Geen webservice? Geen probleem!

Webservice mocking met WireMock

Bas Dijkstra
www.ontestautomation.com
bas@ontestautomation.com
@_basdijkstra

WiFi

Wat gaan we doen?

_Stubbing, mocking en servicevirtualisatie

_WireMock

_Aan de slag

Voorbereiding

_Installatie Eclipse (of een andere IDE)

_Installatie TestNG-plugin (voor Eclipse)

_Installatie m2e (of tegenhanger voor andere IDE)

_Importeren Maven-project in IDE

_Update project (Eclipse) of tegenhanger

Problemen met testomgevingen

_Systemen bestaan uit veel componenten

_Onbeschikbaarheid voor testdoeleinden
  _Parallel ontwikkelen
  _Geen controle over testdata
  _Kosten voor gebruik third party component
  _…

Simulatie tijdens testuitvoer

- Simuleren van gedrag van afhankelijkheden

- Onbeschikbaarheid voor testdoeleinden
  - Parallel ontwikkelen
  - Geen controle over testdata
  - Kosten voor gebruik third party component
  - …

Stubbing

_Vooraf gedefinieerde responses

_Geen flexibiliteit

_Statusverificatie

Mocking

- Gedrag kan worden gedefinieerd tijdens initialisatie test
- (Iets) meer flexibiliteit
- Verificatie van gedrag

Servicevirtualisatie

_Simuleren van gedrag van complexe afhankelijkheden

_‘Next level’ stubbing / mocking

_Ondersteuning veelheid aan protocollen

_Data driven

WireMock

_http://wiremock.org

_Java

_HTTP mock server
  _werkt dus alleen voor HTTP(S)

_open source
  _ontwikkeld en onderhouden door Tom Akehurst

WireMock configureren

_JAR file toevoegen aan Java-project

_Maven (invoegen screenshot met dependency)

_Gradle (invoegen screenshot)

Een voorbeeldmock

_Invoegen eenvoudig voorbeeld waarin veelgebruikte opties worden getoond

Handige WireMock features

_Invoegen aantal features

_Volledige documentatie op
http://wiremock.org/docs/

WireMock standalone draaien

- Starten WireMock server
  - Opties: port, keystore, ...
- Mocks permanent beschikbaar stellen voor meerdere teams
- Mocks herconfigureren via JSON
- Invoegen voorbeeldcommando

WireMock starten en stoppen tijdens testuitvoer

_Integratie in testuitvoer

_Meeleveren mocks met testsuite in versiebeheer

_Integratie met JUnit (invoegen voorbeeld)

_Kan ook zonder gebruik te maken van JUnit

Demo

Gebruik van WireMock in tests

Aan de slag!

_WireMockExercises1

_Opzetten eenvoudige mocks

_Opdrachten staan in commentaar

_Controleren van werking door uitvoeren tests

Record & playback-opties

_Gebruik WireMock als proxy

_Opnemen request-response pairs (traffic)

_Genereren mock uit deze traffic

Record & playback-opties

_Gebruik WireMock als proxy

_Opnemen request-response pairs (traffic)

_Genereren mock uit deze traffic

Demo

Gebruik van record & playback in WireMock

Voor- en nadelen van record & playback

_Voordelen:
- _Snel opzetten mocks
- _Analyseren traffic met onbekende specificaties

_Nadelen:
- _Bij wijziging alles opnieuw opnemen
- _Mocks zijn niet flexibel
- _Mocks zijn slecht uitbreidbaar

_Vergelijk met record & playback bij testautomatisering

Request matching

_Bepalen welk antwoord wordt teruggestuurd op basis van specifieke eigenschappen request

_Opties:
  _URL
  _HTTP-methode
  _Query-parameters
  _Headers
  _Elementen in body van request
  _…

Voorbeeld: matching op URL

_Invoegen voorbeeld

Voorbeeld: matching op body-element

_Invoegen voorbeeld

Voorbeeld: matching op body-element

_Invoegen voorbeeld

Aan de slag!

_WireMockExercises2

_Gebruik request matching

_Opdrachten staan in commentaar

_Controleren van werking door uitvoeren tests

Foutsimulatie

- Uitbreiden van testdekking door simulatie fouten
- Vaak lastig te doen in ‘echte’ systemen
- Eenvoudig met mocks en stubs
- Testen foutafhandeling testobject

Voorbeeld: foutsimulatie door HTTP-statuscode

_Invoegen voorbeeld

Voorbeeld: foutsimulatie door timeout

_Invoegen voorbeeld

Voorbeeld: foutsimulatie door bad response

_Invoegen voorbeeld

Aan de slag!

_WireMockExercises3

_Gebruik foutsimulatie

_Opdrachten staan in commentaar

_Controleren van werking door uitvoeren tests

Stateful mocks

- Mocks tot nu toe waren stateless
- Volgorde van aanroepen methoden maakte niet uit
- Dit is in de ‘echte’ wereld niet altijd het geval
- Request A > request B geeft ander gedrag dan request B > request A

Stateful mocks: een voorbeeld

- Winkel met 2 exemplaren van een artikel in voorraad
- Eerste call naar addToCart: OK
- Tweede call naar addToCart: OK
- Derde call naar addToCart: NOK (foutmelding, foutcode, …)

Stateful mocks: een voorbeeld

_Invoegen implementatie voorbeeld

Aan de slag!

_WireMockExercises4

_Gebruik stateful mocks

_Opdrachten staan in commentaar

_Controleren van werking door uitvoeren tests

Andere nuttige features

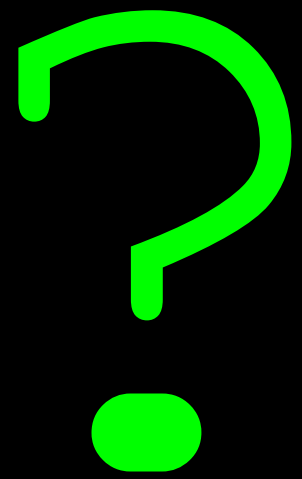
_Verificatie (is een bericht bij de mock aangekomen?)

_Transformatie van responses (via extensions)

_Integratie in een CI / CD pipeline

_Documentatie: http://wiremock.org/docs/

Vragen



Contact

_Email: bas@ontestautomation.com

_Weblog: http://www.ontestautomation.com

_Twitter: @_basdijkstra