



Lektionstillfälle 10

"API & Selenium Tester"

Utbildare: Robert Westin

NACKADEMIN

Kort summering av föregående lektioner

- Kubernetes
 - FluxCD
 - GitOps
- Köp en dator med minst 16 GB RAM...
- Monitorering
- Grafana (Visualisera)
- Loko (central loghantering)

Lektionstillfällets mål och metod

Mål med lektionen:

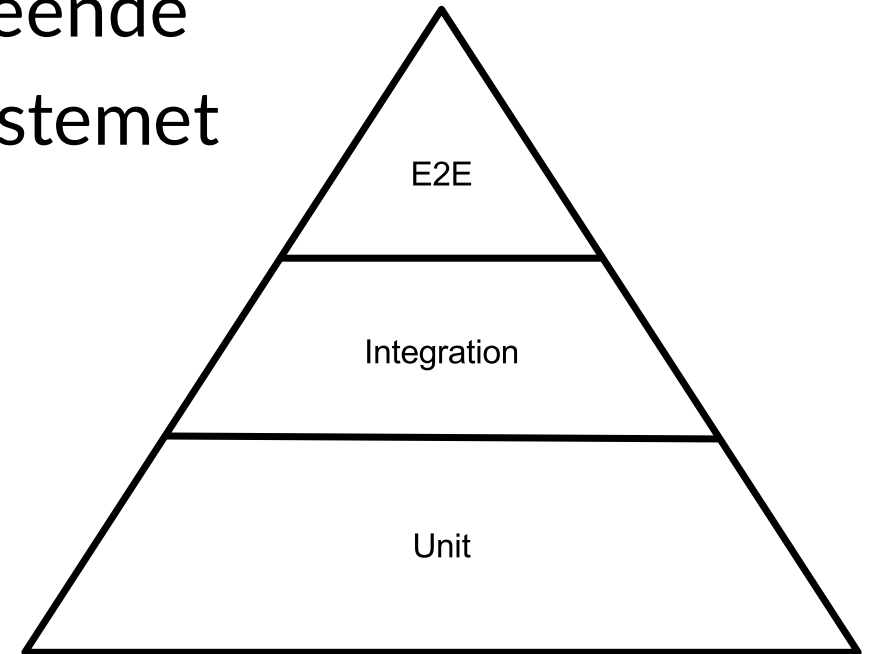
- Varför API, E2E Selenium tester
- Strategi för testdata
- Olika Typer av API:er
- Skriva testfall som API test & Selenium Test

Lektionens arbetsmetod/er:

- Föreläsning, Labb

Varför behövs API & Seleniumtester?

- Enhetstester testar delar av kodens beteende
- Integrationstester testar flera delar av systemet
- E2E tester hela systemet
- Vart testar API?
 - Kan vara både Integration & E2E
- Vart testar Selenium?
 - Oftast som E2E
 - Testar flöden



Test

- **Whitebox**
 - Vi vet vad vi testar, t.ex. i enhetstester
 - Vi kan påverka saker i systemet
 - Tänk: Vi har själva byggt presenten
- **Blackbox**
 - Vi har en låda
 - Tänk: Inslagen present i brevlådan

Exempel - Enhetstest

Test av enskild funktion

```
2  
3  
4  
5 ✓ def test_reverse_name():  
6     name = "Isak"  
7     assert reverse_name(name) == "kasI"  
8  
9
```

Exempel - Integrationstest

- Tänk en dörr
 - Går handtaget att trycka ner? (enhetstest av lås)
 - Går dörren att öppna (enhetstest av dörr)
 - Går dörren att öppna när handtaget trycks ner (integrationstest)
- Testa t.ex. flask med en test client <https://flask.palletsprojects.com/en/2.2.x/testing/#identifying-tests>
- Api Tester som testar delar av system, kan ses som integrationstester (Måste vara snabba)

Exempel – E2E test

- T.ex. selenium tester av ett helt system
- Black box tester
- Skapa en användare och logga in.
 - testar att databas och allt fungerar tillsammans som system

Vad är ett API?

- REST API (http)
- GraphQL
- MQTT
- GRPC
- Websocket

REST API

- Format t.ex. json, html, xml, text
- M2M kommunikations

T.ex. kan ett produktregister exponeras via REST services:

URI	Verb	Action
/products	GET	Hämtar en lista med alla produkter
/products	POST	Skapar en ny produkt
/products/{ProductID}	GET	Returnerar en produkt
/products/{ProductID}	PUT	Ersätter en produkt
/products/{ProductID}	DELETE	Tar bort en produkt
/product_groups/{ProductGroupID}	GET	Returnerar en produktgrupp
/products/{ProductID}	PATCH	Uppdaterar en produkt

Källa bild: wikipedia

NACKADEMIN

GraphQL

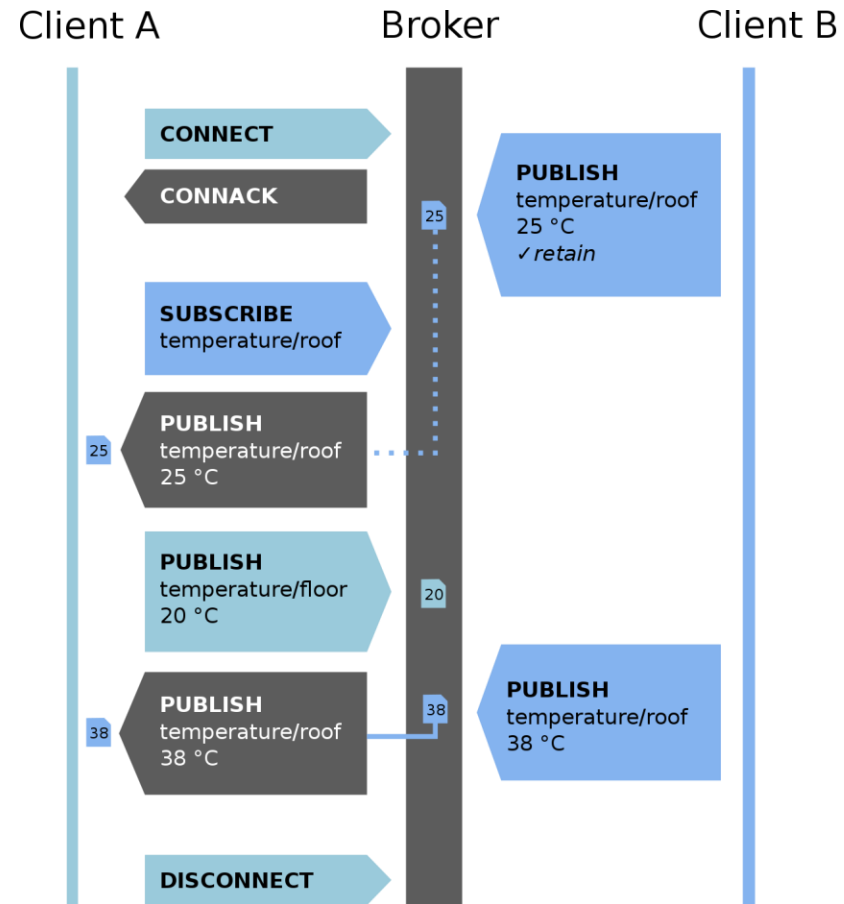
- Facebook
- Publikt 2015

```
{  
  orders {  
    id  
    productsList {  
      product {  
        name  
        price  
      }  
      quantity  
    }  
    totalAmount  
  }  
}
```

NACKADEMIN

MQTT

- MQTT
- IoT



Author: Simon A. Eugster

NACKADEMIN

GRPC

- **GRPC**
- **Google**
- **Http2**
- **Protobuf**

WebSocket

- **Full duplex**
- **Alternativ till Pull i browsern**

Vi kommer testa ett REST API

- **Vanligast**
- **Finns nästan till alla tjänster**
- **Exempel publika API:er är ofta REST**
- **Interna api:er mellan tjänster (om inte grpc, protobuf, events)**
- **Webhooks**

Testramverk

Testramverk i Python

Locust

Request med pytest blir ett ramverk

Pytest

Robot framework

Massa mer

NACKADEMIN

Varför testramverk?

Skriva själv:

Setup

Testfall

Report

Vad kan gå fel?

Utvärdera ramverk färdigt

Användning (nedladdningar)

Aktivt projekt? (Senaste ändring)

Programmeringsspråk

Integrationer

Övning

Skriv ett API – test

Test Data

Ett system är inte komplett utan data

Data påverkar hur systemet beter sig

Ingen data

Första gången du använder ett system

Systemet är fullt

Användare finns eller inte

Produkt finns eller inte

Test Data - Exempel

Testfall:

Skapa användare

Förväntat resultat:

200 Användare skapad

Faktiskt utfall:

400 Användare finns redan

Testdata - Strategi

- **Det finns olika strategier för att hantera testdata**
 - Hoppas det går bra (flaky test)
 - Ladda ren data vid varje körning
 - Kontrollera data vid körning (fixa om trasigt)
 - Sätt upp random data vid körning (rensa efter)
 - Basdata & migreringar via t.ex. liquibase eller yoyo-migrations

Testdata - Strategi

- **Övning skapa slumpvis data i testfall**
 - T.ex. använd tidstämpel i data

API Test - Övning

- **Skriv testfall för product**
 - Skapa (Create)
 - Läs (Read)



NACKADEMIN

Alternativ Selenium

Utökad selenium, cypress, robot framework etc.

- <https://applitools.com/>
- <https://saucelabs.com/>
- <https://www.browserstack.com/>
- Limited but easy start
 - <https://www.cypress.io/>
 - <https://www.browserstack.com/guide/cypress-vs-selenium>

No code

- <https://avoautomation.ai/>

NACKADEMIN

Populära ramverk för selenium

- Nightwatch
- WDIO
- Python Bindings
- <https://pypi.org/project/selenium/>

Hur fungerar selenium

- Python Bindings
- W3C Webdriver
- Browserdriver t.ex. chromedriver
- Browserstöd
- Webdriver exempel
 - Starta browser
 - Sök på google

Hur fungerar selenium

- Övning
 - Installera Chrome
 - Installera ChromeDriver
 - Ge kommando till chromedriver

Selenium - Selectors

Hur hittar vi element på sidan?

12.1 **Locator** Strategies

An **element location strategy** is an [enumerated attribute](#) deciding what technique should be used to search for [elements](#) in the [current browsing context](#). The following **table of location strategies** lists the keywords and states defined for this attribute:

State	Keyword
CSS selector	"css selector"
Link text selector	"link text"
Partial link text selector	"partial link text"
Tag name	"tag name"
XPath selector	"xpath"

NACKADEMIN

Källa: <https://www.w3.org/TR/webdriver1/#locator-strategies>

I Python Selenium

- **Exempel:**
 - Öppna browser
 - Sök på google
 - Ta en screenshot
 - Stäng browser

Browserstack Example

- Se Browserstack

Egen Selenium server

- Exempel arkitektur
- Chrome headless
- Se links

Framåtblick inför nästa lektion

- Enskild avstämning
- Hjälp med gruppuppgift
- Hjälp med enskild VG uppgift