

REST. Hjælp og dokumentation af din REST API (ekstra)

Tidligere opgaver

- [Design REST API](#)
- [Simpel Rest Service](#)
- [Rest med Cors](#)
- [Test Først af ManagerKlasse \(TDD\)](#)
- [RoutningURI-StatusKoder](#)
- [SøgningOgPaging](#)
- [Swagger/hjælpe-sider \(ekstra\)](#)

Opgave 1: Testing REST API med Postman

Du har nu en REST service med en række funktioner / metoder, som kan give forskellige svar tilbage.

Denne opgave går ud på at lave en række test (en test suite) som kan køres for at undersøge om din REST service reagerer som forventet.

OBS! Denne test kan sagtens laves inden selve REST-servicen kodes, hvorved du så laver en REST service, der ville være en TDD.

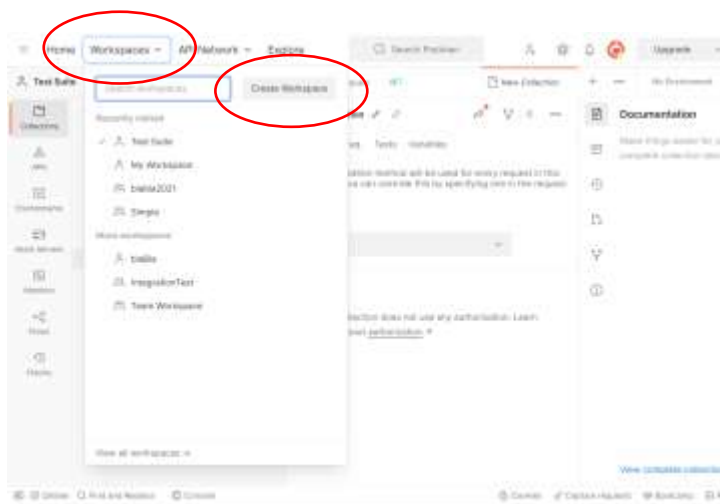
Her laver vi den på den REST-service du allerede har lavet i tidligere opgave.

You are to test your URI are well designed and working i.e. you are going to make an integration test. With this in mind you needed as minimum testing five entries (URI's):

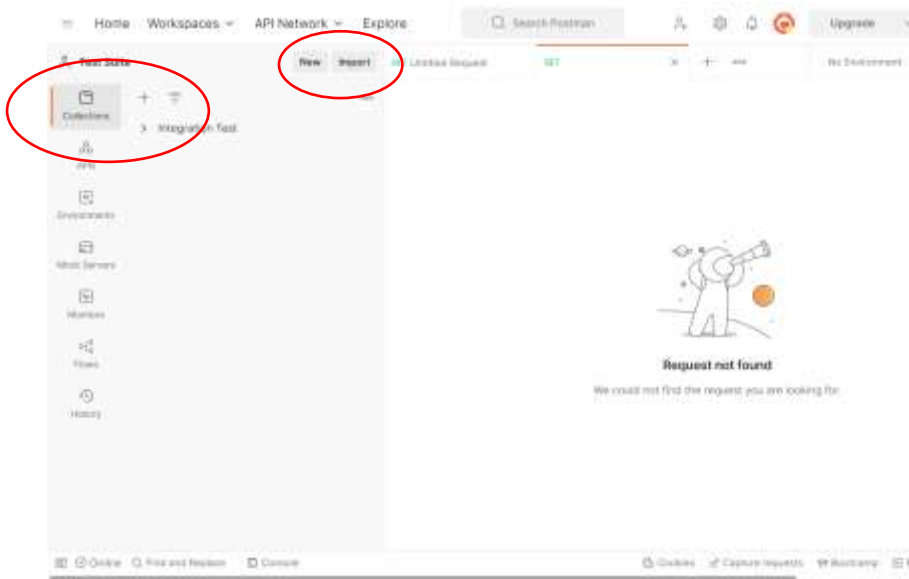
1. Read All (Get)
2. Read One (Get)
3. Create one (Post)
4. Update one (Put)
5. Delete one (Delete)

Opgave 1.1 Opret en test suite i POSTMAN.

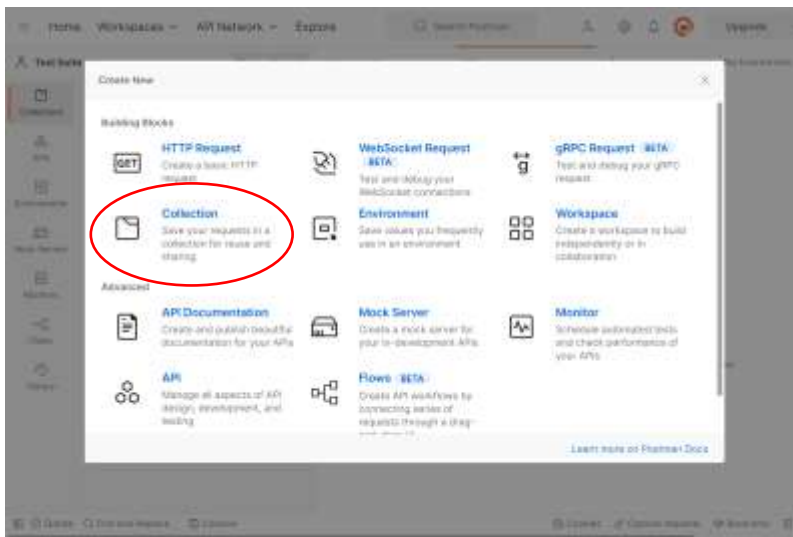
Du skal åbne POSTMAN og lave et nyt workspace fx 'Test Suite'



Derefter skal du oprette en 'collection' fx: 'Integration Test'



Vælg Collection



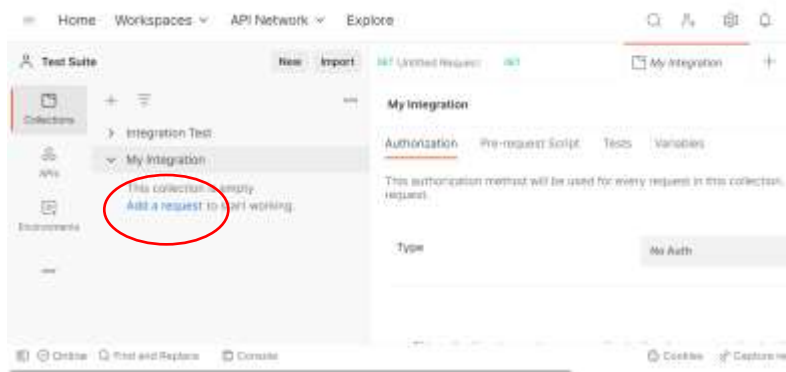
Opgave 1.2 Tilføj test til din Collection i din test suite i POSTMAN.

Du skal nu struktureret opbygge en række test, der har til formål at teste din REST service API, dvs. kommer du til de rette funktioner ved at sammenstille http metoder og URI'er, samt får du de rette statuskoder tilbage.

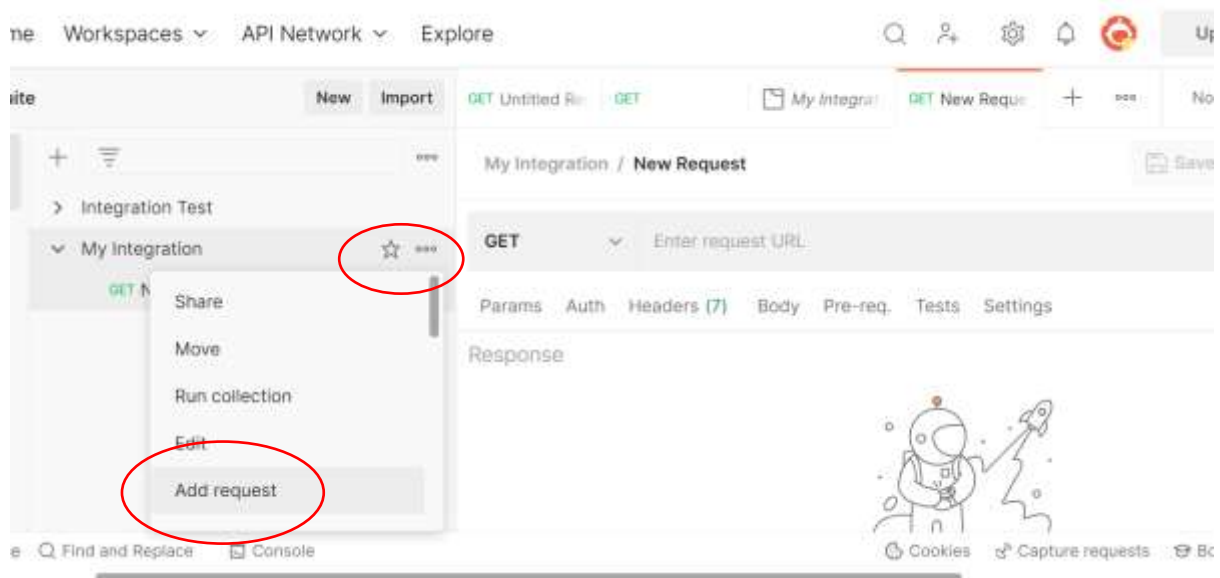
Du skal ikke som sådan teste funktionerne, som er testet i en component-test tidligere, dette er en integrations-test, altså virker REST service grænseflade (dvs. ve REST, HTTP metoder, URI, Statuskoder, evt. header-fields)

Du opbygger en række request (hver request er næste som de forespørgelser du tidligere har lavet med POSTMAN)

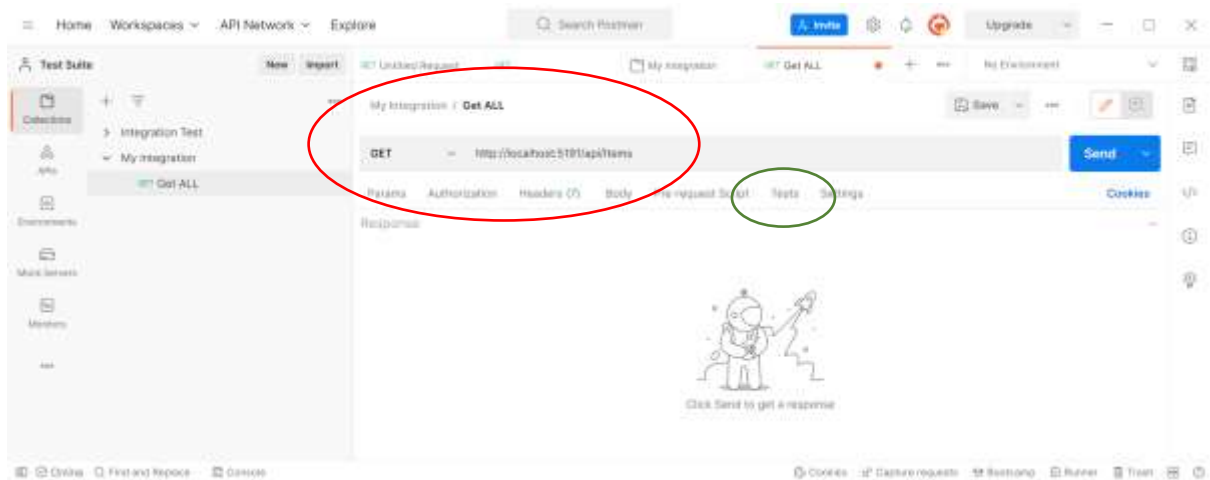
Du tilføjer et nyt request til din collection ved (første gang)



De følgende gange



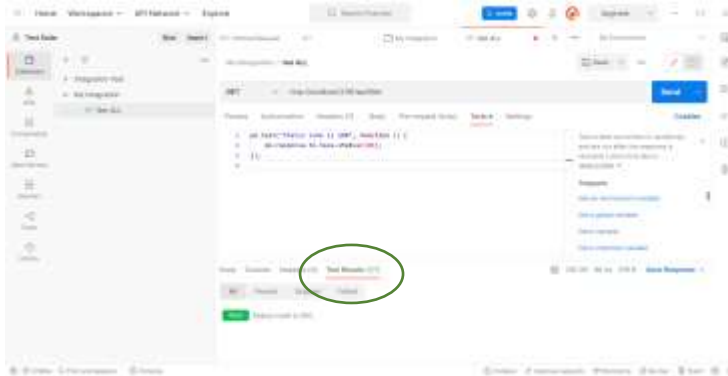
Dernæst opbygger du dit request fx:



Når du har lavet dit request skal du opstille test-kriterier (den grønne cirkel). Du kan fx: teste for status-koder:

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});
```

Prøv at sende dit request – og se svaret i Test-Results



NU kommer det vigtigste – Husk at gemme (save) dit request til din collection

Opgave 1.3 Lav alle dine test til din Collection i din test suite i POSTMAN.

Du skal nu tilføje alle dine test til din collection fx:

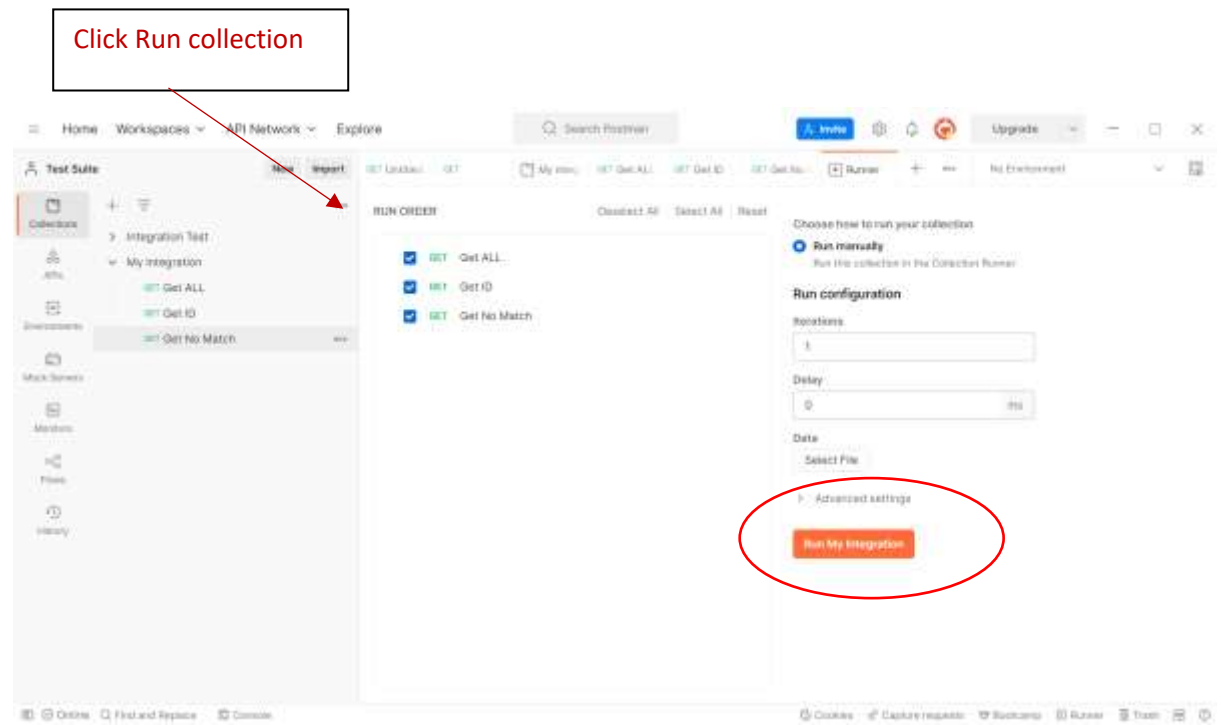
Get alle => Status kode 200

Get En => Status kode 200

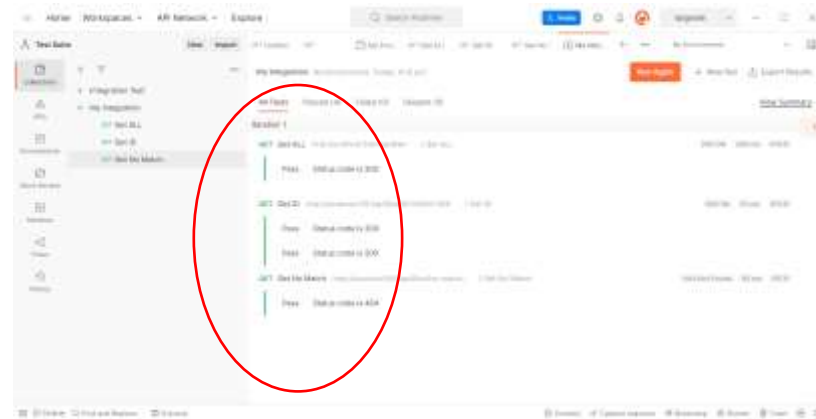
Get En => Status kode 404

Og så videre

Opgave 1.4 kør din Collection i din test suite i POSTMAN.



Dit resultat skulle gerne blive grønt 😊:



Opgave 2: Ekstra – mere Test

Du kan tilføje at du undersøger indholdet af svarene.

se her hvad du kan med test-script-sproget:

* https://learning.getpostman.com/docs/postman/scripts/postman_sandbox_api_reference/

* https://learning.getpostman.com/docs/postman/scripts/intro_to_scripts/

Du kan undersøge om du har sat CORS rigtigt op (benyt Options + aflæs header-fields)

Hvis du ønsker en mere Tutorial tilgang [se denne opgave](#) der har en Item-model-klasse