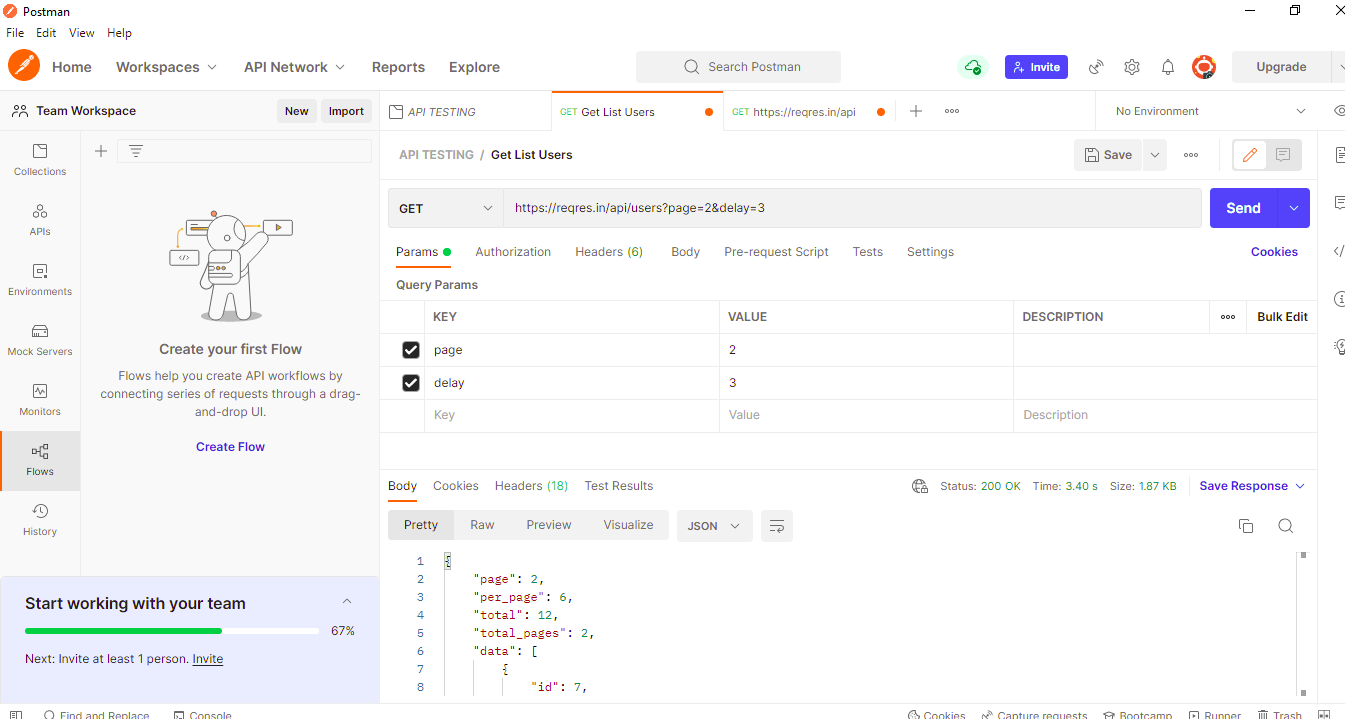
# Tuas HTTP Method GET



Dari perintah tugas yang di sampaikan berdasarka slide tersebut, setidaknya disini saya akan membuat beberapa kali pecobaan disini untuk Untuk melihat perbedaan akan hasil finalnya.

Yang pertama adalah dengan percobaan asli yang dicontohkan pada praktek materi :

1. Pada Postman, dengan menetapkan
   * **Base URL:** <https://reqres.in/api>
   * **Endpoint:** users
   * **Request Parameters:** ?pages=2&per\_pages=1&delay=3
   * **Method:** GET
   * Hasil setelah permintaan request



* **Isi lengkap dari Body yang didapat**

{

    "page": 2,

    "per\_page": 6,

    "total": 12,

    "total\_pages": 2,

    "data": [

        {

            "id": 7,

            "email": "michael.lawson@reqres.in",

            "first\_name": "Michael",

            "last\_name": "Lawson",

            "avatar": "https://reqres.in/img/faces/7-image.jpg"

        },

        {

            "id": 8,

            "email": "lindsay.ferguson@reqres.in",

            "first\_name": "Lindsay",

            "last\_name": "Ferguson",

            "avatar": "https://reqres.in/img/faces/8-image.jpg"

        },

        {

            "id": 9,

            "email": "tobias.funke@reqres.in",

            "first\_name": "Tobias",

            "last\_name": "Funke",

            "avatar": "https://reqres.in/img/faces/9-image.jpg"

        },

        {

            "id": 10,

            "email": "byron.fields@reqres.in",

            "first\_name": "Byron",

            "last\_name": "Fields",

            "avatar": "https://reqres.in/img/faces/10-image.jpg"

        },

        {

            "id": 11,

            "email": "george.edwards@reqres.in",

            "first\_name": "George",

            "last\_name": "Edwards",

            "avatar": "https://reqres.in/img/faces/11-image.jpg"

        },

        {

            "id": 12,

            "email": "rachel.howell@reqres.in",

            "first\_name": "Rachel",

            "last\_name": "Howell",

            "avatar": "https://reqres.in/img/faces/12-image.jpg"

        }

    ],

    "support": {

        "url": "https://reqres.in/#support-heading",

        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"

    }

}

1. Percobaan Pada Cypress, dengan Parameter lengap atau sama dengan postman diatas

* Membuat file script baru dengan isi berikut ini:

/// <reference types="cypress" />

/// <reference types="cypress" />

describe ('GET USERS LIST', () => {

    it('Verify Users list will displayed', () => {

        cy.request({

            method:'GET',

            url: 'https://reqres.in/api/users',

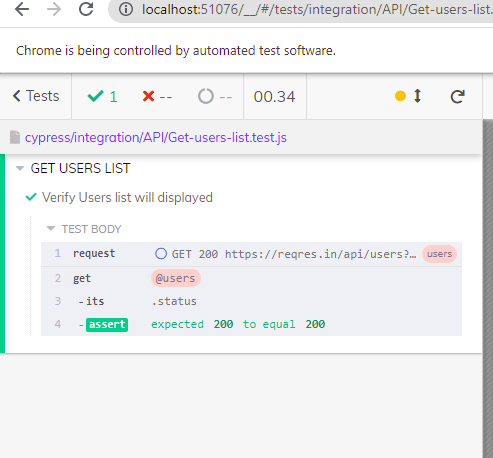
        }).as('users')

        cy.get('@users').its('status').should('equal',200);

    });

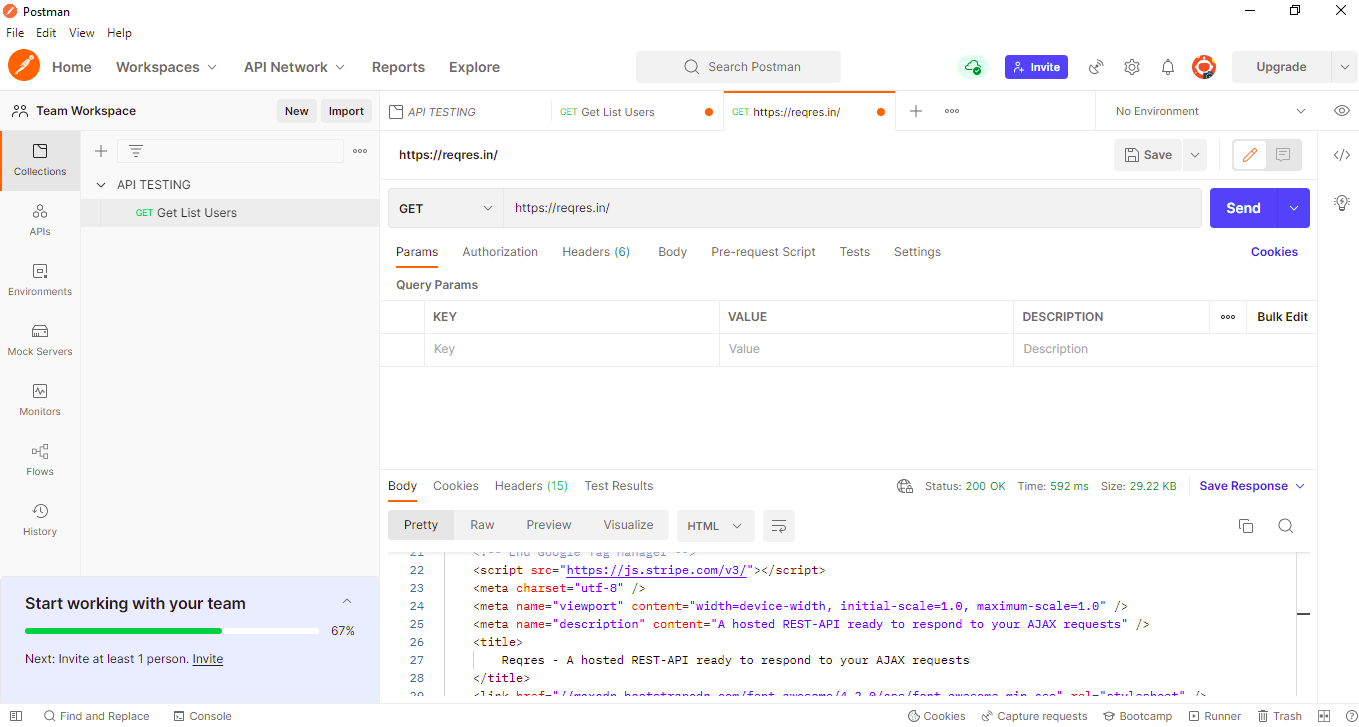
})

* Menjalankan Run Script dan seperti berikutlah hasilnya :



Kemudian yang percobaan kedua ini adalah inti dari percobaan yang sesuai diprintahkan tugas pada slide, yaitu permintaan dengan HTTP tanpa parameter :

1. Pada Postman, dengan menetapkan URL tanpa Parameter
   * **Base URL:** [https://reqres.in/](https://reqres.in/i)
   * **Tampilan Hasil** setelah request



* **Isi lengkap dari Body yang didapa**

<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Google Tag Manager -->

    <script>

        ;(**function** (w, d, s, l, i) {

                w[l] **=** w[l] **||** []

                w[l].**push**({

                    'gtm.start': **new** Date().**getTime**(),

                    event: 'gtm.js',

                })

                var f **=** d.**getElementsByTagName**(s)[0],

                    j **=** d.**createElement**(s),

                    dl **=** l **!=** 'dataLayer' **?** '&l=' **+** l **:** ''

                j.async **=** **true**

                j.src **=** 'https://www.googletagmanager.com/gtm.js?id=' **+** i **+** dl

                f.parentNode.**insertBefore**(j, f)

            })(window, document, 'script', 'dataLayer', 'GTM-KB622KF')

    </script>

    <!-- End Google Tag Manager -->

    <script src="https://js.stripe.com/v3/"></script>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0" />

    <meta name="description" content="A hosted REST-API ready to respond to your AJAX requests" />

    <title>

        Reqres - A hosted REST-API ready to respond to your AJAX requests

    </title>

    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css" rel="stylesheet" />

    <link rel="stylesheet" href="/css/app.css" />

    <script type="text/javascript" src="//m.servedby-buysellads.com/monetization.custom.js"></script>

</head>

<body>

    <div class="header-content">

        <header class="main-header">

            <section class="wrap">

                <h1 class="logo">

                    <a href="/"><img src="/img/logo.png" alt="" /></a>

                </h1>

            </section>

        </header>

        <main class="the-sell">

            <div class="wrap">

                <h2 class="tagline">

                    Test your front-end against a real API

                </h2>

                <section class="key-sells">

                    <div class="sell">

                        <div class="v-center">

                            <h3>Fake data</h3>

                            <p>

                                No more tedious sample data creation, we've

                                got it covered.

                            </p>

                        </div>

                        <i class="fa fa-cloud-download"></i>

                    </div>

                    <div class="sell">

                        <div class="v-center">

                            <h3>Real responses</h3>

                            <p>

                                Develop with real response codes. GET, POST,

                                PUT &amp; DELETE supported.

                            </p>

                        </div>

                        <i class="fa fa-code"></i>

                    </div>

                    <div class="sell">

                        <div class="v-center">

                            <h3>Always-on</h3>

                            <p>

                                24/7 <strong><em>free</em></strong> access

                                in your development phases. Go nuts.

                            </p>

                        </div>

                        <i class="fa fa-toggle-on"></i>

                    </div>

                </section>

                <h2 class="tagline">

                    A hosted REST-API ready to respond to your AJAX

                    requests.

                </h2>

                <div class="t-center main-arrow">

                    <i class="fa fa-arrow-circle-down"></i>

                </div>

                <script async type="text/javascript"

                    src="//cdn.carbonads.com/carbon.js?serve=CE7D6K3E&placement=reqresin" id="\_carbonads\_js"></script>

            </div>

        </main>

    </div>

    <div class="container">

        <div class="home-content">

            <div class="wrap">

                <h2 class="t-center heading">Give it a try</h2>

                <div class="t-center">

                    <button style="float: none">

                <a style="text-decoration: none" href="#support-heading"

                    >Support ReqRes</a

                >

            </button>

                </div>

                <hr class="dark" />

                <section class="console try-api-links" id="console">

                    <div class="endpoints" data-key="endpoints">

                        <ul>

                            <li data-id="users" data-key="endpoint" data-http="get" class="active">

                                <a data-key="try-link"> List users </a>

                            </li>

                            <li data-id="users-single" data-key="endpoint" data-http="get">

                                <a data-key="try-link"> Single user </a>

                            </li>

                            <li data-id="users-single-not-found" data-key="endpoint" data-http="get">

                                <a data-key="try-link"> Single user not found </a>

                            </li>

                            <li data-id="unknown" data-key="endpoint" data-http="get">

                                <a data-key="try-link"> List &lt;resource&gt; </a>

                            </li>

                            <li data-id="unknown-single" data-key="endpoint" data-http="get">

                                <a data-key="try-link"> Single &lt;resource&gt; </a>

                            </li>

                            <li data-id="unknown-single-not-found" data-key="endpoint" data-http="get">

                                <a data-key="try-link">

                                    Single &lt;resource&gt; not found

                                </a>

                            </li>

                            <li data-id="post" data-key="endpoint" data-http="post">

                                <a data-key="try-link"> Create </a>

                            </li>

                            <li data-id="put" data-key="endpoint" data-http="put">

                                <a data-key="try-link"> Update </a>

                            </li>

                            <li data-id="patch" data-key="endpoint" data-http="patch">

                                <a data-key="try-link"> Update </a>

                            </li>

                            <li data-id="delete" data-key="endpoint" data-http="delete">

                                <a data-key="try-link"> Delete </a>

                            </li>

                            <li data-id="register-successful" data-key="endpoint" data-http="post">

                                <a data-key="try-link"> Register - successful </a>

                            </li>

                            <li data-id="register-unsuccessful" data-key="endpoint" data-http="post">

                                <a data-key="try-link"> Register - unsuccessful </a>

                            </li>

                            <li data-id="login-successful" data-key="endpoint" data-http="post">

                                <a data-key="try-link"> Login - successful </a>

                            </li>

                            <li data-id="login-unsuccessful" data-key="endpoint" data-http="post">

                                <a data-key="try-link"> Login - unsuccessful </a>

                            </li>

                            <li data-id="delay" data-key="endpoint" data-http="get">

                                <a data-key="try-link"> Delayed response </a>

                            </li>

                        </ul>

                    </div>

                    <!--

            -->

                    <div class="output">

                        <div class="request">

                            <p class="request-title">

                                <strong

                            >Request <i class="fa fa-arrow-circle-up"></i>

                            <a

                                class="link try-link"

                                data-key="request-output-link"

                            >

                                <span class="url" data-key="url"></span>

                            </a>

                        </strong>

                            </p>

                            <pre data-key="output-request"></pre>

                        </div>

                        <!--

                -->

                        <div class="response">

                            <p class="response-title">

                                <strong

                            >Response <i class="fa fa-arrow-circle-down"></i>

                            <span

                                class="response-code"

                                data-key="response-code"

                            ></span

                        ></strong>

                            </p>

                            <div data-key="spinner" class="spinner" hidden>

                                <div class="cube1"></div>

                                <div class="cube2"></div>

                            </div>

                            <pre data-key="output-response"></pre>

                        </div>

                    </div>

                </section>

                <hr class="dark" />

                <style>

                    #impeccable-js {

                        font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI',

                            Roboto, Oxygen-Sans, Ubuntu, Cantarell, 'Helvetica Neue',

                            Helvetica, Arial, sans-serif;

                        margin-bottom: 2rem;

                    }

                    #impeccable-js \* {

                        margin: initial;

                        padding: initial;

                    }

                    #impeccable-js a {

                        display: flex;

                        justify-content: space-between;

                        align-items: center;

                        z-index: 100;

                        overflow: hidden;

                        max-width: 1200px;

                        border-radius: 3px;

                        background-color: #161616;

                        box-shadow: 0 1px 4px 1px hsla(0, 0%, 0%, 0.1);

                    }

                    #impeccable-js :any-link {

                        text-decoration: none;

                    }

                    #impeccable-js img {

                        padding: 10px;

                        box-sizing: content-box;

                        width: 125px;

                        height: 50px;

                    }

                    #impeccable-js .sponsor-description {

                        margin: 0 1em;

                        font-size: 14px;

                        line-height: 1.4;

                        color: #fff;

                        text-align: left;

                        flex: 1;

                    }

                    #impeccable-js .sponsor-cta {

                        margin-right: 14px;

                        padding: 1em;

                        font-size: 11px;

                        font-weight: 500;

                        line-height: 1;

                        white-space: nowrap;

                        border-radius: 4px;

                    }

                </style>

                <div id="impeccable-js"></div>

                <div>

                    <iframe

                        src="https://codesandbox.io/embed/polished-butterfly-j17lt?autoresize=1&fontsize=14&theme=dark&view=preview"

                        style="

                    width: 100%;

                    height: 700px;

                    border: 0;

                    border-radius: 4px;

                    overflow: hidden;

                " title="polished-butterfly-j17lt"

                        allow="accelerometer; ambient-light-sensor; camera; encrypted-media; geolocation; gyroscope; hid; microphone; midi; payment; usb; vr; xr-spatial-tracking"

                        sandbox="allow-forms allow-modals allow-popups allow-presentation allow-same-origin allow-scripts"></iframe>

                </div>

                <h2 class="t-center heading">It&rsquo;s all in the details</h2>

                <hr class="dark" />

                <section class="technical-bits" id="technical-bits">

                    <ul class="iconed">

                        <li>

                            <i class="fa fa-cloud"></i>

                            <p>

                                <strong

                            >Hosted on

                            <a href="https://www.heroku.com/">Heroku</a></strong>

                            </p>

                            <p>

                                Which means 99.99% Uptime SLA.<br />All you need is the

                        base URL, and you're away: <br /><br /><span

                            class="code"

                            >https://reqres.in/api/</span><br /><br />The API is CORS enabled, so you can make

                        requests right from the browser, no matter what domain,

                        or even from somewhere like JSFiddle or JSBin.

                    </p>

                        </li>

                        <li>

                            <i class="fa fa-file-code-o"></i>

                            <p><strong>Language agnostic</strong></p>

                            <p>

                                A generic API that conforms to REST principles and

                                accepts a content type of <br /><br /><span class="code"

                            >application/json</span><br /><br />Any endpoint that contains

                        "&lt;resource&gt;" can be substituted with anything you

                        supply, ie. "products", "accounts", etc..the API will

                        just respond with various Pantone colours.

                    </p>

                        </li>

                    </ul>

                </section>

                <h2 class="t-center heading">Getting started</h2>

                <hr class="dark" />

                <section class="getting-started t-center" id="getting-started">

                    <i class="fa fa-html5"></i>

                    <div class="tab">

                        <p class="tab-title">jQuery</p>

                        <p>If you, for example, want to create a fake user:</p>

                        <pre>$.ajax({

    url: "https://reqres.in/api/users",

    type: "POST",

    data: {

        name: "paul rudd",

        movies: ["I Love You Man", "Role Models"]

    },

    success: function(response){

        console.log(response);

    }

});

                </pre>

                        <p>For which the response to this request will be...</p>

                        <pre>{

    "name":"paul rudd",

    "movies[]":[

        "I Love You Man",

        "Role Models"

    ],

    "id":"243",

    "createdAt":"2014-10-18T12:09:05.255Z"

}

                </pre>

                        <p>

                            You can see that the API has sent us back whatever user

                            details we sent it, plus an

                            <span class="code">id</span> &amp;

                            <span class="code">createdAt</span> key for our use.

                        </p>

                    </div>

                    <!--

            -->

                    <div class="tab">

                        <p class="tab-title">Native JavaScript</p>

                        <p>

                            If you've already got your own application entities, ie.

                            "products", you can send them in the endpoint URL, like so:

                        </p>

                        <pre>var xhr = new XMLHttpRequest();

xhr.open("GET", "https://reqres.in/api/products/3", true);

xhr.onload = function(){

    console.log(xhr.responseText);

};

xhr.send();

                </pre>

                        <p>

                            It would be impossible for Reqres to know your application

                            data, so the API will respond from a sample set of Pantone

                            colour data

                        </p>

                        <pre>{

    "data":{

        "id":3,

        "name":"true red",

        "year":2002,

        "pantone\_value":"19-1664"

    }

}

                </pre>

                        <p>

                            It's entirely possible to get sample data into your

                            interface in seconds!

                        </p>

                    </div>

                </section>

                <h2 class="t-center heading">Still don't really get it...</h2>

                <hr class="dark" />

                <section class="technical-bits" id="technical-bits">

                    <ul class="iconed">

                        <li>

                            <i class="fa fa-rocket"></i>

                            <p>

                                <strong>Reqres is a <em>real</em> API</strong>

                            </p>

                            <p>

                                Reqres simulates real application scenarios. If you want

                                to test a user authentication system, Reqres will

                                respond to a successful login/register request with a

                                token for you to identify a sample user, or with a 403

                                forbidden response to an unsuccessful login/registration

                                attempt. <br /><br />A common front-end scenario that's

                        easily forgotten is loading states, which can be easily

                        simulated through Reqres by appending <br /><span

                            class="code"

                            >?delay=&lt;a number of seconds&gt;</span>

                                to any endpoint URL, which will delay the API response

                                time. Animated loading GIFs &amp; SVGs at the ready!

                            </p>

                        </li>

                        <li>

                            <i class="fa fa-bolt"></i>

                            <p><strong>Technical demos and tutorials</strong></p>

                            <p>

                                If you're trying to demonstrate a front-end

                                (JavaScript-based) concept, you don't really want the

                                hassle of setting up an API, or even a server

                                (especially during a live workshop or demo).<br /><br />

                        You can just write your HTML, CSS &amp; JavaScript as

                        usual and send Reqres AJAX requests, which will respond

                        with the expected response codes and output.

                    </p>

                        </li>

                        <li>

                            <i class="fa fa-file-code-o"></i>

                            <p><strong>Rapid prototyping of interfaces</strong></p>

                            <p>

                                When prototyping a new interface, you just want an API

                                <em>there</em>, with minimal setup effort involved.

                                Normally, I'd point people, who aren't too familiar with

                                backend programming, to

                                <a href="http://sailsjs.org">Sailsjs</a> which can

                                auto-generate a REST-API for you from the command

                                line.<br /><br />

                        However, you will need Node.js installed and some

                        familiarity of how Node.js works. If that sounds like

                        too much hassle and way too daunting, Reqres is just a

                        URL. Sending it an AJAX request is step 1...there is no

                        step 2.

                    </p>

                        </li>

                    </ul>

                </section>

                <h2 class="t-center heading">Peace of mind</h2>

                <hr class="dark" />

                <section class="peace-of-mind" id="peace-of-mind">

                    <i class="fa fa-lock"></i>

                    <p>

                        It might seem <strong>pretty weird</strong> to be sending your

                        data to a 3rd party API, but I can assure you, Reqres

                        <strong>does not store any of your data</strong> at all. Once

                        you send it to us, we just send it straight back...and then it's

                        gone!

                    </p>

                </section>

                <h2 id="support-heading" class="t-center heading">Support</h2>

                <hr class="dark" />

                <section class="pro support" id="support">

                    <p class="breathe">

                        ReqRes serves nearly

                        <strong>half a billion requests each month</strong>, and is

                        provided free-of-charge.

                    </p>

                    <p class="breathe">

                        To keep ReqRes free, contributions towards running costs are

                        appreciated!

                    </p>

                    <form class="breathe" action="#" id="supportForm">

                        <div>

                            <input

                        type="radio"

                        id="supportOneTime"

                        name="support"

                        value="supportOneTime"

                        checked

                    />

                            <input

                        type="number"

                        placeholder="$10"

                        name="oneTimeAmount"

                    />

                            <label for="supportOneTime">One-time payment ($)</label>

                        </div>

                        <div>

                            <input

                        type="radio"

                        id="supportRecurring"

                        name="support"

                        value="supportRecurring"

                    />

                            <label for="supportRecurring"

                        >Monthly support ($5/month)</label>

                        </div>

                        <button>Support ReqRes</button>

                    </form>

                    <p class="tiny">

                        Selecting either option will redirect you to a hosted Stripe

                        Checkout page to complete payment.

                    </p>

                </section>

                <h2 class="t-center heading">Advertising</h2>

                <hr class="dark" />

                <section class="pro" id="pro">

                    <p>

                        Do you want to advertise your jobs/tools/software/cats through

                        Reqres to millions of developers every week? Click for more

                        <a href="https://www.dropbox.com/s/2w7gaeydx02qs40/Reqres\_Advertising\_Proposal.pdf?dl=0">info

                            [pdf]</a>

                        on the ads. <a

                            href="/cdn-cgi/l/email-protection#9af2fff6f6f5daf8fff4f2f5edfef6ffb4f3f7">Contact</a> for

                        pricing and questions.

                    </p>

                </section>

                <h2 class="t-center heading">Looking for pro features?</h2>

                <hr class="dark" />

                <section class="pro" id="pro">

                    <button id="trigger-pro" class="large">Upgrade</button>

                    <div id="pro-form" class="pro-form">

                        <h3>

                            <strong>Ace.</strong> Pro features coming real soon. Stick

                            your email in here to be notified and help build our pro

                            tier.

                        </h3>

                        <!-- Begin Mailchimp Signup Form -->

                        <link href="//cdn-images.mailchimp.com/embedcode/classic-10\_7.css" rel="stylesheet"

                            type="text/css" />

                        <div id="mc\_embed\_signup">

                            <form

                                action="https://benhowdle.us20.list-manage.com/subscribe/post?u=8c58165749318545a89cd1b21&amp;id=544f278469"

                                method="post" id="mc-embedded-subscribe-form" name="mc-embedded-subscribe-form"

                                class="validate" target="\_blank" novalidate>

                                <div id="mc\_embed\_signup\_scroll">

                                    <div class="mc-field-group">

                                        <label for="mce-EMAIL">Email Address </label>

                                        <input

                                    type="email"

                                    value=""

                                    name="EMAIL"

                                    class="required email"

                                    id="mce-EMAIL"

                                />

                            </div>

                                        <div id="mce-responses" class="clear">

                                            <div class="response" id="mce-error-response" style="display: none"></div>

                                            <div class="response" id="mce-success-response" style="display: none"></div>

                                        </div>

                                        <!-- real people should not fill this in and expect good things - do not remove this or risk form bot signups-->

                                        <div style="position: absolute; left: -5000px" aria-hidden="true">

                                            <input

                                    type="text"

                                    name="b\_8c58165749318545a89cd1b21\_544f278469"

                                    tabindex="-1"

                                    value=""

                                />

                            </div>

                                            <div class="clear">

                                                <input

                                    type="submit"

                                    value="Subscribe"

                                    name="subscribe"

                                    id="mc-embedded-subscribe"

                                    class="button"

                                />

                            </div>

                                            </div>

                            </form>

                        </div>

                        <!--End mc\_embed\_signup-->

                    </div>

                </section>

            </div>

        </div>

    </div>

    <footer>

        <div class="wrap">

            <hr class="dark" />

            <p class="t-center">

                Made for developers and designers by

                <a href="https://benhowdle.im">Ben Howdle</a>

            </p>

        </div>

    </footer>

    <script data-cfasync="false" src="/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js"></script>

    <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/fastclick/1.0.3/fastclick.min.js">

    </script>

    <script src="/js/app.js"></script>

    <!-- Global site tag (gtag.js) - Google Analytics -->

    <script async src="https://www.googletagmanager.com/gtag/js?id=UA-174008107-1"></script>

    <script>

        window.dataLayer **=** window.dataLayer **||** []

**function** gtag() {

                dataLayer.**push**(**arguments**)

            }

            gtag('js', **new** Date())

            gtag('config', 'UA-174008107-1')

    </script>

    <script>

        ;(**function** (i, s, o, g, r, a, m) {

                i['GoogleAnalyticsObject'] **=** r

                ;(i[r] **=**

                    i[r] **||**

**function** () {

                        ;(i[r].q **=** i[r].q **||** []).**push**(**arguments**)

                    }),

                    (i[r].l **=** 1 **\*** **new** Date())

                ;(a **=** s.**createElement**(o)), (m **=** s.**getElementsByTagName**(o)[0])

                a.async **=** 1

                a.src **=** g

                m.parentNode.**insertBefore**(a, m)

            })(

                window,

                document,

                'script',

                '//www.google-analytics.com/analytics.js',

                'ga'

            )

            ga('create', 'UA-55888877-1', 'auto')

            ga('send', 'pageview')

    </script>

    <script>

        ;(**function** () {

**if** (**typeof** \_bsa **!==** 'undefined' **&&** \_bsa) {

                    \_bsa.init(

                        'custom',

                        'CEBIP53J',

                        'placement:reqresin-sponsor',

                        {

                            target: '#impeccable-js',

                            template: `

        <a href="##link##" rel="sponsored noopener" target="\_blank" title="##company## — ##tagline##">

        <img src="##logo##" style="background-color: ##backgroundColor##">

        <div class="sponsor-description">##description##</div>

        <div class="sponsor-cta" style="background-color: ##backgroundColor##; color: ##textColor##">##callToAction##</div>

        </a>

        `,

                        }

                    )

                }

            })()

    </script>

</body>

</html>

1. Percobaan pada cypress, dengan URL tanpa parameter
   * Dengan melakukan sedikit perubahan kode sebagai berikut:

/// <reference types="cypress" />

describe ('GET USERS LIST', () => {

    it('Verify Users list will displayed', () => {

        cy.request({

            method:'GET',

            url: 'https://reqres.in/api/user',

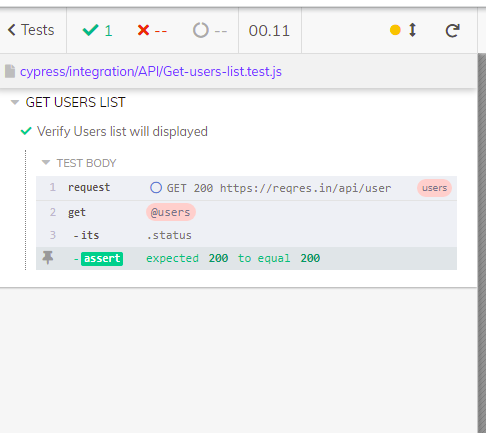
        }).as('users')

        cy.get('@users').its('status').should('equal',200);

    });

})

* Maka inilah hasil pada Test run di cypress



**Kesimpulan :**  Setelah melakukan percobaan percobaan tersebut dapat saya simpulkan, kalau pengujian API kita tidak bisa melihat tampilan E2E interfacenya, dan perbedaan antara pengujian dengan atau tanpa parameter adalah jika menggunakan parameter hasil yang didapat akan lebih tersasar atau lebih spesifik yang didapat karena sesuai dengan Namanya, yaitu parameter adalah sebuah penanda yang lebih spesifik.