

# Test Documentation

Author: Bartosz Koperek

## Contents

Overview .....	3
Used tools and languages .....	3
Technical details .....	3
Assumptions .....	3
Coding standard .....	3
Files structure .....	4
Test.robot .....	4
Configuration file .....	4
Exceptions .....	4
Dealing with timeouts .....	5
Installation guide .....	6
Test execution .....	7
Reports .....	7

## Overview

This test proves that the payment is declined with an invalid payment input.

## Used tools and languages

The test is developed in python, selenium and robot framework. Robot Framework is a generic test automation framework for acceptance testing with support for BDD - Gherkin syntax, providing extensive and clear logs and reports.

IDE: Pycharm with IntelliBot plugin.

## Technical details

Page object pattern was used. All pages are represented as separate classes and inherit from the BasePage class. The BasePage contains common methods for finding elements and operating on them. All selenium locators occurring on a certain page are stored in correlated class so that when developers change something in the source code of the website, a correction for the test has to be applied only in one class.

## Assumptions

- One way flight option is always chosen.
- It is possible to provide only the number of adults and teens as passengers.
- Passengers' details like first and last name are generated automatically as random strings.
- Flight date format: DD/MM/YYYY.
- Card expiration date format: MM/YYYY.
- All other strings required for payment and not passed as parameters are stored in configuration file.

## Coding standard

- Following PEP 8 python standard.
- File names – lower case with underscore as a separator.
- Class names – camel case.
- Function names – lower case with underscore as a separator.
- Descriptive function names so that a name reflects action on a website.
- No implicit time sleeps.
- All locators are stored in correlated classes as constant variables. They are upper case with prefix identifying tag type and suffix identifying locating method:  
TAGTYPE\_NAME\_LOCATINGMETHOD\_LOCATOR, example:  
INPUT\_ONEWAY\_ID\_LOCATOR

## Files structure

### --Test

- test.robot
- config
  - \_\_init\_\_.py
  - configfile.py
- Resources
  - \_\_init\_\_.py
  - decorator\_control.py
  - exceptions.py
  - page\_base.py
  - page\_choose\_flight.py
  - page\_go\_to\_checkout.py
  - page\_home.py
  - page\_payment.py
  - verify\_payment.py

All files are stored in Test folder with 2 subdirectories: config containing the test configuration file and Resources containing all python libraries.

## Test.robot

The acceptance level test is written in robot in Gherkin style.

Example test input:

```
Given I make a booking from 'Bologna' to 'Bucharest' on '16/04/2016' for '2' adults and '1' child
When I pay for booking with card details '5555 5555 5555 5557', '10/2018' and '265'
Then I should get payment declined message
```

The PaymentVerifier class is an interface between Robot framework and Python. It triggers all selenium actions and checks if a payment declined message was found. It is imported in test.robot file in Settings section:

```
*** Settings ***
Library          verify_payment.Payment_Verifier
```

## Configuration file

All parameters like url, timeouts and default strings are stored in a separate file in config\configfile.py

## Exceptions

Custom exceptions are defined in file exceptions.py. They are used to provide a clear output to a test owner when a test fails. The output should indicate what actually happened.

Example test input with a date without any flight from Bologna to Bucharest:

```
Given I make a booking from 'Bologna' to 'Bucharest' on '16/04/2016' for '2' adults and '1' child
When I pay for booking with card details '5555 5555 5555 5557', '10/2018' and '265'
Then I should get payment declined message
```

## Test Execution Log

SUITE

Test

Full Name:Test

Source:C:\Users\koperek\PycharmProjects\Ryanair\test.robot

Start / End / Elapsed:20160413 11:36:38.990 / 20160413 11:36:56.746 / 00:00:17.756

Status:1 critical test, 0 passed, 1 failed  
1 test total, 0 passed, 1 failed

TEST

Test Payment

Full Name:Test.Test Payment

Documentation:Author: Bartosz Koperek This test automates a booking up to a declined payment and verifies the payment result.

Start / End / Elapsed:20160413 11:36:39.282 / 20160413 11:36:56.744 / 00:00:17.462

Status:FAIL (critical)

Message:NoFlightsForThisDateError: There are no flights for this date!

KEYWORD

Given I make a booking from 'Bologna' to 'Bucharest' on '16/04/2016' for '2' adults and '1' child

Start / End / Elapsed:20160413 11:36:39.283 / 20160413 11:36:56.744 / 00:00:17.461

KEYWORD

verify\_paymentPayment\_Verifier.Fill In Flight Details \${flight\_from}, \${flight\_to}, \${flight\_date}, \${adults\_number}, \${children\_number}

Start / End / Elapsed:20160413 11:36:39.284 / 20160413 11:36:56.744 / 00:00:17.460

11:36:50.410INFOChecking if page is loaded properly.

11:36:51.968INFOPage <https://www.ryanair.com/ie/en/> loaded.

11:36:51.968INFOChoosing one way flight.

11:36:52.177INFOEntering flight from.

11:36:53.009INFOEntering flight to.

11:36:53.588INFOEntering flight date.

11:36:56.698INFOChecking if any flight is available on chosen date.

11:36:56.743FAILNoFlightsForThisDateError: There are no flights for this date!

Note: not all exceptions are handled due to a limited time, however this approach can be used to deal with situations where incorrect credit card number, credit card expiration date, email etc. are provided.

### Dealing with timeouts

No implicit waits were used as they can drive a test to fail in a certain conditions. Instead WebDriverWait method was used.

Example (waiting for an element with a timeout until it appears on a page):

```
element = WebDriverWait(self.driver,
timeout).until(EC.presence_of_element_located((By.ID,locator)))
```

Another important method where the WebDriverWait was used is wait\_for\_change in the BasePage class. This method deals with dynamic objects like ajax widgets and protects the test from failing when an element did not manage to load.

```
source = self.driver.page_source
def compare_source(driver):
    try:
        return source != driver.page_source
    except WebDriverException:
        pass
WebDriverWait(self.driver, timeout).until(compare_source)
```

## Installation guide

For Windows:

- Install python 2.7.6:
  - <https://www.python.org/download/releases/2.7.6/>
- Install setup tools:
  - Download and execute below file:
  - [https://bootstrap.pypa.io/ez\\_setup.py](https://bootstrap.pypa.io/ez_setup.py)

```
Installed c:\python27\lib\site-packages\setuptools-20.7.0-py2.7.egg
Processing dependencies for setuptools==20.7.0
Finished processing dependencies for setuptools==20.7.0
```

- Install pip:
  - <https://pypi.python.org/pypi/pip>

```
Installed c:\python27\lib\site-packages\pip-8.1.1-py2.7.egg
Processing dependencies for pip==8.1.1
Finished processing dependencies for pip==8.1.1
```

- Install selenium:
  - pip install selenium

```
D:\userdata\koperek\Downloads\python\pip-8.1.1.tar\pip-8.1.1>pip install selenium
Collecting selenium
C:\Python27\lib\site-packages\pip-8.1.1-py2.7.egg\pip\_vendor\requests\packages\urllib3\uti
ing: An HTTPS request has been made, but the SNI (Subject Name Indication) extension to TLS
tform. This may cause the server to present an incorrect TLS certificate, which can cau
information, see https://urllib3.readthedocs.org/en/latest/security.html#snimissingwarn
SNIMissingWarning
C:\Python27\lib\site-packages\pip-8.1.1-py2.7.egg\pip\_vendor\requests\packages\urllib3
rmWarning: A true SSLContext object is not available. This prevents urllib3 from config
ause certain SSL connections to fail. For more information, see https://urllib3.readthe
insecureplatformwarning.
InsecurePlatformWarning
Using cached selenium-2.53.1-py2-none-any.whl
Installing collected packages: selenium
Successfully installed selenium-2.53.1
```

- Install Robot framework:
  - pip install robotframework

```
D:\userdata\koperek\Downloads\python\pip-8.1.1.tar\pip-8.1.1>pip install robotframework
Collecting robotframework
C:\Python27\lib\site-packages\pip-8.1.1-py2.7.egg\pip\_vendor\requests\packages\urllib3\uti
ing: An HTTPS request has been made, but the SNI (Subject Name Indication) extension to TLS
tform. This may cause the server to present an incorrect TLS certificate, which can cause v
information, see https://urllib3.readthedocs.org/en/latest/security.html#snimissingwarning.
SNIMissingWarning
C:\Python27\lib\site-packages\pip-8.1.1-py2.7.egg\pip\_vendor\requests\packages\urllib3\uti
rmWarning: A true SSLContext object is not available. This prevents urllib3 from configurin
ause certain SSL connections to fail. For more information, see https://urllib3.readthedocs
insecureplatformwarning.
InsecurePlatformWarning
Using cached robotframework-3.0.tar.gz
Installing collected packages: robotframework
Running setup.py install for robotframework ... done
Successfully installed robotframework-3.0
```

For Mac OS:

- Install python 2.7.6:
  - <https://www.python.org/download/releases/2.7.6/>
- Install selenium:
  - `sudo easy_install selenium`
- Install Robot framework:
  - [https://mitnk.com/wiki/2011/10/install\\_robot\\_framework\\_on\\_mac\\_os\\_x/](https://mitnk.com/wiki/2011/10/install_robot_framework_on_mac_os_x/)

Make sure that python is added to the PATH environment variable.

## Test execution

To run the test execute the pybot command in a directory where test.robot file is located:

```
pybot --pythonpath Resources --loglevel INFO -- test.robot
```

or for more detailed logs:

```
pybot --pythonpath Resources --loglevel DEBUG -- test.robot
```

Example console output:

```
C:\Users\koperek\PycharmProjects\Ryanair>pybot --pythonpath Resources --loglevel INFO -- test.robot
=====
Test
=====
Test Payment :: Author: Bartosz Koperek                                     ! PASS !
=====
Test
1 critical test, 1 passed, 0 failed                                     ! PASS !
1 test total, 1 passed, 0 failed
=====
Output:  C:\Users\koperek\PycharmProjects\Ryanair\output.xml
Log:     C:\Users\koperek\PycharmProjects\Ryanair\log.html
Report:  C:\Users\koperek\PycharmProjects\Ryanair\report.html
C:\Users\koperek\PycharmProjects\Ryanair>
```

## Reports

Pybot automatically generates three files which can be used for reporting:

- log.html – the test result details,
- report.html - the test result summary,
- output.xml - the test results in a portable XML format for integration with other tools.

The report.html file contains all executed tests results with basic info like Test Name, Elapsed time, Start / End date:

Test Details							LOG
Totals Tags Suites Search							
Type:	<input type="radio"/> Critical Tests <input checked="" type="radio"/> All Tests						
Status:	1 total, 1 passed, 0 failed						
Total Time:	00:00:58.721						
Name	Documentation	Tags	Crit.	Status	Message	Elapsed	Start / End
Test Payment	Author: Bartosz Koperek This test automates a booking up to a declined payment and verifies the payment result. The test is passed if payment declined message is found. Requirements: date format: DD/MM/YYYY, card expiration date format: MM/YYYY, correct credit card number		yes	PASS		00:00:58.721	20160412 22:41:41.003 20160412 22:42:39.724

It also contains a link to log.html file with more detailed info about the test execution like Test Statistics and Test Execution Log. The execution log is very helpful, especially for debugging. All keywords can be expanded to show python logs.

## Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:59	<div></div>
All Tests	1	1	0	00:00:59	<div></div>

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					<div></div>

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
<a href="#">Test</a>	1	1	0	00:00:59	<div></div>

## Test Execution Log

[-] **SUITE** Test

Full Name:

Test

Source:

[C:\Users\koperek\PycharmProjects\Ryanair\test.robot](#)

Start / End / Elapsed:

20160412 22:41:40.498 / 20160412 22:42:39.725 / 00:00:59.227

Status:

1 critical test, 1 passed, 0 failed  
1 test total, 1 passed, 0 failed

[-] **TEST** Test Payment

Full Name:

Test.Test Payment

Documentation:

Author: Bartosz Koperek This test automates a booking up to a declined payment and verifies the payment result.  
format: MM/YYYY, correct credit card number

Start / End / Elapsed:

20160412 22:41:41.003 / 20160412 22:42:39.724 / 00:00:58.721

Status:

**PASS** (critical)

[+] **KEYWORD** Given I make a booking from 'DUB' to 'SXF' on '11/06/2016' for '2' adults and '1' child

[+] **KEYWORD** When I pay for booking with card details '5555 5555 5555 5557', '10/2018' and '265'

[-] **KEYWORD** Then I should get payment declined message

Start / End / Elapsed:

20160412 22:42:39.647 / 20160412 22:42:39.724 / 00:00:00.077

[-] **KEYWORD** `verify_payment.Payment_Verifier.Check Payment Result`

Start / End / Elapsed:

20160412 22:42:39.647 / 20160412 22:42:39.724 / 00:00:00.077

22:42:39.647

**INFO**

Checking if payment declined message appeared.

22:42:39.724

**INFO**

Payment declined message found

8