**Skenario Delete Note**

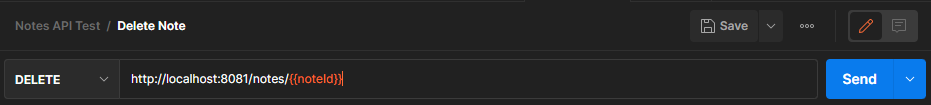
Kita menuju skenario terakhir yaitu menghapus catatan. Seperti biasa, yuk kita ingat-ingat kembali kriterianya.

Skenario 5: Delete Note (Menghapus 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 yang sesuai.
* Ketika mengakses catatan yang dihapus
* Pastikan catatan yang dihapus tidak ditemukan.
* Sudah siap? Ayo kita mulai!

Buat request baru di dalam collection dengan nama “Delete Note”.

Kemudian pada request URL, isi dengan nilai localhost:8081/notes/{{noteId}} dan ubah method yang digunakan menjadi DELETE.

Tak bosan-bosan, pertama kita buat pengujian untuk menguji status code dengan nilai 200. Silakan buat test dengan nama “response status code should have 200 value”.

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

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

});

*Done!*

* ~~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 yang sesuai.
* Ketika mengakses catatan yang dihapus
* Pastikan catatan yang dihapus tidak ditemukan.
* Sudah siap? Ayo kita mulai!

Lanjut buat tes dengan nama “‘response Content-Type header should have application/json value”.

Seperti biasa, di dalamnya kita uji response header Content-Type dengan nilai “application/json; charset=utf-8”.

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')

});

* + ~~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 yang sesuai.
  + Ketika mengakses catatan yang dihapus
* Pastikan catatan yang dihapus tidak ditemukan.

Kemudian, uji responseJson bahwa ia harus sebuah objek.

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

const responseJson = pm.response.json();

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

});

Done!

Skenario 5: Delete Note (Menghapus 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 yang sesuai.
* Ketika mengakses catatan yang dihapus
* Pastikan catatan yang dihapus tidak ditemukan.

Di dalamnya, dapatkan nilai body response dalam bentuk JSON objek.

Kemudian, uji responseJson bahwa ia harus memiliki properti dan nilai yang sesuai.

pm.test('response body should have 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('message');

pm.expect(responseJson.message).to.equals('Catatan berhasil dihapus');

});

*Done!*

* + ~~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 yang sesuai.~~
  + Ketika mengakses catatan yang dihapus
  + Pastikan catatan yang dihapus tidak ditemukan.

Terakhir, kita akan melakukan hal yang sama seperti skenario sebelumnya. Untuk memastikan catatan benar-benar dihapus oleh server, kita akan melakukan permintaan kepada catatan tersebut. Server harus mengembalikan response dengan status code 404 dan objek body yang relevan dengan status tersebut.

Yuk kita buat test baru dengan nama “when request the deleted note”.

Di dalamnya, lakukan request ke http://localhost:8081/notes/${noteId}. Nilai noteId kita dapatkan dari environment variable.

pm.test('when request the delete note', ()=>{

const noteId = pm.environment.get('noteId');

pm.sendRequest(`http://localhost:8081/notes/${noteId}`, (response, error)=> {

})

})

Di dalam fungsi callback response, bila parameter error tidak memiliki nilai, silakan buat test dengan nama “the deleted note should be not found”.

Di dalam test tersebut, uji status code response dengan nilai 404.

pm.test('when request the deleted note', () => {

const noteId = pm.environment.get('noteId');

pm.sendRequest(`http://localhost:5000/notes/${noteId}`, (error, response) => {

if(!error) {

pm.test('the deleted note should be not found', () => {

pm.expect(response.code).to.equals(404);

});

}

});

});

Anda bisa uji juga nilai body pada response. Pastikan ia memiliki properti dan nilai yang relevan dengan status code-nya.

pm.test('when request the deleted note', () => {

const noteId = pm.environment.get('noteId');

pm.sendRequest(`http://localhost:5000/notes/${noteId}`, (error, response) => {

if(!error) {

pm.test('the deleted note should be not found', () => {

pm.expect(response.code).to.equals(404);

const responseJson = response.json();

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

pm.expect(responseJson.message).to.equals('Catatan tidak ditemukan');

});

}

});

});

*Done!*

* + ~~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 yang sesuai.~~
  + ~~Ketika mengakses catatan yang dihapus~~
* ~~Pastikan catatan yang dihapus tidak ditemukan.~~

Skenario pengujian terakhir selesai kita tuliskan! Silakan kirim permintaan dengan klik tombol Send. Lihat hasil pengujiannya.

