Тестовое задание для python middle

Написать сервис на любом известном фреймворки.
Сервис выполняет заявку на провоз лыж.

Сервис должен:

1. **принимает на вход**
    a. **id брони это 6-ти значная строка может содержать только цифры и латинские буквы**
    b. **Фамилию на латинском**
2. **Сервис должен сделать запрос к заказу в сторонний сервис (реального url нет, для тестов использовать мок) пример ниже под номером 1**
3. **Обработать и вычленить нужную информацию из заказ:**
    a. **Все passengerId**
    b. **Все routeId**
    c. **Все baggageIds с equipmentType == ski**
4. **Из полученных данных сформировать запрос на добавление лыж в заказ (реального url нет, для тестов использовать мок) пример ниже под номером 2**
5. **Проверять результат на всех этапах**
6. **Сервис должен содержать пояснительную записку как его запустить и протестировать.**
7. **Сервис должен быть выложен в любой из известных открытых систем хранения кода например github, gitlab …**
8. **Компоненты сервиса и название не должны содержать приставок s7**
9. **Как плюс сервис должен сохранять все промежуточные результаты для разбора инцидентов (структуру придумать самому)**

1. Пример запроса к стороннему сервису заказа:

Пример запроса:

```
import requests
url = "??/orders"
querystring = {"number":"AAAAAA","passengerId":"ivanov"}
payload = ""
headers = {}
response = requests.request("GET", url, data=payload, headers=headers, params=querystring)
```

Пример ответ:


```
{"ancillariesPricings":[{"airId":"ef8ff876-9b29-448f-97ba-094898deef98","baggagePricings":[{"passengerIds":
["dKCLeweYNb6iDO66","qauJTpuMlDrASaty"],"passengerTypes":["ADT"],"purchaseType":"PAID","routeId":"
RyucZ4TVI1EseYCp","baggages":[{"id":"nqNNipOwlK7i9fRr","overWeight":true,"amount":1,"unit":"KG","weight":
{"amount":50,"unit":"KG"},"code":"0IK","descriptions":["EXCESS WEIGHT"],"registered":false},{"id":"
q0YIbjcv2zSx4JGK","overWeight":false,"amount":1,"unit":"PC","weight":{"amount":23,"unit":"KG"},"code":"0CC","
descriptions":["CHECKED BAG FIRST"],"registered":false},{"id":"KChsLeEHhHqEvGmw","overWeight":false,"amount":2,"
unit":"PC","weight":{"amount":23,"unit":"KG"},"code":"0CD","descriptions":["CHECKED BAG SECOND"],"registered":
false},{"id":"siEct88JoxGWpe5v","overWeight":false,"amount":1,"unit":"PC","code":"0DD","descriptions":["SNOW
SKI SNOWBOARD EQUIPMENT"],"registered":false,"equipmentType":"ski"}]},{"passengerIds":["dKCLeweYNb6iDO66","
qauJTpuMlDrASaty"],"passengerTypes":["ADT"],"purchaseType":"PAID","routeId":"iqCrFYw8oDTwVpWD","baggages":
[{"id":"xawp8dUZHYaJqmVS","overWeight":true,"amount":1,"unit":"KG","weight":{"amount":50,"unit":"KG"},"code":"
0IK","descriptions":["EXCESS WEIGHT"],"registered":false},{"id":"AzD5GiHPkxVruI3B","overWeight":false,"amount":
1,"unit":"PC","weight":{"amount":23,"unit":"KG"},"code":"0CC","descriptions":["CHECKED BAG FIRST"],"registered":
false},{"id":"7UPUB3KhGSI12ZXF","overWeight":false,"amount":2,"unit":"PC","weight":{"amount":23,"unit":"KG"},"
code":"0CD","descriptions":["CHECKED BAG SECOND"],"registered":false},{"id":"CMQs0BgMVGpAJcOP","overWeight":
false,"amount":1,"unit":"PC","code":"0DD","descriptions":["SNOW SKI SNOWBOARD EQUIPMENT"],"registered":false,"
equipmentType":"ski"}]}],"baggageDisabled":false,"seatsDisabled":false,"mealsDisabled":false,"upgradesDisabled":
true,"loungesDisabled":false,"fastTracksDisabled":false,"petsDisabled":true}]}
```

2. Пример запроса к стороннему сервису для формирование заказа:

Пример запроса:

```
import requests
url = "??/bags"
querystring = ""
payload = {"baggageSelections":[{"passengerId":"dKCLeweYNb6iDO66","routeId":"RyucZ4TVI1EseYCp","baggageIds":
["siEct88JoxGWpe5v"],"redemption":false},{"passengerId":"dKCLeweYNb6iDO66","routeId":"iqCrFYw8oDTwVpWD","
baggageIds":["CMQs0BgMVGpAJcOP"],"redemption":false}],{"passengerId":"qauJTpuMlDrASaty","routeId":"
RyucZ4TVI1EseYCp","baggageIds":["siEct88JoxGWpe5v"],"redemption":false},{"passengerId":"qauJTpuMlDrASaty","
routeId":"iqCrFYw8oDTwVpWD","baggageIds":["CMQs0BgMVGpAJcOP"],"redemption":false}]}
headers = {}
response = requests.request("PUT", url, data=payload, headers=headers, params=querystring)
```

Пример ответа:

```
   (Status_code 200):
{
  "shoppingCart": {....}
}
   (Status_code 200):
{
  "error": {
        "code": "conversation.not.found",
        "message": "      .",
  },
  "shoppingCart": null
}
```