

Robot Framework RequestsLibrary



Sample Test Cases with Shopping Cart

- Test Cases: Checkout Diner Set from Store Service
- Tasks:

1. Get Product List

- input URL PATH: /api/v1/product
- input HTTP Method: GET
- input HTTP Headers: Accept application/json
- assert http status code should be 200
- assert products should have 2 items
- assert products should contain '43 Piece dinner Set'

2. Get Product Detail

- input URL PATH: /api/v1/product/{product id}
- input HTTP Method: GET
- input HTTP Headers: Accept application/json
- assert http status code should be 200
- assert product name should contain '43 Piece dinner Set'

Sample Test Cases with Shopping Cart

- Test Cases: Checkout Diner Set from Store Service
- Tasks:

3.Order Diner Set

- input URL PATH: /api/v1/order
- input HTTP Method: POST
- input HTTP Headers: Accept application/json
- input HTTP Headers: Content-Type application/json
- input HTTP **Body**
- assert http status code should be 200
- assert product total price should be 14.95

4.Confirm Payment

- input URL PATH: /api/v1/confirmPayment
- input HTTP Method: POST
- input HTTP Headers: Accept application/json
- input HTTP Headers: Content-Type application/json
- input HTTP **Body**
- assert http status code should be 200
- assert notification message should be 'วันที่ชำระเงิน 1/3/2020 13:30:00 หมายเลขคำสั่งซื้อ 8004359104 คุณสามารถติดตามสินค้าผ่านช่องทาง Kerry หมายเลข 1785261900'

Sample Test Cases with Shopping Cart

PATH: /api/v1/product

Method: GET

Headers: Accept application/json

Get Product List

http status code is 200

products have 2 items

products contain '43 Piece dinner Set'

PATH: /api/v1/product/{product id}

Method: GET

Headers: Accept application/json

Get Product Detail

http status code is 200

product name is '43 Piece dinner Set'

PATH: /api/v1/order

Method: POST

Headers: Accept application/json

Headers: Content-Type
application/json

Body

Order

http status code is 200

product total price is 14.95

PATH: /api/v1/confirmPayment

Method: POST

Headers: Accept application/json

Headers: Content-Type
application/json

Body

Confirm Payment

notification message is 'วันเวลาที่

ชำระเงิน 1/3/2020 13:30:00

หมายเลขคำสั่งซื้อ 8004359104 คุณ

สามารถติดตามสินค้าผ่านช่องทาง

Kerry หมายเลข 1785261900'

©2020 Siam Chamnankit Company Limited



Sample Test Cases with Shopping Cart

- How to pass the `id` from 'Get Product List' to 'Get Product Detail' `id`?
- How to pass the return `order id` from 'Order Diner Set' to 'Confirm Payment' `order id`?
- Which parameters in each task that we have to concern?
 - consequence
 - specific



HTTP RequestsLibrary(Python)

Library for HTTP level testing using Python Requests internally.

[lean more](#)



Prerequisite

1. Python3
2. Robot Framework
`pip install robotframework`
3. HTTP RequestsLibrary(Python)
`pip install robotframework-requests`

Sections in Test Case Files

Robot Framework data is defined in different sections.

[learn more](#)

```
*** Settings ***
```

```
*** Variables ***
```

```
*** Test Cases ***
```

```
*** Keywords ***
```


Settings in Test Case Files

```
import RequestsLibrary
```

checkout-success.robot

```
*** Settings ***
```

```
Library           RequestsLibrary
```

Test Cases

in Test Case Files

Get Product List

Assert Product List(Task)

Request:

Get Product List

url = api/v1/product

Get Product List

header = Accept=text/xml

Response:

Assert

http status code = 200

Assert

product total = 2

Test Cases in Test Case Files

Create Checkout Diner Set(Test case)

checkout-success.robot

*** Test Cases ***

Checkout Dinner Set

Get Product List

Get Product Detail

Order Product

Confirm Payment

Test Cases in Test Case Files

Get Product List(Keywords)

checkout-success.robot

```
*** Keywords ***
```

```
Get Product List
```

```
    Create Session    toy_store    http://localhost:8000
```

RequestsLibrary

Create Session API

[learn more](#)

Create Session	<i>alias, url, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=False, debug=0, max_retries=3, backoff_factor=0.1, disable_warnings=0, retry_status_list=[], retry_method_list=['DELETE', 'OPTIONS', 'HEAD', 'PUT', 'GET', 'TRACE']</i>
-----------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Create Session: create a HTTP session to a server

alias Robot Framework alias to identify the session

url Base url of the server

headers Dictionary of default headers

cookies Dictionary of cookies

auth List of username & password for HTTP Basic Auth

timeout Connection timeout

proxies Dictionary that contains proxy urls for HTTP and HTTPS communication

verify Whether the SSL cert will be verified. A CA_BUNDLE path can also be provided.

debug Enable http verbosity option more information

https://docs.python.org/2/library/httplib.html#httplib.HTTPConnection.set_debuglevel

max_retries Number of maximum retries each connection should attempt. By default it will retry 3 times in case of connection errors only. A 0 value will disable any kind of retries regardless of other retry settings. In case the number of retries is reached a retry exception is raised.

disable_warnings Disable requests warning useful when you have large number of testcases

backoff_factor Introduces a delay time between retries that is longer after each retry. eg. if backoff_factor is set to 0.1 the sleep between attempts will be: 0.0, 0.2, 0.4 More info here: <https://urllib3.readthedocs.io/en/latest/reference/urllib3.util.html>

retry_method_list List of uppercased HTTP method verbs where retries are allowed. By default retries are allowed only on HTTP requests methods that are considered to be idempotent (multiple requests with the same parameters end with the same state). eg. set to ['POST', 'GET'] to retry only those kind of requests.

retry_status_list List of integer HTTP status codes that, if returned, a retry is attempted. eg. set to [502, 503] to retry requests if those status are returned. Note that max_retries must be greater than 0.

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robot
```

Test Cases in Test Case Files

Get Product List

checkout-success.robot

```
*** Keywords ***
```

```
Get Product List
```

```
    Create Session    toy_store    http://localhost:8000
```

```
    &{accept}=    Create Dictionary    Accept=text/xml
```

```
    ${productList}=    Get Request    toy_store    /api/v1/product
```

```
    headers=&{accept}
```

RequestsLibrary

Get Request **API**

[learn more](#)

Get Request

alias, uri, headers=None, data=None, json=None, params=None, allow_redirects=None, timeout=None

Send a GET request on the session object found using the given *alias*

alias that will be used to identify the Session object in the cache

uri to send the GET request to

params url parameters to append to the uri

headers a dictionary of headers to use with the request

data a dictionary of key-value pairs that will be urlencoded and sent as GET data or binary data that is sent as the raw body content

json a value that will be json encoded and sent as GET data if data is not specified

allow_redirects Boolean. Set to True if POST/PUT/DELETE redirect following is allowed.

timeout connection timeout

Test Cases in Test Case Files

Get Product List

Assert HTTP Status Code

checkout-success.robot

```
*** Keywords ***
```

```
Get Product List
```

```
    Create Session    toy_store    http://localhost:8000
```

```
    &{accept}=    Create Dictionary    Accept=text/xml
```

```
    ${productList}=    Get Request    toy_store    /api/v1/product
```

```
    headers=&{accept}
```

```
    Status Should Be    200    ${productList}
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robot
```

Test Cases in Test Case Files

Assert Product Total

checkout-success.robot

```
*** Keywords ***
```

```
Get Product List
```

```
    Create Session    toy_store    http://localhost:8000
```

```
    &{accept}=    Create Dictionary    Accept=text/xml
```

```
    ${productList}=    Get Request    toy_store    /api/v1/product
```

```
    headers=&{accept}
```

```
    Status Should Be    200    ${productList}
```

```
    Should Be Equal    ${productList.json()["total"]}    ${2}
```



Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robot
```

Test Cases in Test Case Files

Assert Product Total(Task)

checkout-success.robot

*** Keywords ***

Get Product List

Create Session toy_store http://localhost:8000

&{accept}= Create Dictionary Accept=text/xml

\${productList}= Get Request toy_store /api/v1/product

headers=&{accept}

Status Should Be 200 \${productList}

\${text}= Decode Bytes To String \${productList.content} UTF-8

\${xml} Parse Xml \${text}

\${total}= Get Element \${xml} total

Should Be Equal As Integers \${total.text} \${2}

@{products}= Get Element \${xml} products

Length Should Be \${products} 2



Variables in Robotframework

Dictionary

[learn more](#)

```
response =  
    <product-list>                                ${xml}  
        <total>2</total>                          Get Element    ${xml}    total  
        <products>                                Get Element    ${xml}    products  
            <product>  
                <id>1</id>  
                <product_name>Balance Training Bicycle</product_name>  
                <product_price>119.95</product_price>  
                <product_image>/Balance_Training_Bicycle.png</product_image>  
            </product>  
            <product>  
                <id>2</id>  
                <product_name>43 Piece dinner Set</product_name>  
                <product_price>12.95</product_price>  
                <product_image>/43_Piece_dinner_Set.png</product_image>  
            </product>  
        </products>  
    </product-list>
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The command is '\$ robot checkout-success.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robot
```

Test Cases in Test Case Files

Get Product Detail

Assert Product Detail

Request:

Get Product Detail /api/v1/product/2

Get Product Detail product id = 2

Response:

Assert product_id = 2

Assert product_name = 43 Piece dinner Set



Test Cases in Test Case Files

Get Product Detail

Assert Product Detail

checkout-success.robot

*** Test Cases ***

Get Product Detail

Create Session toy_store http://localhost:8000

&{accept}= Create Dictionary Accept=text/xml

\${productDetail}= Get Request toy_store /api/v1/product/2

headers=&{accept}

Request Should Be Successful \${productDetail}

\${xml} Parse Xml \${productDetail.content}

\${name}= Get Element Text \${xml} product_name

Should Be Equal \${name} 43 Piece dinner Set

Test Cases

in Test Case Files

Order Dinner Set

Assert Order Status

Request:

Order Dinner Set

Order Dinner Set

Order Dinner Set

list of [product id = 2, quantity = 1]

shipping address

headers = Accept=text/xml

Content-Type=text/xml

Response:

Assert

total price = 14.95

RequestsLibrary

Get Request **API**

[learn more](#)

Post Request

alias, uri, data=None, json=None, params=None, headers=None, files=None, allow_redirects=None, timeout=None

Send a POST request on the session object found using the given *alias*

alias that will be used to identify the Session object in the cache

uri to send the POST request to

data a dictionary of key-value pairs that will be urlencoded and sent as POST data or binary data that is sent as the raw body content or passed as such for multipart form data if **files** is also defined or file descriptor retrieved by Get File For Streaming Upload

json a value that will be json encoded and sent as POST data if files or data is not specified

params url parameters to append to the uri

headers a dictionary of headers to use with the request

files a dictionary of file names containing file data to POST to the server

allow_redirects Boolean. Set to True if POST/PUT/DELETE redirect following is allowed.

timeout connection timeout

Test Cases in Test Case Files

checkout-success.robot

Order Dinner Set

Assert Order Status

*** Keywords ***

Order Product

```
Create Session    toy_store    http://localhost:8000
${order}=        Encode String To Bytes    <orders><cart><order><product_id>2</
product_id><quantity>1</quantity></order></cart><shipping_method>Kerry</
shipping_method><shipping_address>405/37 ถ.มหิดล</
shipping_address><shipping_sub_district>ท่าศาลา</
shipping_sub_district><shipping_district>เมือง</
shipping_district><shipping_province>เชียงใหม่</
shipping_province><shipping_zip_code>50000</
shipping_zip_code><recipient_name>ณัฐญา ชุตินทร</
recipient_name><recipient_phone_number>0970809292</recipient_phone_number></
orders>          UTF-8
```

```
&{post_headers}=    Create Dictionary    Accept=text/xml
Content-Type=text/xml
```

```
${orderStatus}=    Post Request    toy_store    /api/v1/order
data=${order}      headers=&{post_headers}
```

Test Cases in Test Case Files

Order Dinner Set

Assert Order Status

checkout-success.robot

```
*** Keywords ***
```

```
Order Product
```

```
...
```

```
    ${orderStatus}=    Post Request    toy_store    /api/v1/order  
    data=${order}      headers=&{post_headers}
```

```
    Status Should Be    200    ${orderStatus}  
    ${xml}    Parse Xml    ${orderStatus.content}  
    ${total_price}=    Get Element Text    ${xml}    total_price  
    Should Be Equal    ${total_price}    14.95  
    ${order_id}=    Get Element Text    ${xml}    order_id
```

```
    Set Test Variable    ${order_id}    ${order_id}
```

Order Status Response

```
<order>  
  <order_id>8004359122</order_id>  
  <total_price>14.95</total_price>  
</order>
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The command is '\$ robot checkout-success.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robot
```

Test Cases

in Test Case Files

Confirm Payment

Assert Confirm Payment Status(Task)

Request:

Confirm Payment

/api/v1/confirmPayment

Confirm Payment

order id = \${order_id}

Confirm Payment

payment method

Confirm Payment

total price = 14.95

Confirm Payment

headers = Accept=application/json

Content-Type=application/json

Response:

Assert

status code = 200

Assert

payment date = format d/m/yyyy HH:MM:SS

Assert

tracking id = format number 10 digits

Test Cases in Test Case Files

Confirm Payment

checkout-success.robot

Assert Confirm Payment Status

*** Keywords ***

Confirm Payment

```
Create Session    toy_store    http://localhost:8000
${confirmPayment}=    Encode String To Bytes    <confirm-
payment><order_id>${order_id}</order_id><payment_type>credit</
payment_type><type>visa</type><card_number>4719700591590995</
card_number><cvv>752</cvv><expired_month>7</
expired_month><expired_year>20</expired_year><card_name>Karnwat
Wongudom</card_name><total_price>${total_price}</total_price></confirm-
payment>    UTF-8
```



Test Cases in Test Case Files

checkout-success.robot

Confirm Payment

*** Keywords ***

Checkout Diner Set

...

```
{confirmPaymentStatus}=      Post Request      toy_store      /api/v1/  
confirmPayment      data={confirmPayment}      headers=&{post_headers}  
Request Should Be Successful      {confirmPaymentStatus}
```

```
{text}=      Decode Bytes To String      {confirmPaymentStatus.content}      UTF-8
```

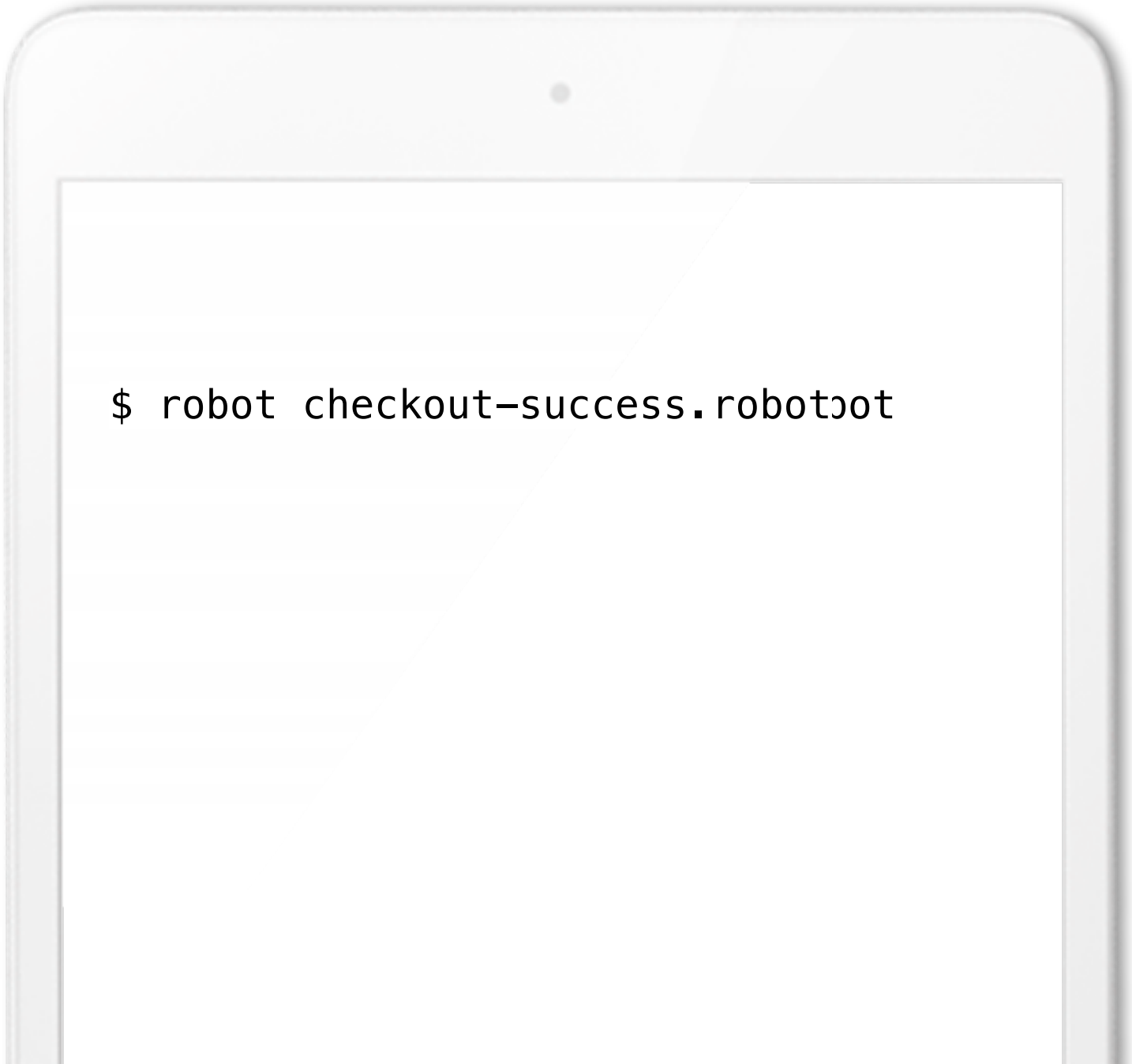
```
{xml}      Parse Xml      {text}  
{message}=      Get Element Text      {xml}  
{payment_date}=      Get Element Text      {xml}      payment_date  
Should Match Regexp      {payment_date}      ^\\d{1,2}/\\d{1,2}/\\d{4} \\  
\\d{2}:\\d{2}:\\d{2}$  
{tracking_id}=      Get Element Text      {xml}      tracking_id  
Should Match Regexp      {tracking_id}      ^\\d{10}$
```

Confirm Payment Status

Response

```
<notify_message>  
  <payment_date>26/11/2020 16:45:39</payment_date>  
  <order_id>867152161</order_id>  
  <shipping>Kerry</shipping>  
  <tracking_id>1280567270</tracking_id>  
</notify_message>
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robotoot
```

Test Cases in Test Case Files

Create Checkout Diner Set(Test case)

Delete Session

checkout-success.robot

```
*** Test Cases ***
```

```
Checkout Diner Set
```

```
...
```

```
Delete All Sessions
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in black text. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success.robot
```



HTTP RequestsLibrary(Python)

*** Variables ***

[lean more](#)

Test Cases in Test Case Files

Hard Code!

checkout-success-variables.robot

```
*** Keywords ***
```

```
Get Product List
```

```
Create Session    toy_store    http://localhost:8000
```

```
&{accept}=    Create Dictionary    Accept=text/xml
```

```
${productList}=    Get Request    toy_store    /api/v1/product
```

```
headers=&{accept}
```

```
Status Should Be    200    ${productList}
```


Test Cases in Test Case Files

Hard Code!

checkout-success-variables.robot

*** Variables ***

`${toy_store}`

`${URL}` `http://localhost:8000`

`&{ACCEPT}` `Accept=text/xml`

`${OK}` `200`

*** Keywords ***

Get Product List

Create Session `${toy_store}` `${URL}`

`${productList}=` Get Request `${toy_store}` `/api/v1/`

product headers=`&{ACCEPT}`

Status Should Be `${OK}` `${productList}`

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The device has a thin grey border and a small circular camera lens at the top center.

```
$ robot checkout-success-variables.robot
```

Test Cases in Test Case Files

Hard Code!

checkout-success-variables.robot

*** Keywords ***

Order Product

...

```
&{orders}=      Encode String To Bytes      <orders><cart><order><product_id>${id}</product_id><quantity>2</quantity></order></cart><shipping_method>Kerry</shipping_method><shipping_address>405/37ถ.มหิดล</shipping_address><shipping_sub_district>ท่าศาลา</shipping_sub_district><shipping_district>เมือง</shipping_district><shipping_province>เชียงใหม่</shipping_province><shipping_zip_code>50000</shipping_zip_code><recipient_name>ณัฐญา ชุตินุตร</recipient_name><recipient_phone_number>0970809292</recipient_phone_number></orders>      UTF-8
```

```
&{post_headers}=      Create Dictionary      Accept=text/xmlContent-Type=text/xml
```

Test Cases in Test Case Files

checkout-success-variables.robot

Hard Code!

*** Variables ***

\${toy_store}

\${URL} http://167.99.77.238:8000

&{CONTENT_TYPE} Content-Type=text/xml

&{ACCEPT} Accept=text/xml

&{POST_HEADERS} &{ACCEPT} &{CONTENT_TYPE}

\${ORDER_TEMPLATE} <orders><cart><order><product_id>2</product_id><quantity>1</quantity></order></cart><shipping_method>Kerry</shipping_method><shipping_address>405/37 ถ.มหิดล</shipping_address><shipping_sub_district>ท่าศาลา</shipping_sub_district><shipping_district>เมือง</shipping_district><shipping_province>เชียงใหม่</shipping_province><shipping_zip_code>50000</shipping_zip_code><recipient_name>ณัฐญา ชุตินบุตร</recipient_name><recipient_phone_number>0970809292</recipient_phone_number></orders>

*** Keywords ***

Order Product

...

&{orders}= Encode String To Bytes \${ORDER_TEMPLATE}

\${orderStatus}= Post Request \${toy_store} /api/v1/order

json=\${orders} headers=&{POST_HEADERS}

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The device has a thin grey border and a small circular camera lens at the top center.

```
$ robot checkout-success-variables.robot
```

Test Cases in Test Case Files

Hard Code!

checkout-success-variables.robot

```
*** Keywords ***
```

```
Confirm Payment
```

```
...
```

```
${confirmPayment}=    Encode String To Bytes    <confirm-  
payment><order_id>${order_id}</  
order_id><payment_type>credit</payment_type><type>visa</  
type><card_number>4719700591590995</card_number><cvv>752</  
cvv><expired_month>7</expired_month><expired_year>20</  
expired_year><card_name>Karnwat Wongudom</  
card_name><total_price>${total_price}</total_price></confirm-  
payment>
```



Test Cases in Test Case Files

Hard Code!

checkout-success-variables.robot

```
*** Variables ***
```

```
${CONFIRM_PAYMENT_TEMPLATE}    <confirm-payment><order_id>\${order_id}</  
order_id><payment_type>credit</payment_type><type>visa</  
type><card_number>4719700591590995</card_number><cvv>752</  
cvv><expired_month>7</expired_month><expired_year>20</  
expired_year><card_name>Karnwat Wongudom</card_name><total_price>$  
{total_price}</total_price></confirm-payment>
```

```
*** Keywords ***
```

```
Confirm Payment
```

```
...
```

```
    ${confirmPayment}=    Encode String To Bytes    ${CONFIRM_PAYMENT_TEMPLATE}
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The device has a thin grey border and a small circular camera lens at the top center.

```
$ robot checkout-success-variables.robot
```


Run with Options

[learn more](#)

```
$ robot -v URL:https://xxx.xxx.xx.x  
checkout-success-variables.robot
```

Test Cases in Test Case Files

checkout-success-variables.robot

*** Settings ***

Library RequestsLibrary

Test Setup Create Session \${toy_store} \${URL}

Test Teardown Delete All Sessions

*** Test Cases ***

Checkout Diner Set

...

~~Delete All Sessions~~

*** Keywords ***

Get Product List

~~Create Session \${toy_store} \${URL}~~

Get Product Detail

~~Create Session \${toy_store} \${URL}~~

Order Product

~~Create Session \${toy_store} \${URL}~~

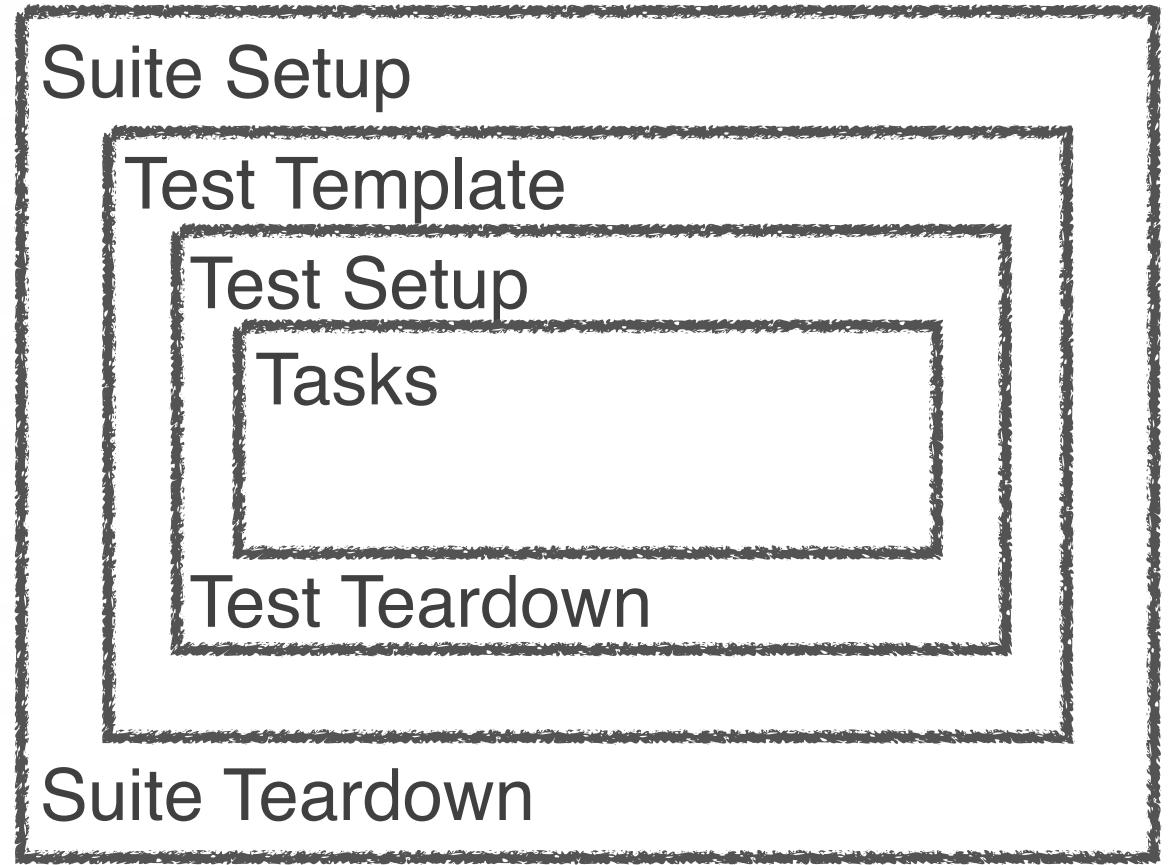
Confirm Payment

~~Create Session \${toy_store} \${URL}~~

Execution Flow in **Test Case Files**

Robot Framework data is defined in different sections.

[learn more](#)





HTTP RequestsLibrary(Python)

*** Keywords ***

[Arguments]

[learn more](#)

Test Cases in Test Case Files

```
*** Test Cases ***
```

```
Checkout Diner Set
```

```
    Get Product List
```

```
    Get Product Detail
```

```
    Order Product
```

```
    Confirm Payment
```

```
*** Keywords ***
```

```
...
```



Test Cases in Test Case Files

checkout-success-keywords.robot

create Get Product List keyword

```
*** Test Cases ***
```

```
Checkout Diner Set
```

```
    Get Product List
```

```
    ...
```

```
*** Keywords ***
```

```
Get Product List
```

```
    ${productList}=    Get Request    ${toy_store}    /api/v1/
```

```
    product    headers=&{ACCEPT}
```

```
    Status Should Be    200    ${productList}
```

```
    ${text}=    Decode Bytes To String    ${productList.content}
```

```
    UTF-8
```

```
    ${xml}    Parse Xml    ${text}
```

```
    ${total}=    Get Element    ${xml}    total
```

```
    Should Be Equal As Integers    ${total.text}    ${2}
```

Test Cases in Test Case Files

checkout-success-keywords.robot

create Get Product List keyword

```
*** Test Cases ***
```

```
Checkout Diner Set
```

```
    Get Product List    ${2}
```

```
    ...
```

```
*** Keywords ***
```

```
Get Product List
```

```
    [Arguments]    ${expected_total}
```

```
    ${productList}=    Get Request    ${toy_store}    /api/v1/product
```

```
    headers=&{ACCEPT}
```

```
    Status Should Be    200    ${productList}
```

```
    ${text}=    Decode Bytes To String    ${productList.content}    UTF-8
```

```
    ${xml}    Parse Xml    ${text}
```

```
    ${total}=    Get Element    ${xml}    total
```

```
    Should Be Equal As Integers    ${total.text}    ${expected_total}
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-keywords.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-keywords.robot
```


Test Cases in Test Case Files

Get Product Detail

Assert Product Detail(Task)

Request:

Get Product Detail product id = 2

Response:

Assert product id = 2

Assert product name = 43 Piece dinner Set



Test Cases in Test Case Files

checkout-success-keywords.robot

*** Test Cases ***

Checkout Diner Set

Get Product List \${2}

Get Product Detail

...

*** Test Cases ***

Get Product Detail

\${productDetail}= Get Request \${toy_store} /api/v1/product/2

headers=&{ACCEPT}

Request Should Be Successful \${productDetail}

\${xml} Parse Xml \${productDetail.content}

\${name}= Get Element Text \${xml} product_name

Should Be Equal \${name} 43 Piece dinner Set



Test Cases in Test Case Files

checkout-success-keywords.robot

*** Test Cases ***

Checkout Diner Set

Get Product List \${2}

Get Product Detail 2 43 Piece dinner Set

...

*** Test Cases ***

Get Product Detail

[Arguments] \${product_id} \${product_name}

\${productDetail}= Get Request \${toy_store} /api/v1/product/\$

{product_id} headers=&{ACCEPT}

Request Should Be Successful \${productDetail}

\${xml} Parse Xml \${productDetail.content}

\${name}= Get Element Text \${xml} product_name

Should Be Equal \${name} \${product_name}



Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-keywords.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-keywords.robot
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-keywords.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-keywords.robot
```

Test Cases in Test Case Files

checkout-success-keywords.robot

Order Dinner Set

Assert Order Status(Task)

Request:

Order Dinner Set

Order Dinner Set

Order Dinner Set

list of [product id = 2, quantity = 1]

shipping address

headers = Accept=text/xml

Content-Type=text/xml

Response:

Assert

total price = 14.95

Test Cases in Test Case Files

Hard Code!

checkout-success-keywords.robot

```
*** Test Cases ***
```

```
Checkout Diner Set
```

```
    Get Product List    ${2}
```

```
    Get Product Detail    2    43 Piece dinner Set
```

```
    Order Diner Set    2    1    14.95
```

```
    Confirm Payment
```

```
*** Keywords ***
```

```
...
```



Test Cases in Test Case Files

checkout-success-keywords.robot

Hard Code!

```
*** Variables ***
${ORDER_TEMPLATE}

...           <orders>
...           <cart>
...           <order>
...             <product_id>\${product_id}</product_id>
...             <quantity>\${quantity}</quantity>
...           </order>
...         </cart>
...         <shipping_method>Kerry</shipping_method>
...         <shipping_address>405/37 ถ.มหิดล</shipping_address>
...         <shipping_sub_district>ท่าศาลา</
shipping_sub_district>
...         <shipping_district>เมือง</shipping_district>
...         <shipping_province>เชียงใหม่</shipping_province>
...         <shipping_zip_code>50000</shipping_zip_code>
...         <recipient_name>ณัฐญา ชุติบุตร</recipient_name>
...         <recipient_phone_number>0970809292</
recipient_phone_number>
...       </orders>
```


Test Cases in Test Case Files

checkout-success-keywords.robot

Hard Code!

```
*** Keywords ***
```

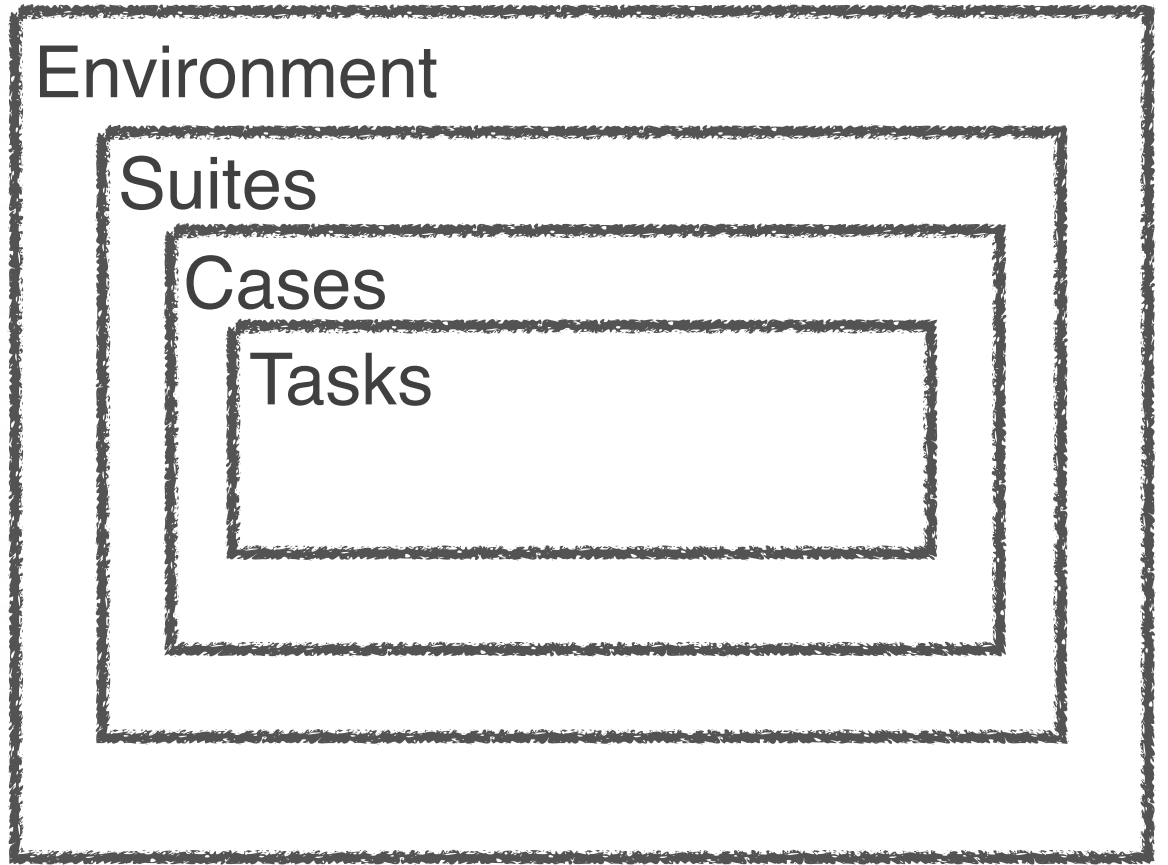
```
Order Diner Set
```

```
[Arguments]      ${product_id}    ${quantity}    ${expected_total_price}
${order_message}=    Replace Variables    ${ORDER_TEMPLATE}
${order}=    Encode String To Bytes    ${order_message}    UTF-8
${orderStatus}=    Post Request    ${toy_store}    /api/v1/order
data=${order}    headers=&{POST_HEADERS}
Status Should Be    200    ${orderStatus}
${xml}    Parse Xml    ${orderStatus.content}
${actual_total_price}=    Get Element Text    ${xml}    total_price
Should Be Equal    ${expected_total_price}    ${actual_total_price}
${order_id}=    Get Element Text    ${xml}    order_id
Set Test Variable    ${order_id}    ${order_id}
```

Scope in Test Case Files

Robot Framework data is defined in different sections.

[learn more](#)



Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-keywords.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-keywords.robot
```

Test Cases

in Test Case Files

Confirm Payment

Assert Confirm Payment Status(Task)

Request:

Confirm Payment

Confirm Payment

Confirm Payment

Confirm Payment

`order id = ${order_id}`

`payment method`

`total price = 14.95`

`headers = Accept=text/xml`

`Content-Type=text/xml`

Response:

Assert

`status code = 200`

Assert

`payment date = format d/m/yyyy HH:MM:SS`

Assert

`tracking id = format number 10 digits`

Test Cases in Test Case Files

Hard Code!

checkout-success-keywords.robot

```
*** Test Cases ***
```

```
Checkout Diner Set
```

```
    Get Product List
```

```
    Get Product Detail      2      43 Piece dinner Set
```

```
    Order Diner Set        2      1      14.95
```

```
    Confirm Payment        14.95
```

```
*** Keywords ***
```

```
...
```



Test Cases in Test Case Files

checkout-success-keywords.robot

Hard Code!

```
*** Variables ***
${CONFIRM_PAYMENT_TEMPLATE}

...      <confirm-payment>
...      <order_id>\${order_id}</order_id>
...      <payment_type>credit</payment_type>
...      <type>visa</type>
...      <card_number>4719700591590995</card_number>
...      <cvv>752</cvv>
...      <expired_month>7</expired_month>
...      <expired_year>20</expired_year>
...      <card_name>Karnwat Wongudom</card_name>
...      <total_price>\${total_price}</total_price>
...      </confirm-payment>
```

Test Cases in Test Case Files

checkout-success-keywords.robot

Hard Code!

```
*** Keywords ***
```

```
Confirm Payment
```

```
    [Arguments]    ${total_price}    ${notify_message}
```

```
    ${payment_message}=    Replace Variables    ${CONFIRM_PAYMENT_TEMPLATE}
```

```
    ${confirmPayment}=    Encode String To Bytes    ${payment_message}
```

```
    UTF-8
```

```
    ${confirmPaymentStatus}=    Post Request    ${toy_store}    /api/v1/
```

```
    confirmPayment    data=${confirmPayment}    headers=&{POST_HEADERS}
```

```
    Request Should Be Successful    ${confirmPaymentStatus}
```

```
    ${xml}    Parse Xml    ${confirmPaymentStatus.content}
```

```
    ${message}=    Get Element Text    ${xml}
```

```
    Should Be Equal As Strings    ${message}    ${notify_message}
```

Scope in Test Case Files

Robot Framework data is defined in different sections.

[learn more](#)

Environment

Suites

Cases

`${order_id}`

Tasks

Order Diner Set
Confirm Payment

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-keywords.robot'. The device has a thin grey border and a small circular camera lens at the top center.

```
$ robot checkout-success-keywords.robot
```



HTTP RequestsLibrary(Python)

*** Template ***

[lean more](#)

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-template.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-template.robot
```

Test Cases in Test Case Files

checkout-success-template.robot

Hard Code!

*** Test Cases ***

Checkout Diner Set

Get Product List	{2}		
Get Product Detail	2	43 Piece dinner Set	
Order Diner Set	2	1	14.95
Confirm Payment	14.95		

Checkout Bicycle

Get Product List	{2}		
Get Product Detail	1	Balance Training Bicycle	
Order Diner Set	1	2	241.90
Confirm Payment	241.90		

Test Cases in Test Case Files

checkout-success-template.robot

Hard Code!

*** Test Cases ***

Checkout Diner Set

Checkout Product	\${2}	2	43 Piece dinner Set	1	14.95
------------------	-------	---	---------------------	---	-------

*** Keywords ***

Checkout Product

[Arguments]	\${total}	\${product_id}	\${product_name}	\$
{quantity}	\${total_price}	\${notify_message}		

Get Product List \${total}

Find Product by Name \${product_name}

Get Product Detail \${product_name}

Order Diner Set \${quantity} \${total_price}

Confirm Payment \${total_price} \${notify_message}

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-template.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-template.robot
```

Test Cases in Test Case Files

checkout-success-template.robot

Hard Code!

```
*** Test Cases ***
```

```
Checkout Bicycle
```

```
    Checkout Product    ${2}    1    Balance Training Bicycle    2    241.90
```

```
*** Keywords ***
```

```
...
```

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-template.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-template.robot
```


Test Cases in Test Case Files

checkout-success-template.robot

Hard Code!

*** Settings ***

Library RequestsLibrary

Suite Setup Create Session \${toy_store} \${URL}

Suite Teardown Delete All Sessions

Test Template Checkout Product

*** Test Cases ***

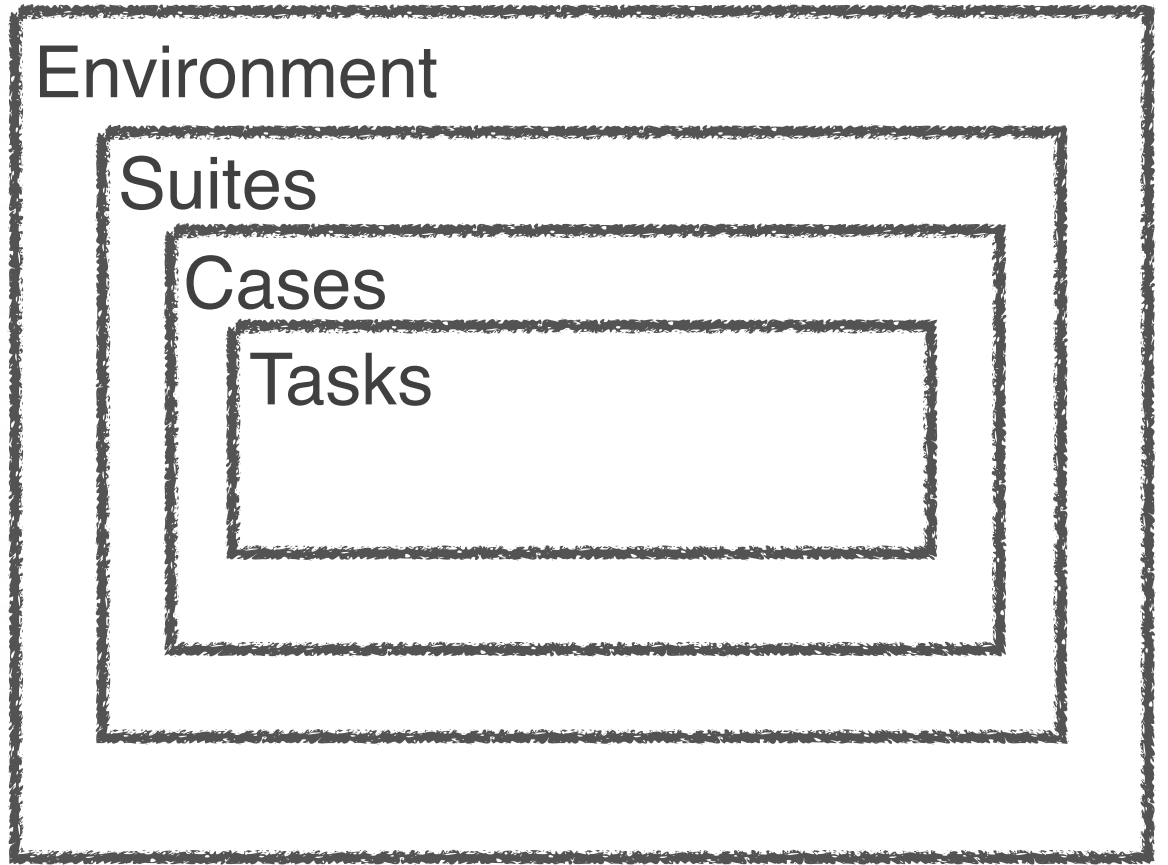
Checkout Diner Set \${2} 2 43 Piece dinner Set 1 14.95

Checkout Bicycle \${2} 1 Balance Training Bicycle 2 241.90

Scope in Test Case Files

Robot Framework data is defined in different sections.

[learn more](#)



Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The command is '\$ robot checkout-success-template.robot'. The device has a thin grey border and a small grey dot at the top center, representing a camera or sensor.

```
$ robot checkout-success-template.robot
```



HTTP RequestsLibrary(Python)

*** Resource ***

[lean more](#)

Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The device has a thin grey border and a small circular camera lens at the top center.

```
$ robot checkout-success-resources.robot
```

Test Cases in Test Case Files

checkout-success-resources.robot

*** Variables ***

`${toy_store}`

`${URL}` http://localhost:8000

`&{CONTENT_TYPE}` Content-Type=application/json

`&{ACCEPT}` Accept=application/json

`&{POST_HEADERS}` `&{ACCEPT}` `&{CONTENT_TYPE}`

...

*** Test Cases ***

Checkout Diner Set ...

*** Keywords ***

Checkout Product

[Arguments] `${total}` `${product_id}` `${product_name}` \$

`{quantity}` `${total_price}` `${notify_message}`

Get Product List `${total}`

Get Product Detail `${product_id}` `${product_name}`

Order Diner Set `${product_id}` `${quantity}` `${total_price}`

Confirm Payment `${total_price}` `${notify_message}`

...

Test Cases in Test Case Files

checkout-success-resources.robot

move

*** Variables ***

*** Keywords ***

to resources.robot

*** Settings ***

Library RequestsLibrary

Library Collections

Suite Setup Create Session \${toy_store} \${URL}

Suite Teardown Delete All Sessions

Test Template Checkout Product

Resource ./resources.robot

*** Test Cases ***

Checkout Diner Set \${2} 2 43 Piece dinner Set 1 14.95

Checkout Bicycle \${2} 1 Balance Training Bicycle 2 241.90

*** Keywords ***

Checkout Product

[Arguments] \${total} \${product_id} \${product_name} \${quantity}

\${total_price} \${notify_message}

Get Product List \${total}

Get Product Detail \${product_id} \${product_name}

Order Diner Set \${product_id} \${quantity} \${total_price}

Confirm Payment \${total_price} \${notify_message}

Test Cases in Test Case Files

resources.robot

```
*** Variables ***
${toy_store}
${URL}          http://localhost:8000
&{CONTENT_TYPE} Content-Type=application/json
&{ACCEPT}       Accept=application/json
&{POST_HEADERS} &{ACCEPT}    &{CONTENT_TYPE}
...

*** Keywords ***
Get Product List
...
Get Product Detail
...
Order Product
...
Confirm Payment
...
```


Run Test Case Files

A stylized illustration of a tablet or smartphone screen. The screen is white and displays a terminal command in a black monospaced font. The device has a thin grey border and a small circular camera lens at the top center.

```
$ robot checkout-success-resources.robot
```



thanks