

PORTFOLIO

Quality Assurance adalah jaminan yang digunakan untuk memberikan keamanan ataupun kepercayaan terhadap produk yang dibuat oleh perusahaan (<https://sarjanaekonomi.co.id/quality-assurance/>). Terdapat beberapa macam Quality Assurance, salah satunya adalah Software Quality Assurance (SQA). Software Quality Assurance adalah jaminan mutu sebuah produk yang dihasilkan oleh sebuah perusahaan teknologi berbasis perangkat lunak guna memastikan kelayakan dari sebuah produk tersebut.

Terdapat dua macam testing untuk memastikan sebuah kelayakan dari produk yang dihasilkan yaitu, Manual Testing dan Automated Testing. Manual testing adalah langkah untuk mencari bug atau cacat pada program dari perangkat lunak, pengujian dilakukan secara manual tanpa menggunakan tools yang bertujuan untuk memastikan aplikasi perangkat lunak bekerja sesuai apa yang diharapkan. Sedangkan automated testing adalah langkah untuk mencari bug atau cacat pada program dari perangkat lunak menggunakan tools ataupun script (<https://medium.com/skyshidigital/perbedaan-manual-testing-dan-automated-testing-d13373a36e>)

Contoh penerapan Manual Testing dilakukan pada Web yang terdapat Login Page dan menggunakan Microsoft Excell sebagai penulisan Test Case. Secara umum standart untuk melakukan manual testing sebagai berikut:

- | | |
|----------------------|---|
| 1. Title | : Judul dari Test Case yang akan dilakukan. |
| 2. Feature | : Fitur-fitur apa saja yang akan diuji. |
| 3. Short Description | : Deskripsi secara singkat mengenai pengujian fitur. |
| 4. Type of Test | : Memeriksa aplikasi melakukan apa yang diharapkan dengan data valid ataupun tidak valid. |
| 5. Steps | : Langkah-langkah dalam menguji kasusnya. |
| 6. Test Data | : Data input yang akan diujikan |
| 7. Expected Result | : Hasil yang seharusnya diinginkan dari skenario test. |
| 8. Actual Result | : Hasil test yang terjadi. |
| 9. Note | : Memberikan catatan untuk dilakukannya testing. |

Berikut merupakan contoh penerapan Manual Testing Login User:

<https://docs.google.com/spreadsheets/d/1Etf2hJbyAXxaNkUrFbbdy7YcOogRqhiXrpI7HKxWOMw/edit?usp=sharing>

Contoh penerapan Automated Testing Login Page akan diujikan pada <https://opensource-demo.orangehrmlive.com>. Terdapat berbagai tools untuk penerapan testing yaitu, Postman, Selenium, Katalon Studio, dan lain sebagainya. Disini saya menggunakan dua cara pada penerapan Automated Testingnya, yaitu record web dan menggunakan script.

- Automated Testing pertama: Penerapan Automated Testing disini saya menggunakan tools bernama Katalon Studio dan menggunakan record web. Adapun caranya sebagai berikut:
 1. Buka software Katalon Studio.
 2. Klik New Project lalu pilih Web Type.

Kemudian aktivitas yang telah kita lakukan akan ter-record dan masuk pada Test Case yang sudah dibuat sebelumnya.

Web Recorder

URL

Hint: Right-click on the web elements, select Katalon Studio verification actions to record [Hide](#)

RECORDED ACTIONS [Show Captured Objects >](#)

Add Recent keywords Remove Move Up ? Run

Item	Object	Input	Output
1 - Open Browser			
2 - Navigate To Url		"https://opensource-demo	
3 - Set Text	input_LOGIN Panel_txtUser	"admin"	
4 - Set Encrypted Text	input_username_txtPasswo	"hUKwJTbofgPU9eVlw/CnD	
5 - Click	input_Password_Submit		
6 - Click	a_Welcome Paul		
7 - Click	a_Logout		
8 - Close Browser			

Variables Logs

Status **PASSED**

```
click(findTestObject("Page_OrangeHRM/input_Password_Submit"))
2020-12-04 13:56:40.161 DEBUG testcase. - 6:
click(findTestObject("Page_OrangeHRM/a_Welcome Paul"))
2020-12-04 13:56:42.931 DEBUG testcase. - 7:
click(findTestObject("Page_OrangeHRM/a_Logout"))
2020-12-04 13:56:44.974 DEBUG testcase. - 8:
closeBrowser()
2020-12-04 13:56:45.202 INFO c.k.k.core.main.WSVerificationExecutor - END
Verification
```

Activate Windows
Go to Settings to activate Windows.

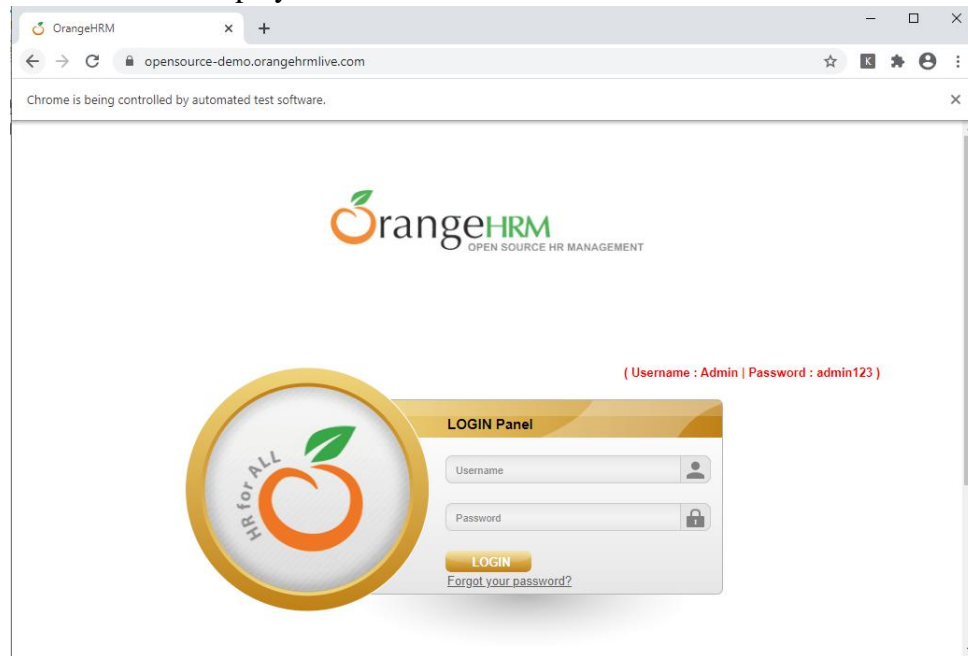
? Save Script Cancel

Katalon Help Login

Add Recent keywords Delete Move up

Item	Object
1 - Open Browser	
2 - Navigate To Url	
3 - Set Text	input_LOGIN Panel_txtUsernam
4 - Set Encrypted Text	input_username_txtPassword
5 - Click	input_Password_Submit
6 - Verify Element Visible	a_Welcome Paul
7 - Click	a_Welcome Paul
8 - Click	a_Logout
9 - Close Browser	

5. Setelah itu klik tombol save script dan akan terdapat Test Case yang telah dibuat, kemudian dilakukan play kembali.



Katalon Help Login

Job Progress

Test Cases/Login - Chrome - 20201204_135925	1/1
<Passed> - Chrome	
Test Cases/Login - Chrome - 20201204_135217	1/1
<Passed> - Chrome	
Test Cases/Login - Chrome - 20201204_134817	1/1
<Passed> - Chrome	

Item | Object

1 - Open Browser	
2 - Navigate To Url	
3 - Set Text	input_LOGIN Panel_txtUsernam
4 - Set Encrypted Text	input_username_txtPassword
5 - Click	input_Password_Submit
6 - Verify Element Visible	a_Welcome Paul
7 - Click	a_Welcome Paul
8 - Click	a_Logout
9 - Close Browser	

Manual </> Script Variables </> Variables (Script mode)

Problems Event Log Console Log Viewer Self-h

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0

Test Cases/Login (33.387s)

- 1 - openBrowser("") (9.654s)
- 2 - navigateToUrl("https://opensource-demo.orangehrmlive.com/") (8.010s)
- 3 - setText(findTestObject("Object Repository/Page_OrangeHRM/input_LOGIN Panel_txtUsername"), "Admin") (0.000s)
- 4 - setEncryptedText(findTestObject("Object Repository/Page_OrangeHRM/input_Password_Submit"), "admin123") (0.000s)
- 5 - click(findTestObject("Object Repository/Page_OrangeHRM/input_Password_Submit")) (0.000s)
- 6 - verifyElementVisible(findTestObject("Page_OrangeHRM/a_Welcome Paul")) (0.000s)
- 7 - click(findTestObject("Object Repository/Page_OrangeHRM/a_Welcome Paul")) (0.000s)
- 8 - click(findTestObject("Object Repository/Page_OrangeHRM/a_Logout")) (0.000s)
- 9 - closeBrowser() (0.896s)

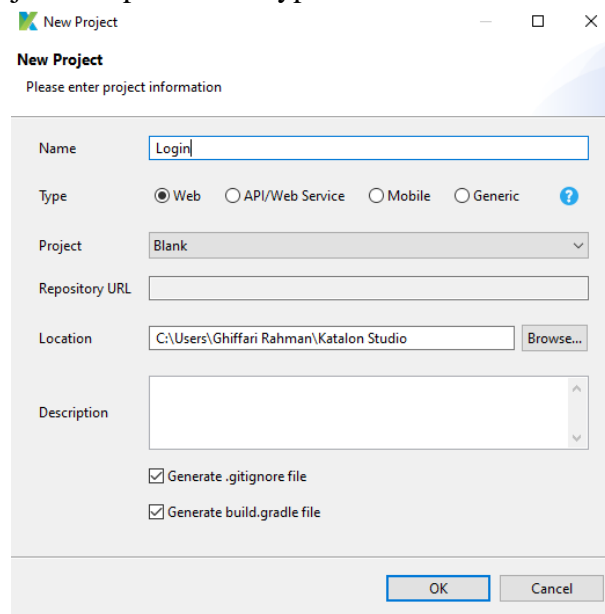
12-04-2020 01:59:47 PM Test Cases/Login

Elapsed time: 33.387s

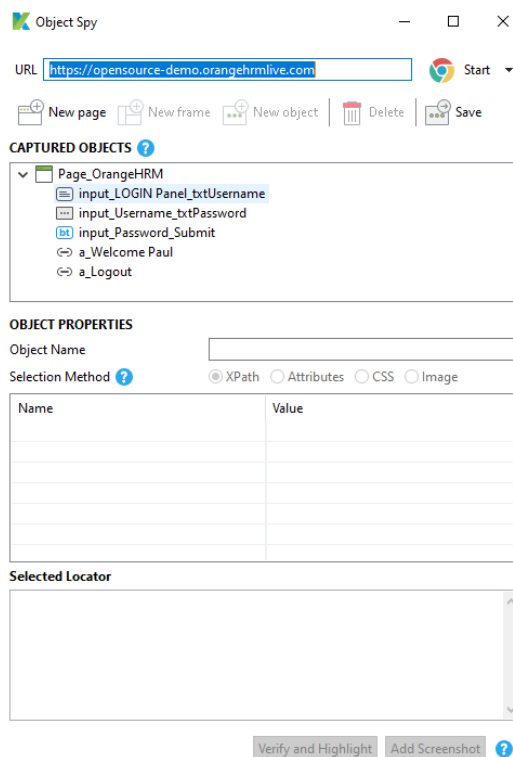
Test Cases/Login

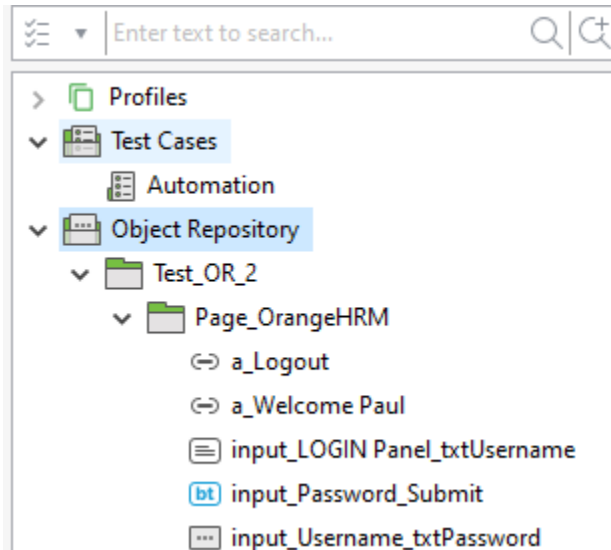
Activate Windows

- Automated Testing kedua: Penerapan Automated Testing disini saya menggunakan tools bernama Katalon Studio dan menggunakan script. Adapun caranya sebagai berikut:
 - Buka software Katalon Studio.
 - Klik New Project lalu pilih Web Type.



- Kemudian dilakukan spy web yang digunakan untuk menyimpan objek apa saja yang nantinya akan dilakukan pengujian. Objek-objek tersebut akan disimpan pada object Repository.





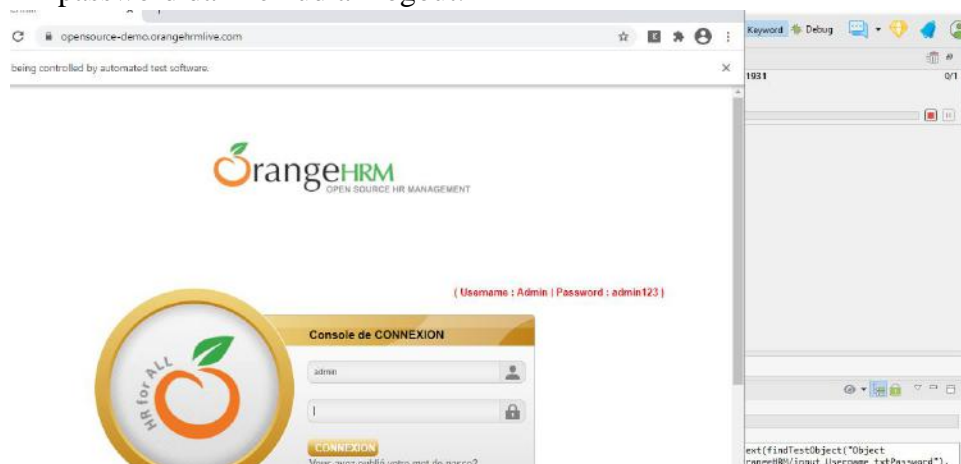
4. Kemudian dilakukan pembuatan Test Case menggunakan script.

```

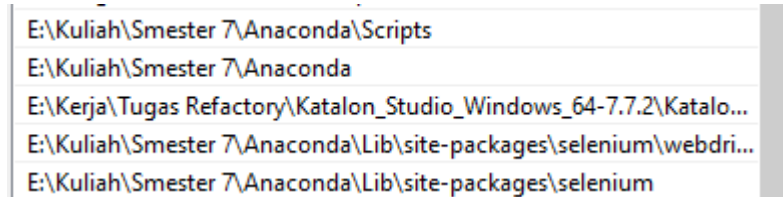
Katalon Help Automation
1 import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
17
18 //Open Browser
19 WebUI.openBrowser("https://opensource-demo.orangehrmlive.com/")
20
21 //SetText
22 WebUI.setText(findTestObject('Object Repository/Test_OR_2/Page_OrangeHRM/input_LOGIN Panel_txtUsername'), "admin")
23
24 //SetText
25 WebUI.setText(findTestObject('Object Repository/Test_OR_2/Page_OrangeHRM/input_Username_txtPassword'), "admin123")
26
27 WebUI.click(findTestObject('Object Repository/Test_OR_2/Page_OrangeHRM/input_Password_Submit'))
28
29 WebUI.click(findTestObject('Object Repository/Test_OR_2/Page_OrangeHRM/a_Welcome Paul'))
30
31 WebUI.click(findTestObject('Object Repository/Test_OR_2/Page_OrangeHRM/a_Logout'))
32

```

5. Kemudian run program yang sudah jadi dan secara otomatis program tersebut menjalankannya yang diawali dengan membuka browser, mengisi username, mengisi password dan kemudian logout.



- Automated Testing ketiga: Penerapan Automated Testing disini saya melakukan pada <http://www.facebook.com> dan menggunakan selenium webdriver dan python.
 - Pertama install python yang bisa di download di python.org. Disini saya menginstall Anaconda yang didalamnya terdapat python.
 - Kemudian tentukan envirotnmentnya seperti gambar dibawah:



- Kemudian install selenium dengan perintah pip install selenium pada python command prompt yang kemudian seleniumnya terinstall dimana python tersebut terinstall.
- Kemudian pembuatan file python yang digunakan untuk Automated Testing pada python command prompt.

```

*login.py - E:\Kuliah\Smester 7\Anaconda\Lib\site-packages\login.py (3.7.5)*
File Edit Format Run Options Window Help
from selenium import webdriver

driver = webdriver.Chrome('chromedriver.exe') #visit chrome
driver.get('https://www.facebook.com/') #visit facebook
driver.find_element_by_name('email').send_keys('Your Email') #input email
driver.find_element_by_name('pass').send_keys('Your Password') #input password
driver.find_element_by_name('login').click() #klik button login
  
```

- Secara otomatis program akan menjalankan Login Testing menggunakan selenium webdriver pada <http://www.facebook.com>

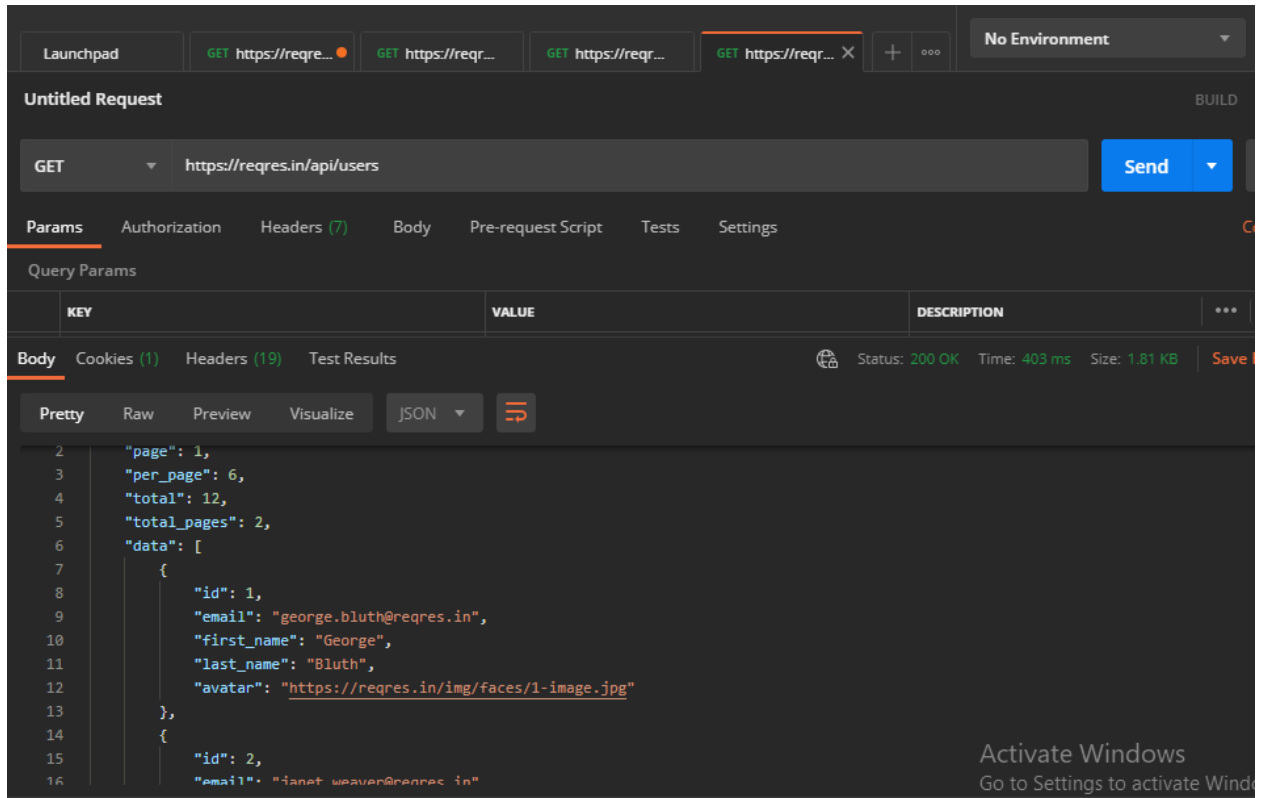
API Testing

API merupakan kepanjangan dari *Application Programming Interface*, yang memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan (<https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8>).

Pada API terdapat istilah Rest API yaitu standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Didalam Rest API terdapat Rest Client dan Rest Server dimana Rest Client mengakses dan menampilkan *resources* atau sumber data untuk penggunaan selanjutnya, sedangkan Rest Server menyediakan *resources* atau sumber data tersebut (<https://medium.com/@cecepahmadfauzi93/rest-rest-api-dan-client-server-527a15e68ff2>).

Penerapan Testing API dilakukan pada <https://reqres.in/> menggunakan beberapa method atau request dari sisi client yaitu, GET, POST, PUT, dan DELETE.

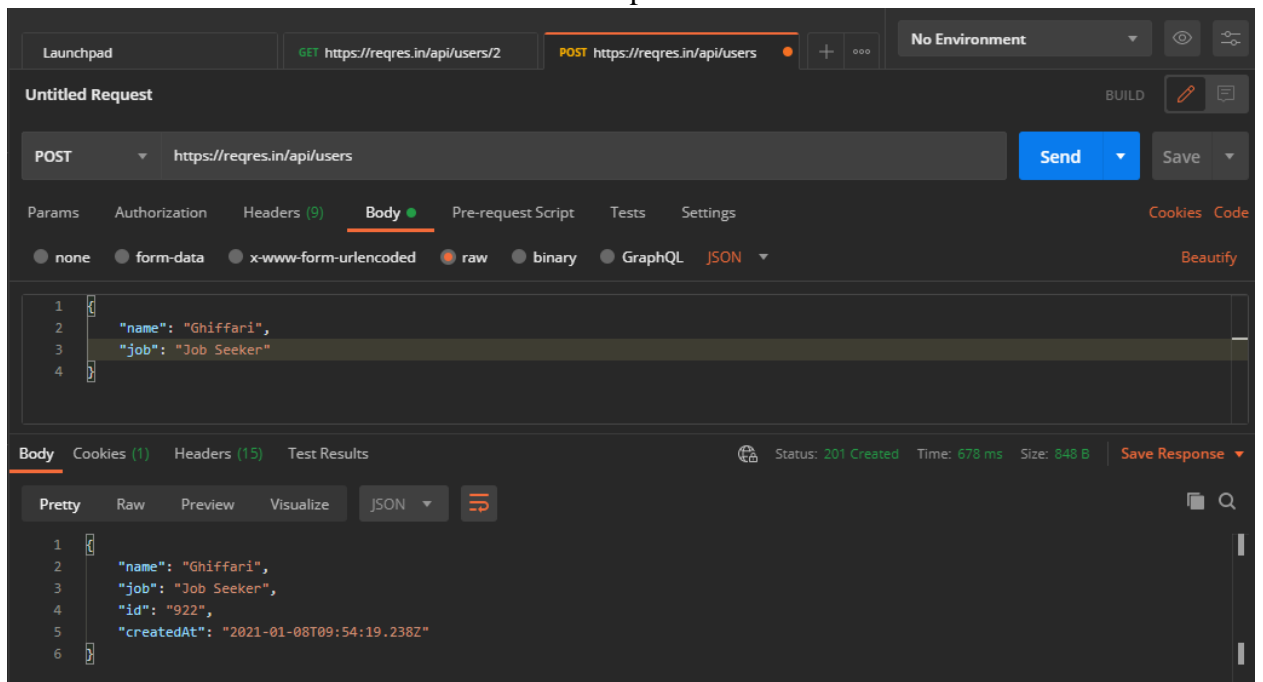
Method GET: Membaca sebuah record atau data dari REST Server.



The screenshot shows the Postman interface with a GET request to `https://reqres.in/api/users`. The response is a JSON object with the following structure:

```
2  "page": 1,
3  "per_page": 6,
4  "total": 12,
5  "total_pages": 2,
6  "data": [
7    {
8      "id": 1,
9      "email": "george.bluth@reqres.in",
10     "first_name": "George",
11     "last_name": "Bluth",
12     "avatar": "https://reqres.in/img/faces/1-image.jpg"
13   },
14   {
15     "id": 2,
16     "email": "ianat.weaver@reqres.in"
```

Method POST: Membuat record atau data baru pada REST Server.



The screenshot shows the Postman interface with a POST request to `https://reqres.in/api/users`. The request body is a JSON object with the following structure:

```
1  {
2    "name": "Ghiffari",
3    "job": "Job Seeker"
4  }
```

The response is a JSON object with the following structure:

```
1  {
2    "name": "Ghiffari",
3    "job": "Job Seeker",
4    "id": "922",
5    "createdAt": "2021-01-08T09:54:19.238Z"
6  }
```


Method PUT: Mengupdate sebuah record atau data pada REST Server.

The screenshot shows a REST client interface with a PUT request to `https://reqres.in/api/users/1`. The request body is a JSON object: `{ "name": "Ghiffari", "job": "Job Seeker" }`. The response status is `200 OK` with a time of `630 ms` and a size of `879 B`. The response body is a JSON object: `{ "name": "Ghiffari", "job": "Job Seeker", "updatedAt": "2021-01-08T09:57:44.887Z" }`.

Method DELETE: Menghapus sebuah record atau data pada REST Server.

The screenshot shows a REST client interface with a DELETE request to `https://reqres.in/api/users/1`. The response status is `204 No Content` with a time of `625 ms` and a size of `710 B`. The response body is empty.