.pb

例如对于data2.proto + http.pb：

|  |  |
| --- | --- |
| data2.proto内容 | data2.proto网页转换内容 |
|  |  |
| http.json内容 | 网页中data2.proto+http.pb的解析结果 |
|  |  |

但是使用Google的MessageToJson函数是同样可以解析成功的，解析结果为：



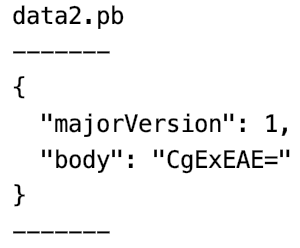
1. Python解析：

共测试495种文件组合，出现这类情况的文件包括但不限于：http.proto + data2.pb / test6.pb / test4.pb / simple.pb / test2.pb / GpsLocation.pb / dog.pb / SensorData.pb / common2.pb

例如对于http.proto + data2.pb：

|  |  |
| --- | --- |
| http.proto内容 | http.proto网页转换内容 |
|  |  |
| data2.json内容 | 网页中http.proto + data2.pb的解析结果 |
|  |  |

但是使用google的MessageToJson函数是同样可以解析成功的，解析结果为：



由此，我认为可以得出结论：对于上传不相关的proto schema和二进制文件依然能解析成功的问题，这应该不能算是个BUG，可理解为强制类型转换，造成信息丢失，且生成无意义的json内容。