**Skenario Getting All Notes**

Kita mulai skenario ke 2. Mari ingat-ingat kembali kriteria pengujiannya.

Skenario 2: Getting All Notes (Mendapatkan seluruh catatan)

Pastikan response memiliki status code 200.

Pastikan header response Content-Type memiliki nilai application/json.

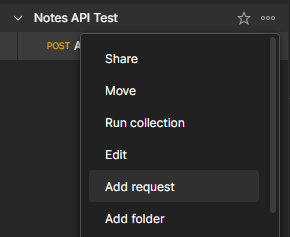
Pastikan body response adalah object.

Pastikan body response memiliki properti dan nilai atau tipe data yang sesuai.

Pastikan data pada response body memiliki array notes dan terdapat minimal 1 item di dalamnya.

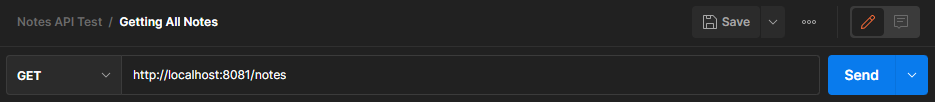
Sudah siap? Yuk buat request baru pada collection Notes API Test.

Klik kanan pada Notes API Test collection dan pilih Add Request.



Kemudian beri nama request tersebut dengan “Getting All Notes”.

Isi request URL dengan localhost:5000/notes dan biarkan method tetap GET.



Kita tidak perlu melampirkan data apa pun pada request ini. Jadi, mari langsung tuliskan kode pengujian pada tab Test.

202103090914016977ff68e286334e7ddedb79e2a4262d.png

Yuk mulai tuliskan testing pertama untuk menguji apakah response code bernilai 200.

Buat testing baru dengan nama “response status code should have 200 value”.

pm.test('response status code should have 200 value', () => {

});

Di dalamnya, lakukan uji nilai status code dengan nilai 200.

pm.test('response status code should have 200 value', () => {

pm.response.to.have.status(200);

});

Done!

Skenario 2: Getting All Notes (Mendapatkan seluruh catatan)

~~Pastikan response memiliki status code 200.~~

Pastikan header response Content-Type memiliki nilai application/json.

Pastikan body response adalah object.

Pastikan body response memiliki properti dan nilai atau tipe data yang sesuai.

Pastikan data pada response body memiliki array notes dan terdapat minimal 1 item di dalamnya.

Lanjut, buat testing baru dengan nama ‘response Content-Type header should have application/json value’.

pm.test('response Content-Type header should have application/json value', () => {

});

Di dalamnya, uji nilai response header Content-Type dengan nilai “application/json; charset=utf-8”. Anda sudah tahu bagaimana caranya kan?

pm.test('response Content-Type header should have application/json value', () => {

pm.expect(pm.response.headers.get('Content-Type')).to.equals('application/json; charset=utf-8');

});

Done!

Skenario 2: Getting All Notes (Mendapatkan seluruh catatan)

~~Pastikan response memiliki status code 200.~~

~~Pastikan header response Content-Type memiliki nilai application/json.~~

Pastikan body response adalah object.

Pastikan body response memiliki properti dan nilai atau tipe data yang sesuai.

Pastikan data pada response body memiliki array notes dan terdapat minimal 1 item di dalamnya.

Lanjut, buat testing baru dengan nama ‘response body should an object’.

pm.test('response body should an object', () => {

});

Di dalamnya kita uji body respons haruslah sebuah object.

Pertama, kita dapatkan dulu nilai body respons dalam bentuk JSON.

pm.test('response body should an object', () => {

const responseJson = pm.response.json();

});

Kemudian uji responseJson, ia harus merupakan object.

pm.test('response body should an object', () => {

const responseJson = pm.response.json();

pm.expect(responseJson).to.be.an('object');

});

Done!

Skenario 2: Getting All Notes (Mendapatkan seluruh catatan)

~~Pastikan response memiliki status code 200.~~

~~Pastikan header response Content-Type memiliki nilai application/json.~~

~~Pastikan body response adalah object.~~

Pastikan body response memiliki properti dan nilai atau tipe data yang sesuai.

Pastikan data pada response body memiliki array notes dan terdapat minimal 1 item di dalamnya.

Lanjut. Silakan buat testing baru bernama ‘response body should have the correct property and value’.

pm.test('response body should have the correct property and value', () => {

});

Di dalamnya, uji nilai response body yang merupakan object bahwa ia harus memiliki properti dan nilai yang sesuai. Untuk melakukannya, tulis kode berikut:

pm.test('response body should have the correct property and value', () => {

const responseJson = pm.response.json();

pm.expect(responseJson).to.have.ownProperty('status');

pm.expect(responseJson.status).to.equals('success');

pm.expect(responseJson).to.have.ownProperty('data');

pm.expect(responseJson.data).to.be.an('object');

});

Done!

Skenario 2: Getting All Notes (Mendapatkan seluruh catatan)

~~Pastikan response memiliki status code 200.~~

~~Pastikan header response Content-Type memiliki nilai application/json.~~

~~Pastikan body response adalah object.~~

Pastikan body response memiliki properti dan nilai atau tipe data yang sesuai.

Pastikan data pada response body memiliki array notes dan terdapat minimal 1 item di dalamnya.

Lanjut ke pengujian terakhir, kita akan uji objek data yang ada di dalam response body. Objek data harus memiliki array notes yang setidaknya memiliki 1 item note di dalamnya.

Buat test dengan nama ‘response body data should have a notes array and contain at least 1 item’.

pm.test('response body data should have a notes array and contain at least 1 item', () => {

});

Di dalamnya, dapatkan objek data dari response body dalam bentuk JSON.

pm.test('response body data should have a notes array and contain at least 1 item', () => {

const responseJson = pm.response.json();

const { data } = responseJson;

});

Kemudian uji nilai objek data bahwa ia harus memiliki properti notes yang merupakan array dan minimal memiliki 1 item di dalamnya.

pm.test('response body data should have a notes array and contain at least 1 item', () => {

const responseJson = pm.response.json();

const { data } = responseJson;

pm.expect(data).to.have.ownProperty('notes');

pm.expect(data.notes).to.be.an('array');

pm.expect(data.notes).lengthOf.at.least(1);

});

Skenario ke-2 selesai!

Skenario 2: Getting All Notes (Mendapatkan seluruh catatan)

~~Pastikan response memiliki status code 200.~~

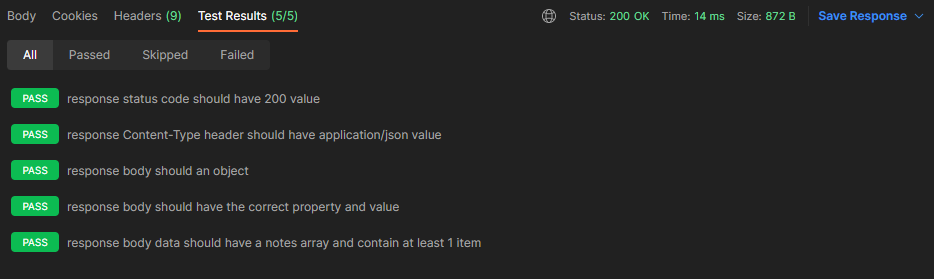
~~Pastikan header response Content-Type memiliki nilai application/json.~~

~~Pastikan body response adalah object.~~

~~Pastikan body response memiliki properti dan nilai atau tipe data yang sesuai.~~

~~Pastikan data pada response body memiliki array notes dan terdapat minimal 1 item di dalamnya.~~

Sekarang kita kirim permintaan dengan klik tombol Send dan lihat hasil pengujiannya.



Nice! Response yang dihasilkan server sesuai dengan ekspektasi kita

Simpan request ini dengan klik tombol Save dan kita lanjutkan ke skenario berikutnya.