

Frank Serneels

portfolio vooruitgang

Software Testing

Redon Miftari

R0803827

Student Toegepaste Informatica

**INHOUDSTAFEL**

***No table of contents entries found.***

27/09/2021

Installatie van Python en PyCharm

12/10/2021

Ik ben gestart met het bestuderen en volgen van de python crash course op de cursus van Udemy.

Hier was het meer bestuderen van de syntax van python en hoe de code in elkaar zit, omdat deze toch verschillend is van C# en Java.

17/10/2021

[15:30] Ik ben gestart met de stappen te volgen op Udemy voor het setup van de test environment en werken met pycharm.

[16:10] Ik had veel tussentijdse problemen met instellen van de environment, maar heb alles gevonden en kon daaraan beginnen.

Websites die geholpen hebben met setup:

==> https://www.py4u.net/discuss/11889

==> https://phoenixnap.com/kb/install-pip-windows

[16:20] Github Repo in orde gebracht en beginnen pushen van files.

[16:40] Eerste test op github gepushed

25/10/2021

[16:05 - 17:05] Verder werken op Udemy aan testen, leren meerdere testen uitvoeren, verschil tussen unittest en integrated tests

[17:10 - 17:30] System tests met de patch functies en mock package

26/10/2021

[16:05 - 16:50] Verder werken aan Udemy testen.

30/11/2021

[22:30-23:05] Testen herbekijken en begrijpen, kennis terug opbouwen door lange stop.

[23:05-23:35] Verder werken aan testen met patches en aanleren van wat setUp is. Sectie 3 van Udemy Automated Software Testing with Python afgewerkt.

07/01/2022

[15:30 - 16:25] Section 4: Testing a Flask Endpoint 🡺 Beginnen en eindigen van deze sectie.

[19:00 – 20:10] Section 5: REST API testing Part I 🡺 Beginnen en eindigen van deze sectie.

[20:35-21:30] Section 6: REST API testing Part II 🡺 Beginnen en eindigen van deze sectie.

Verschillende testen:

Unit test: Is een test data 1 enkel deel van een app test dat onafhankelijk is van andere delen.

Integration test: Link testen tussen 2 verschillende delen van een app.

System tests: Test van het hele systeem dat de app bekijkt als de client van da gehele systeem.

## Testing a Flask Endpoint

**Text

Description automatically generated**

Eerst een folder aanmaken om hierin te werken, dan een virtual environment aanmaken binnen deze folder.

Graphical user interface, text

Description automatically generated

Setup van de testing environment voltooien.

Graphical user interface, application, website

Description automatically generated

Flask module importeren. (\_\_name\_\_) in python betekent een unieke string door de 2 \_.

Text

Description automatically generated

Testen dat Flask werkt:

Text

Description automatically generated

Logo

Description automatically generated with medium confidence

We gebruiken hier jsonify omdat een Flask Endpoint geen gewone dict kan returnen, dus moest dit in JSON-formaat uitgevoerd worden.

Text

Description automatically generated

Eerste test ging fout, na opzoeken op internet en in de gestelde vragen bleek dat het importeren van de file app in de test file fout was bij mij. Ik moest “**from app.app”** gebruiken zoals hieronder.



Test\_client() 🡺 Gaat minitesten in korte tijd mogelijk maken.

## REST API testing Part I



Deze commando wordt gebruikt om in de virtual environment te kunnen geraken en werken.

Text

Description automatically generated

2 eerste Unittests die bekijkt of de methode voor het aanmaken van items werkt. De weergave in JSON-formaat van de item werd ook getest, beide testen waren gelukt.

Text

Description automatically generated

BaseTest file wordt ook gebruikt als parent file voor de testen en de importatie van de modules te vergemakkelijken.

Text

Description automatically generated

Integration testing: Testen van de Databank en de CRUD-methodes die aan de databank gelinked zijn.

## Section 6: REST API testing Part II

Text

Description automatically generated

Text

Description automatically generated

Integrated testing van de CRUD operaties bij de stores en items.