



Belastingdienst

Leergang Regelbeheer

Testen - advanced



- Best practice valideren en testen
- Massale testfuncties
- Debuggen
- Testdekking
- Testen service



Best practice valideren en testen



Valideren en testen (1)

Begrippen:

- **Valideren**

Tijdens analyse-fase met voorbeelden toetsen of brondocumentatie goed is verwerkt en tijdens specificatie-fase toetsen of opgestelde **specificaties (regels en services)** hetzelfde resultaat geven als de voorbeelden uit de analyse-fase.

- **Testen**

Functioneel en technisch toetsen of de **(gegenereerde) functionaliteit** werkt conform de specificaties.



Valideren en testen (2)

Valideren

Wat	Fase voortbrenging	Met
Bron (wet, beleid, applicatie-documentatie)	Analyseren	Handmatig (voorbeelden)
Regels/services	Specificeren	ALEF-interpreter

Testen

Wat	Fase voortbrenging	Met
Regels/services	Genereren – build	Gegenereerde programmacode
Services	Genereren – build	Geïnstalleerde service
Services	Ketentest/integratietest	Geïnstalleerde service
Services	Acceptatietest	Geïnstalleerde service
Services	Productie – verificatie	Geïnstalleerde service



Scope van validatie in ALEF (1)

1. Validatie van attributen

Doel: valideren of waarden van individuele attributen goed worden afgeleid.

2. Validatie van individuele regels

Doel: valideren of specifieke regels goed zijn opgesteld.

- Dit alleen doen voor specifieke taalpatronen: consistentieregels, kenmerktoekenningen, constructies en feitcreaties (voor andere taalpatronen attribuuttesten gebruiken).

3. Validatie besturing in niet-declaratief model

Doel: valideren of condities in de besturing goed zijn ingericht.

- Stel testen op die voorspellen welke flows en regelgroepen geraakt worden i.p.v. welke waarden worden uitgerekend.



Scope van validatie in ALEF (2)

4. Validatie topflow of functionele component

Doel: valideren of regels in samenhang goed werken.

- Stel een (beperkte) set topflow-testen op die jaarafhankelijk zijn en daarmee het toepassen van de totale regeling inclusief het juist gebruik van de constanten testen.

5. Validatie specificatie van de service

Doel: valideren of specificatie van de service consistent is met andere delen van het model.

- Stel een (beperkte) set service-testen op. Baseer die op de topflow-testen zoals die voor de startflow van de service zijn opgesteld.



Werkwijze algemeen

Testen bijzondere gevallen

Stel voor attributen en regels validatiegevallen op met alle variaties die logisch mogelijk zijn (leeg/gevuld; grenswaarden; constanten variëren, enz.).

Opnemen constanten in testgeval

Jaaraanpassingen betreffen vaak ook wijzigingen van normen en constanten.

Door normen en constanten op te nemen in de invoer van een attribuut-/regeltest kan jaaronafhankelijk worden getest of de regel nog correct werkt.

Controleren volledigheid

Gebruik het dekkingsrapport om te toetsen op volledigheid van de validatieset.



Opdracht ALEF40 – Testen declaratieve flow

Opdracht ALEF41 – Testen niet-declaratieve flow

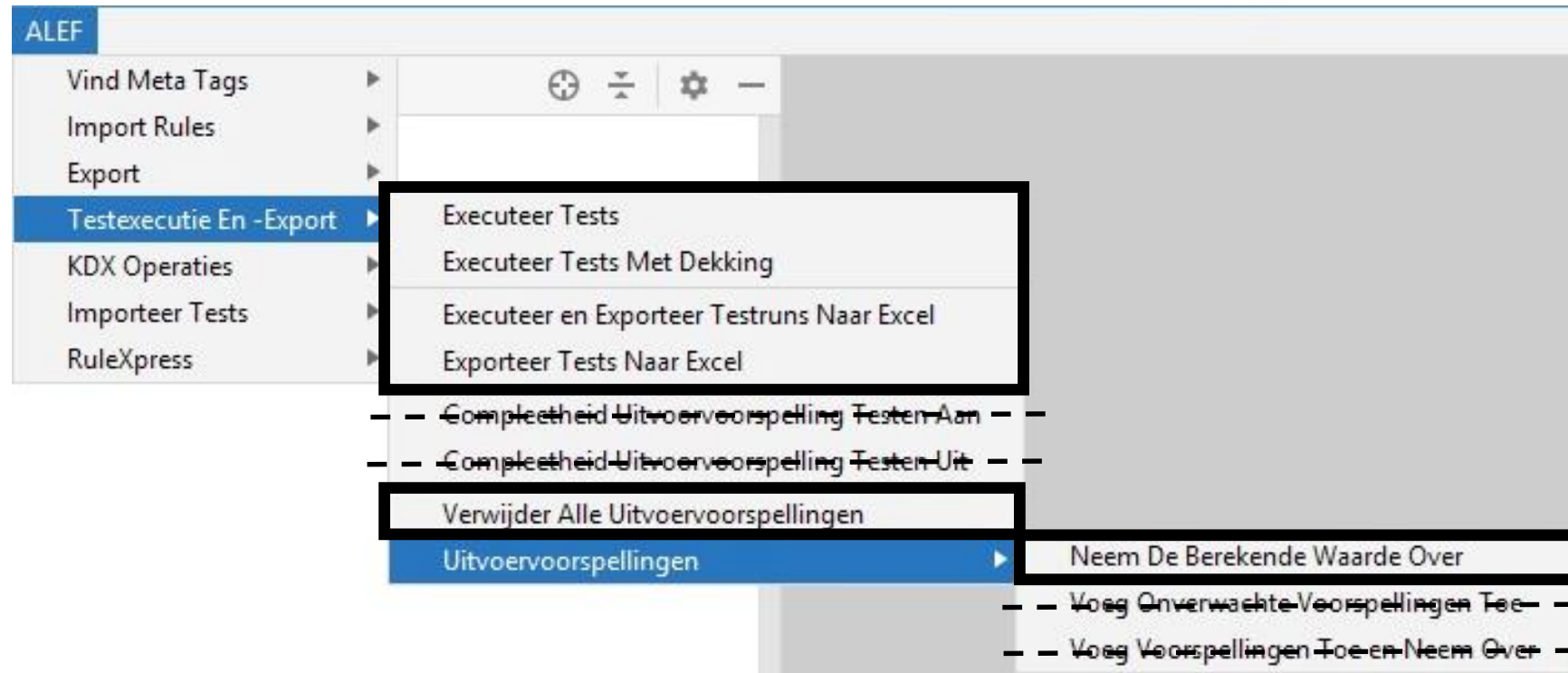


Massale testfuncties



Massale testfuncties (1)

In het ALEF-menu staan onder "Testexecutie En Export" een aantal functies die kunnen worden uitgevoerd op het niveau van testmodel, submap of testset:





Massale testfuncties (2)

Executeer Tests:

Alle testen binnen scope worden uitgevoerd. Het resultaat wordt in Excel samengevat.

Executeer Tests met dekking:

Alle testen binnen scope worden uitgevoerd. Het resultaat wordt in Excel samengevat en er wordt een dekkingsrapport aangemaakt.

Executeer en Exporteer Testruns naar Excel (voor een specifiek jaar):

Alle testen binnen scope worden uitgevoerd. Het resultaat wordt in Excel samengevat. De inhoud van de uitgevoerde testgevallen wordt naar een ander Excel-bestand geëxporteerd.

Exporteer Tests naar Excel:

De inhoud van de testgevallen binnen scope wordt naar een Excel-bestand geëxporteerd.

Verwijder Alle Uitvoervoorstellingen:

De volledige uitvoervoorstelling wordt verwijderd. Zowel de attributen/kenmerken als de waarden.

Neem De Berekende Waarde Over:

De op basis van de regels berekende waarde wordt overgenomen in de uitvoervoorstelling.



Massale testfuncties (3)

Run – massaal uitvoeren testen binnen ALEF

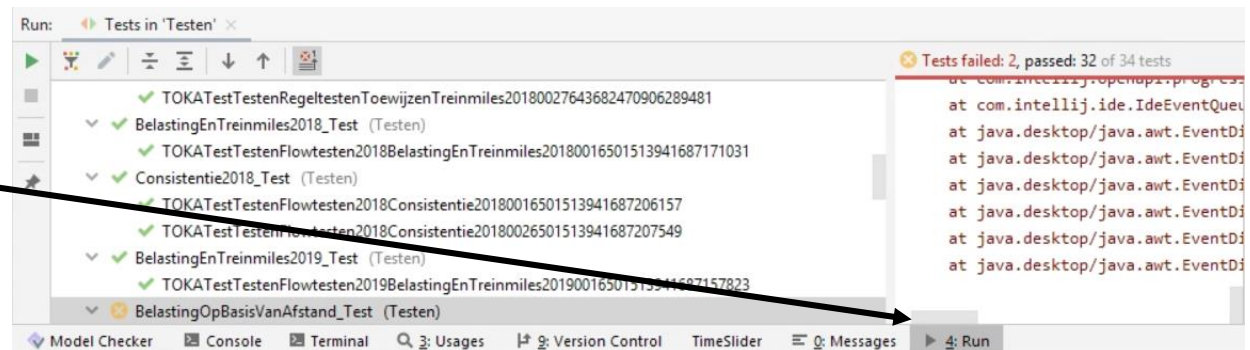
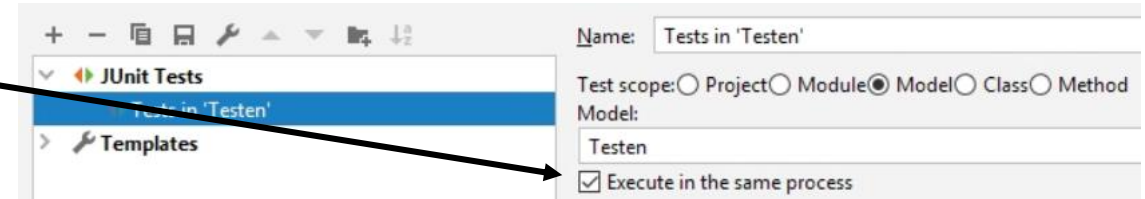
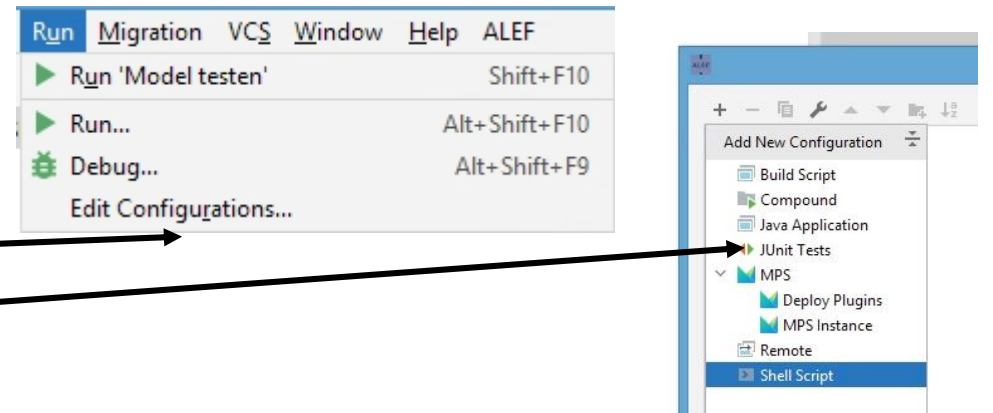
Aanmaken van testscope op het niveau van project, solution of model via "Run - Edit configurations...".

Kies vervolgens  en  JUnit Tests

Let op: in configuratie "Execute in the same proces" aanvinken.

Met menu-optie "Run" worden alle testen binnen scope uitgevoerd.

Het resultaat wordt getoond in het deelscherm onder de tab "4. Run"

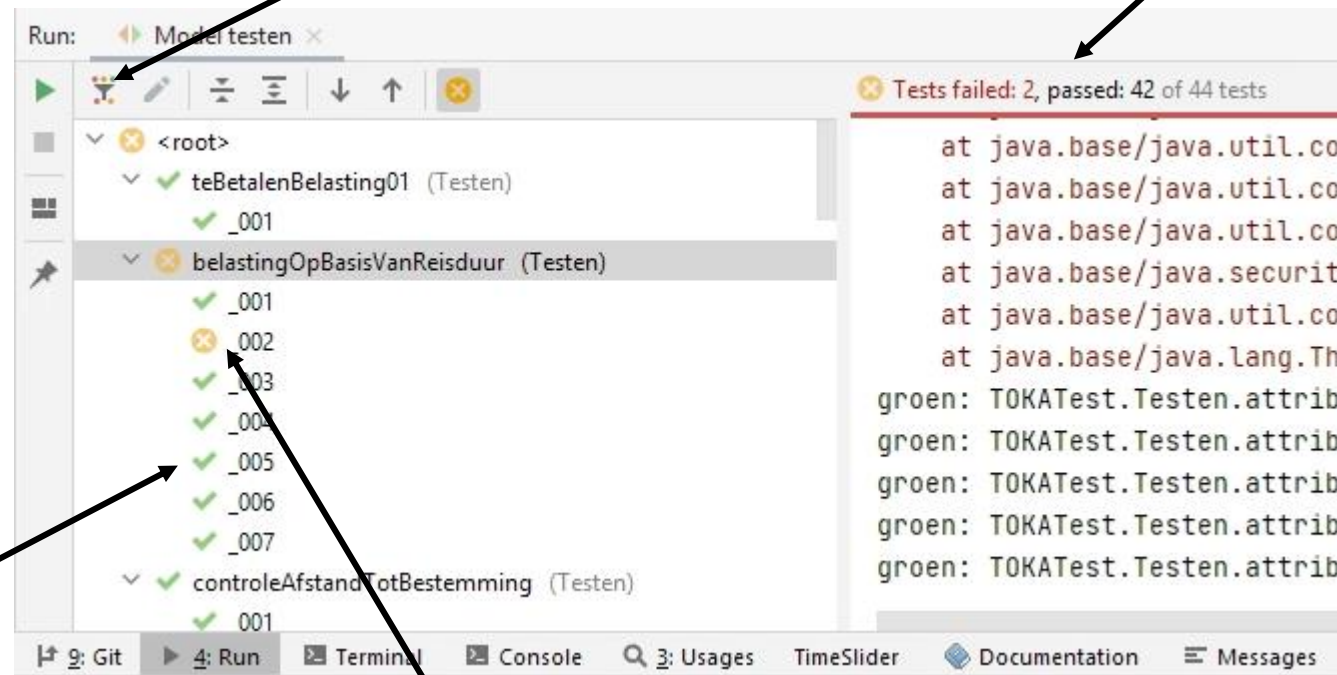




Massale testfuncties (4)

 = Filter voor alleen gefaalde testen

Totaal resultaat



Tests failed: 2, passed: 42 of 44 tests

```
at java.base/java.util.co
at java.base/java.util.co
at java.base/java.util.co
at java.base/java.securit
at java.base/java.util.co
at java.base/java.lang.Th
groen: TOKATest.Testen.attrib
groen: TOKATest.Testen.attrib
groen: TOKATest.Testen.attrib
groen: TOKATest.Testen.attrib
groen: TOKATest.Testen.attrib
```

✓ = Geslaagde test

✗ = Gefaalde test



Demo massale testfuncties in ALEF



Debuggen



Aanpak debuggen (1)

Enkele algemene tips voor het werken met de debugger bij complexe fouten zijn:

1. Controleer de invoer en uitvoervoorspelling.
2. Kijk, met materiekkennis, naar de berekende waarden en zoek naar opvallende waarden. Klik op het vraagteken (?) achter de afwijkende waarde om naar de regel te gaan die die waarde heeft afgeleid en loop vanaf daar de keten af.
3. Gebruik de executielijst om snel te zien welke variabelen welke waarden hebben gekregen en, in een niet-declaratief model, op welk punt in de berekening.



Aanpak debuggen (2)

Als er sprake is van een groot aantal testen dat niet slaagt, dan kan voor de analyse het volgende globale stappenplan worden gevolgd:

1. Kies een testgeval op het "laagste" niveau uit de niet-geslaagde testen (dus liefst op regelniveau). Deze zijn het meest eenvoudig te analyseren.
Draai de individuele test.
2. Analyseer met behulp van de debugger de fout.
3. Voer de aanpassing door die de fout herstelt.
4. Draai de individuele test opnieuw.
5. Als de test nu groen is,
 - a) draai dan opnieuw de massale test,
 - b) bekijk welke testen nog niet goed zijn,
 - c) herhaal de voorgaande stappen 1 tot en met 4.Als de test nog rood is, analyseer de test opnieuw en probeer vast te stellen of de aanpassing desondanks terecht was of dat die moet worden teruggedraaid.



Opdracht ALEF43 – Debuggen



Testdekking



Testdekking

- Geeft inzicht in mate waarin de testen in de testsets alle paden in de regels raken.
- Hulpmiddel om testset te analyseren en aan te vullen.

```

3 belasting op basis van afstand 03
3 belasting op basis van afstand 03 vanaf 2018
3 De belasting op basis van afstand van een passagier = ...
3 7
  Er wordt voldaan aan alle volgende voorwaarden : - de reis is een belaste reis - de passagier voldoet aan ten minste één van de volgende voorwaarc

9 1 De reis is een belaste reis
4 5
  De passagier voldoet aan ten minste één van de volgende voorwaarden : -- de passagier is een passagier jonger dan 18 jaar -- de passagier is ee

1 8 De passagier is een passagier jonger dan 18 jaar
3 5 De passagier is een passagier van 25 tot en met 64 jaar
3 1
  De reis voldoet aan alle volgende voorwaarden : -- de afstand tot bestemming in kilometers van de reis is groter dan de BOVENGRENS AFSTAND I

3 1 De afstand tot bestemming in kilometers van de reis is groter dan de BOVENGRENS AFSTAND EERSTE SCHIJF
3 0 De afstand tot bestemming in kilometers van de reis is kleiner of gelijk aan de BOVENGRENS AFSTAND TWEEDE SCHIJF

```

Aantal malen dat de regel / regelversie tot een toekenning heeft geleid.

N.B. Binnen een testgeval kan een regel meerdere malen worden uitgevoerd als sprake is van een model met meervoudige objecten.

Rood: Aantal malen dat niet aan voorwaarde wordt voldaan.

Groen: Aantal malen dat aan voorwaarde wordt voldaan.



Demo dekkingsrapport in ALEF



Opdracht ALEF44 – Testdekking



Valideren en testen service



Valideren service binnen ALEF

- Doel: valideren of service functioneel correct gespecificeerd is
 - Berichten
 - Mapping
- ServiceTestSets waarin berichten conform service-specificatie worden opgesteld. Let op: volgorde van elementen in berichten is deels verplicht (zie xsd).
- Notatie van gegevens:
 - Decimalen met een punt weergeven
 - True/false
 - Datumnotatie: yyyy-mm-dd
- Het vullen van het resultaat van de service-aanroep is niet nodig voor testen binnen ALEF, maar wel voor een correct resultaat bij het uitvoeren van de servicetesten met Blaze tijdens de build:
 - Resultaatcode = 1
 - Resultaatmelding = SERVICE_OKAls sprake is van een technische fout, dan wordt een SOAP-fault-bericht teruggegeven.
- Debuggen conform Testsets.

Dient het volgende resultaat te hebben:

```
Service aanroep
resultaatcode    = 1
resultaatmelding = SERVICE_OK
```



Demo opvoeren servicetest in ALEF



Opdracht ALEF45 – Valideren rekenservice

Opdracht ALEF46 – Valideren acceptatieservice