

# DSL-程序设计实践 测试文档

## 目录

DSL-程序设计实践 测试文档.....	1
0 测试说明.....	1
1 自动测试脚本 .....	2
1.1 自动测试脚本源码.....	2
1.2 自动测试脚本说明.....	3
1.2 测试脚本文件 chatScript.bot.....	4
1.2.1 脚本文件内容.....	4
1.2.2 测试用例 1.....	6
1.2.2 测试用例 2.....	7
1.2.3 测试用例 3.....	9
2 测试桩.....	10
2.1 测试用户 1.....	11
2.2 测试用户 2.....	12

## 0 测试说明

测试人员：陈朴炎

学号：2021211138

班级：2021211307

测试时间：2023-11-27

# 1 自动测试脚本

## 1.1 自动测试脚本源码

```
1. from scriptParser import ScriptParser
2.
3. # 自动测试脚本
4. class AutoTest:
5.     def __init__(self, input_path, output_path):
6.         print("init...")
7.         self.test_robot = ScriptParser("E:\\bupt-homework/程序设计实践
           /DSL/script_file/chatScript.bot")
8.         # 模拟运行环境
9.         self.user_states = self.test_robot.get_entry()
10.        self.input_path = input_path
11.        self.output_path = output_path
12.        # 输入文本
13.        self.input = []
14.        # 检验文本
15.        self.exam = []
16.        self.output = []
17.
18.        # 导入输入文件和验证文件
19.        def load_file(self):
20.            with open(self.input_path, 'r', encoding='utf-8') as file:
21.                input_str = file.read()
22.            with open(self.output_path, 'r', encoding='utf-8') as file:
23.                output_str = file.read()
24.            for line in input_str.split('\n'):
25.                self.input.append(line.strip())
26.            for line in output_str.split('\n'):
27.                self.exam.append(line.strip())
28.
29.        # 获取生成的输出
30.        def get_output(self):
31.            for line in self.input:
32.                responses, state = self.test_robot.get_response(line, self.user_s
                    tates)
33.                for response in responses:
34.                    self.output.append(response)
35.                self.user_states = state
36.            for response in self.output:
37.                print(response)
```

```

38.
39.     # 检查输出是否正确
40.     def exam_output(self):
41.         # 1.检查生成语句的数量
42.         if len(self.output) != len(self.exam):
43.             return False
44.         # 2.逐一比较输出内容
45.         for i in range(len(self.output)):
46.             print("output: {} \texam:{}".format(self.output[i], self.exam[i])
47.             )
48.             if self.output[i] != self.exam[i]:
49.                 return False
50.         return True
51.     # 运行自动测试
52.     def run(self):
53.         self.load_file()
54.         self.get_output()
55.         if self.exam_output() is not True:
56.             print("测试不通过")
57.         else:
58.             print("测试通过")
59.
60.
61. if __name__ == "__main__":
62.     input_path = "test_module/testInput1.txt"
63.     output_path = "test_module/testOutput1.txt"
64.     test_bot = AutoTest(input_path, output_path)
65.     test_bot.run()

```

## 1.2 自动测试脚本说明

自动测试脚本的运行主要看它的 run 方法，当主程序调用这个自动测试脚本的 run 方法时，会先导入预先定义好的输入和相应的回应检验。然后调用解释器，通过解释器来获取回应信息，一一比对，如果有误，则报错“测试不通过”，如果都正确，则报告“测试通过”。

自动测试脚本的创建需要引入两个文件的路径，input\_path 是预先定义好的输入文件路径，每个输入用换行符隔开。output\_path 是正确输出的文件路径，

里面同样将每个回应用换行符隔开。

`load_file()`函数将两个文件打开, 将 `input_file` 里的每一行都添加到 `input` 列表中。同理, 将 `output_file` 里的每一行都添加到 `exam` 列表中。

`get_output()` 函数通过调用解释器的函数, 获取每个 `input` 的回应, 并将回应添加到 `output` 列表中。

`exam_output()` 函数先比对 `exam` 和 `output` 列表里的元素数量是否相同, 若不相同, 直接抱错; 若相同, 则一一比对里面的字符串, 如果有一对不一样, 则返回错误, 若全一样, 则返回正确。

## 1.2 测试脚本文件 chatScript.bot

### 1.2.1 脚本文件内容

```
1. entry "初始"
2.
3. when "你好" state_transfer "初始"
4. when "售前|推荐" state_transfer "售前"
5. when "售后|问题" state_transfer "售后" response "现在给您转售后..."
6. when "谢谢|再见|好的|好吧" response "很高兴能帮助到您, 祝您生活愉快~"
7. when "订单" response "$username 的订单是: $order"
8.
9. state "初始"
10.     response "您好, 请问您有什么需要帮助的吗?"
11.     when "你好|您好" response "您好, 请问您有什么需要帮助的吗?"
12.     when "机器人" response "您好, 我是您的客服机器人, 有什么需要帮助的吗?"
13.     when "人工" response "请求人工客服, 请拨打 2021211138"
14.     when "售后|问题" state_transfer "售后" response "现在给您转售后..."
15.     when "售前|推荐" state_transfer "售前"
16.
17. state "售前"
18.     response "您好, 我们有手机、电脑、平板、手表, 您需要哪一种呢?"
19.     when "手机" state_transfer "售前手机"
20.     when "电脑" state_transfer "售前电脑"
21.     when "平板" state_transfer "售前平板"
22.     when "手表" state_transfer "售前手表"
```

```

23.
24. state "售后"
25.     response "您好，请问您的什么产品遇到问题了呢"
26.     when "手机" state_transfer "售后手机"
27.     when "电脑" state_transfer "售后电脑"
28.     when "平板" state_transfer "售后平板"
29.     when "手表" state_transfer "售后手表"
30.
31. state "售前手机"
32.     response "手机名称：北邮一号；价格：¥8888"
33.     when "电池" response "电池容量是 4600mah，续航 12 小时"
34.     when "屏幕" response "屏幕为 LCD 屏，分辨率 2k 2560 x 1440"
35.     when "尺寸|大小" response "尺寸为 5.6 寸"
36.     when "保修" response "保修 4 年"
37.
38. state "售前电脑"
39.     response "电脑名称：北邮二号；价格：¥88888"
40.     when "电池" response "电池容量是 75Wh，续航 12 小时"
41.     when "屏幕" response "屏幕为 LCD 屏，分辨率 2k 2560 x 1440"
42.     when "尺寸|大小" response "尺寸为 16 寸"
43.     when "保修" response "保修 4 年"
44.
45. state "售前平板"
46.     response "平板名称：北邮三号；价格：¥3888"
47.     when "电池" response "电池容量是 7600mah，续航 24 小时"
48.     when "屏幕" response "屏幕为 LCD 屏，分辨率 2k 2560 x 1440"
49.     when "尺寸|大小" response "尺寸为 11 寸"
50.     when "保修" response "保修 4 年"
51.
52. state "售前手表"
53.     response "手表名称：北邮四号；价格：¥888"
54.     when "电池" response "续航一个月"
55.     when "尺寸|大小" response "手表大小 42mm"
56.     when "保修" response "保修 4 年"
57.
58. state "售后手机"
59.     response "您好，您的手机遇到什么问题了呢？"
60.     when "电池" response "电池问题请自行更换电池"
61.     when "屏幕" response "屏幕问题请自行更换屏幕"
62.
63. state "售后电脑"
64.     response "您好，您的电脑遇到什么问题了呢？"
65.     when "电池" response "电池问题请自行更换电池"
66.     when "屏幕" response "屏幕问题请自行更换屏幕"

```

```

67.
68. state "售后平板"
69.     response "您好，您的平板遇到什么问题了呢？"
70.     when "电池" response "电池问题请自行更换电池"
71.     when "屏幕" response "屏幕问题请自行更换屏幕"
72.
73. state "售后手表"
74.     response "您好，您的手表遇到什么问题了呢？"
75.     when "电池" response "电池问题请自行更换电池"
76.     when "屏幕" response "屏幕问题请自行更换屏幕"

```

### 1.2.2 测试用例 1

input 文件如下：

input

你好

请问有什么推荐的吗？

我想看手机

屏幕多大？

好的

output 文件如下：

抱歉，我不明白您的意思

您好，请问您有什么需要帮助的吗？

您好，我们有手机、电脑、平板、手表，您需要哪一种呢？

手机名称：北邮一号；价格：¥ 8888

屏幕为 LCD 屏，分辨率 2k 2560 x 1440

很高兴能帮助到您，祝您生活愉快~

抱歉，我不明白您的意思

执行结果：

```
output: 抱歉,我不明白您的意思    exam:抱歉,我不明白您的意思
output: 您好,请问您有什么需要帮助的吗?    exam:您好,请问您有什么需要帮助的吗?
output: 您好,我们有手机、电脑、平板、手表,您需要哪一种呢?    exam:您好,我们有手机、电脑、平板、手表,您需要哪一种呢?
output: 手机名称:北邮一号;价格:¥8888    exam:手机名称:北邮一号;价格:¥8888
output: 屏幕为LCD屏,分辨率2k 2560 x 1440    exam:屏幕为LCD屏,分辨率2k 2560 x 1440
output: 很高兴能帮助您,祝您生活愉快~    exam:很高兴能帮助您,祝您生活愉快~
output: 抱歉,我不明白您的意思    exam:抱歉,我不明白您的意思
测试通过
```

图 1-1 测试用例 1 执行结果

可以看到,自动测试脚本测试通过了。

## 1.2.2 测试用例 2

input 文件如下：

啊啊啊啊啊啊啊啊啊啊啊啊

我是谁？

你是机器人吗？

我有问题

什么？

好吧

再见

不行？

你不懂？

你可以给我推荐产品吗？

我要看手表

我的手表有问题

电池有问题

好吧

output 文件如下：

抱歉，我不明白您的意思

抱歉，我不明白您的意思

您好，我是您的客服机器人，有什么需要帮助的吗

现在给您转售后...

您好，请问您的什么产品遇到问题了呢

抱歉，我不明白您的意思

很高兴能帮助到您，祝您生活愉快~

很高兴能帮助到您，祝您生活愉快~

抱歉，我不明白您的意思

抱歉，我不明白您的意思

您好，我们有手机、电脑、平板、手表，您需要哪一种呢？

手表名称：北邮四号；价格：¥ 888

现在给您转售后...

您好，请问您的什么产品遇到问题了呢

现在给您转售后...

很高兴能帮助到您，祝您生活愉快~

执行结果如下，测试通过：



```

output: 抱歉, 我不明白您的意思      exam: 抱歉, 我不明白您的意思
output: 抱歉, 我不明白您的意思      exam: 抱歉, 我不明白您的意思
output: 您好, 我是您的客服机器人, 有什么需要帮助的吗      exam: 您好, 我是您的客服机器人, 有什么需要帮助的吗
output: 现在给您转售后...      exam: 现在给您转售后...
output: 您好, 请问您的什么产品遇到问题了呢      exam: 您好, 请问您的什么产品遇到问题了呢
output: 抱歉, 我不明白您的意思      exam: 抱歉, 我不明白您的意思
output: 很高兴能帮助到您, 祝您生活愉快~      exam: 很高兴能帮助到您, 祝您生活愉快~
output: 很高兴能帮助到您, 祝您生活愉快~      exam: 很高兴能帮助到您, 祝您生活愉快~
output: 抱歉, 我不明白您的意思      exam: 抱歉, 我不明白您的意思
output: 抱歉, 我不明白您的意思      exam: 抱歉, 我不明白您的意思
output: 您好, 我们有手机、电脑、平板、手表, 您需要哪一种呢?      exam: 您好, 我们有手机、电脑、平板、手表, 您需要哪一种呢?
output: 手表名称: 北邮四号; 价格: ¥888      exam: 手表名称: 北邮四号; 价格: ¥888
output: 现在给您转售后...      exam: 现在给您转售后...
output: 您好, 请问您的什么产品遇到问题了呢      exam: 您好, 请问您的什么产品遇到问题了呢
output: 现在给您转售后...      exam: 现在给您转售后...
output: 很高兴能帮助到您, 祝您生活愉快~      exam: 很高兴能帮助到您, 祝您生活愉快~
测试通过

```

图 1-2 测试用例 2 执行结果

### 1.2.3 测试用例 3

input 文件如下:

你好

请给我转人工

好吧

我的手机有问题

我的手机坏了

屏幕坏了

output 文件如下:

您好, 请问您有什么需要帮助的吗?

请求人工客服, 请拨打 2021211138

很高兴能帮助到您, 祝您生活愉快~

现在给您转售后...

您好, 请问您的什么产品遇到问题了呢

您好，您的手机遇到什么问题了呢？

屏幕问题请自行更换屏幕

执行结果如下，测试通过：

```
output: 您好，请问您有什么需要帮助的吗？      exam: 您好，请问您有什么需要帮助的吗？
output: 请求人工客服，请拨打20212111138      exam: 请求人工客服，请拨打2021211138
output: 很高兴能帮助到您，祝您生活愉快~      exam: 很高兴能帮助到您，祝您生活愉快~
output: 现在给您转售后...      exam: 现在给您转售后...
output: 您好，请问您的什么产品遇到问题了呢？      exam: 您好，请问您的什么产品遇到问题了呢？
output: 您好，您的手机遇到什么问题了呢？      exam: 您好，您的手机遇到什么问题了呢？
output: 屏幕问题请自行更换屏幕      exam: 屏幕问题请自行更换屏幕
测试通过
```

图 1-3 测试用例 3 执行结果

## 2 测试桩

```
1. # 环境变量
2. test_order1 = Order(1111111111, "手机", 1688, "2023-11-24")
3. test_order2 = Order(2222222222, "电脑", 6888, "2023-11-05")
4. test_order3 = Order(3333333333, "手表", 688, "2023-11-15")
5. test_order4 = Order(4444444444, "平板电脑", 2688, "2023-11-27")
6. test_order_list1 = [test_order1, test_order2]
7. test_user1 = Vars("1", "", test_order_list1)
8. test_order_list2 = [test_order3, test_order4]
9. test_user2 = Vars("2", "", test_order_list2)
10.
11. # 导入用户数据的接口
12. def load_user_data():
13.     user_data = {"1":test_user1,"2":test_user2}
14.     return user_data
15.
16. # 测试桩
17. user_data = load_user_data()
18. chat_robot = ScriptParser("../script_file/chatScript.bot")
19. sockets_user = {}
```

在服务器文件下，预留了导入用户数据的接口，通过 `load_user_data()` 函数来导入用户的数据。函数具体如何实现，并不完全属于本模块内容，而是交由其他模块实现的开发人员实现。这里仅提供一个测试桩，用于测试在上述这些环境变量下服务器能否正常工作。

## 2.1 测试用户 1

登录用户 1 的账号，询问客服机器人用户 1 的订单，客服返回消息如下：

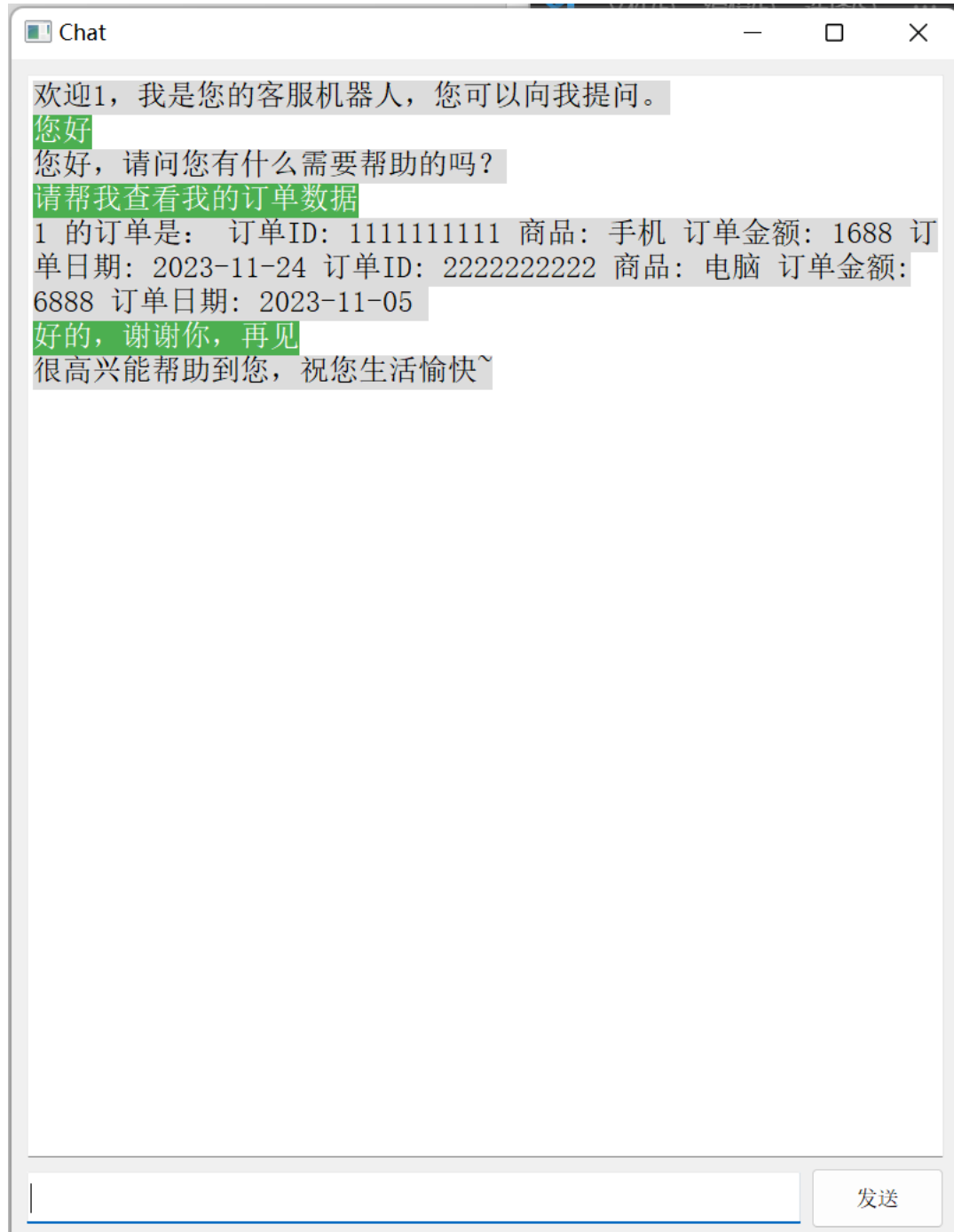


图 2-1 用户 1 的订单查询结果图

可以看到，客服机器人成功输出了用户 1 的账号和订单信息。

## 2.2 测试用户 2

登录用户 2 的账号，询问客服机器人用户 2 的订单，客服返回消息如下：

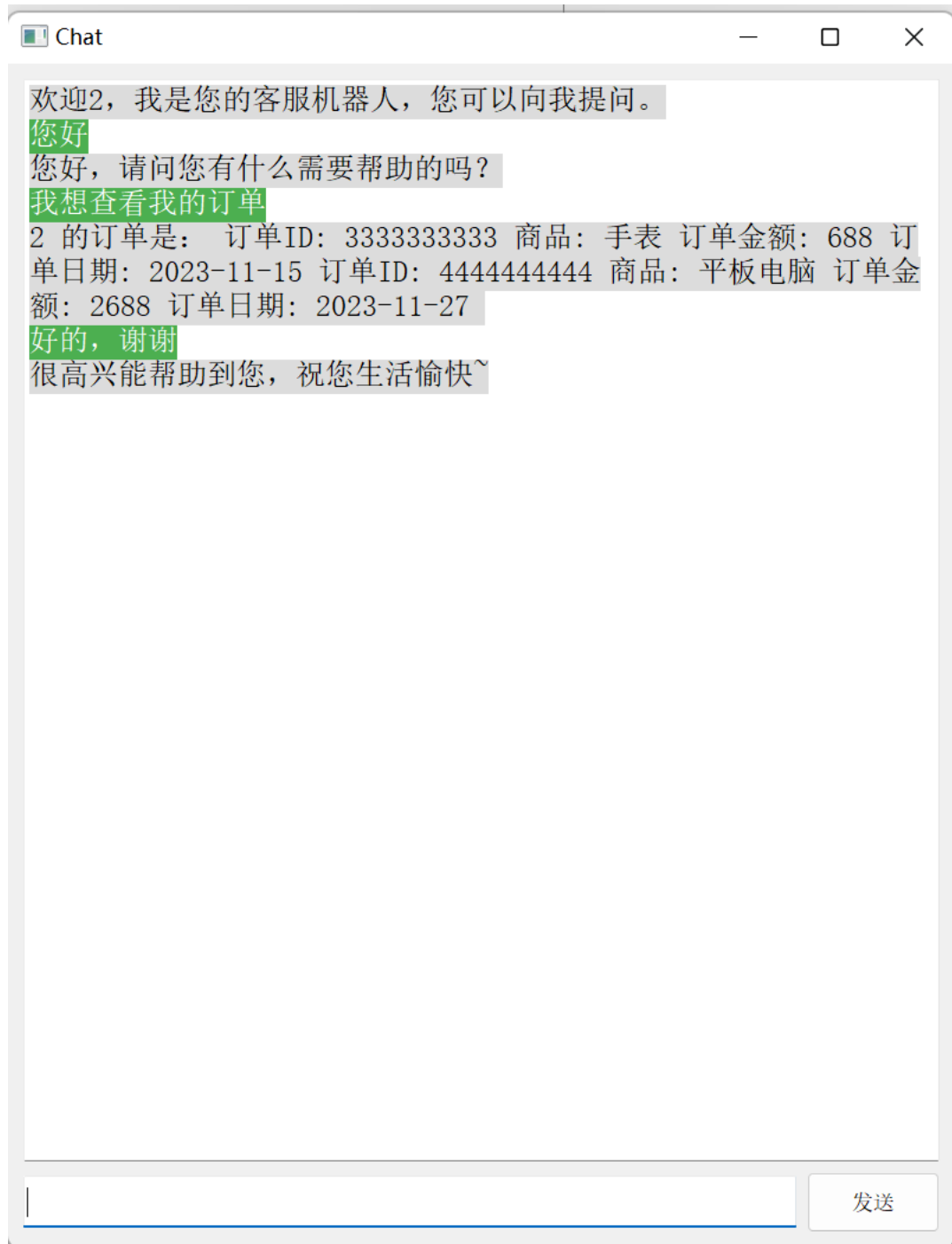


图 2-2 用户 2 的订单查询结果图

可以看到，用户 2 的订单也被成功查询，并输出出来。