



Belastingdienst

Leergang Regelbeheer

Testen - basis



- Valideren in ALEF
- Gebruik van de TestSet
- Debuggen



Valideren in ALEF



Objecten van validatie/test

Belangrijk onderdeel van de visie op voortbrenging van BRM-services is het voortdurend valideren/testen van (tussen-)producten tijdens de voortbrenging.

De “Best practice valideren en testen” beschrijft hoe projecten validatie en test kunnen inrichten. In de module ALEF301 Testen – advanced wordt deze best practice toegelicht.

ALEF biedt uitgebreide functionaliteit voor het valideren/testen van:

- Attributen, regels en regelgroepen
- Flows voor de besturing van de uitvoering van regelgroepen
- Flows voor de werking van een (deel van een) totaalmodel
- Services



Valideren van regels en regelgroepen

Validatiegevallen van regels en regelgroepen kunnen worden gespecificeerd in een **TestSet**.

In een TestSet wordt de scope van de validatie vastgelegd. De scope geeft aan welke regels betrokken worden in de validatie.

Mogelijke niveaus van de in te stellen scope:

- Attribuut
- Regel
- Regelgroep
- Regelgroepenlijst

Hiermee kunnen losse elementen in het regelmodel worden gevalideerd.



Valideren/testen van flows

- Flowtesten worden (net als testen voor regel(groepen)) gespecificeerd in een **TestSet**
- Flowtesten worden gebruikt voor het valideren van de besturing van een **procedureel** regelmodel.
- Deze testen kunnen worden uitgevoerd voor de hoofdfLOW of voor losse sub-flows.
- Het **hele model** kan worden gevalideerd met een testset van de **hoofdfLOW**.
- Alleen de regels uit regelgroepen die zijn opgenomen in de flow vallen binnen de scope van de validatie.

Valideren/testen van flows wordt verder behandeld in **ALEF310 Testen – advanced**.



Valideren/testen van services

- Voor het valideren van een service moet een **ServiceTestSet** worden aangemaakt. Die kent een andere specificatie dan de TestSet.
- Dit is een **interne validatie** binnen ALEF waarmee de **specificatie** van de service gevalideerd wordt in relatie tot het regelmodel.
- Na het genereren en deployen van de service wordt de “echte” service in de delivery pipeline of met behulp van SOAP-UI getest.
- Ten slotte wordt de service uiteraard ook **extern getest**, technisch en functioneel, in combinatie met een consumer (afnemende applicatie) die de service gebruikt.

Valideren/testen van services wordt verder behandeld in **ALEF310 Testen – advanced**.



Gebruik van de testset



Scope en geldigheidsperiode

De **scope** van de test wordt in de kopregel vastgelegd

Welke **parametersets** worden gebruikt voor de validatie, wordt afgeleid van de geldigheidsperiode van de testset en de geldigheid van de parametersets.

Testset te betalen belasting voor alle regels die te betalen belasting bepalen

Gebruikte parameters: *Parameters 2018*

Geldigheidsperiode vanaf 1-1-2018 t/m 2020

Datums voor testuitvoering 1-7-2018*, 1-7-2019, 1-7-2020 (lokaal te gebruiken datum 1-7-2018)

Een TestSet heeft een **geldigheidsperiode** die aansluit op de geldigheid van regels en parameters in scope van de test. Per jaar moet een testdatum worden opgenomen.

Voor welk jaar/welke datum de validatie in ALEF wordt uitgevoerd wordt vermeld als "**lokaal te gebruiken datum**". Tijdens de geautomatiseerde validatie in de Delivery Pipeline wordt ook de validatie voor de andere datums uitgevoerd.



Gebruik meervoudige objecten

In een testgeval waarbij attributen van meerdere objecten worden gebruikt, moeten beide **objecten** in de in- en/of uitvoer worden opgenomen.

Daarbij moet door middel van een rol de relatie met het andere object worden opgenomen.

► Test 001

de volgende situatie:

Vlucht KL1002 met
passagier = {Jaap}
is belaste reis = waar
reisduur per trein in minuten = 150

Natuurlijk persoon Jaap met
belasting op basis van afstand = 100

moet het volgende resultaat hebben:

Natuurlijk persoon Jaap met
belasting op basis van reisduur = 100

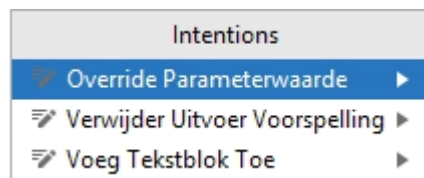


Gebruik parameters

Standaard worden parameters uit parametersets gebruikt.

Om testgevallen onafhankelijk van periodieke wijzigingen van parameters te maken, kunnen parameters worden opgenomen in het individuele testgeval.

Intention "Override Parameterwaarde".



▶ Test 001

de volgende parameterwaarden:

BOVENGRENS REISDUUR EERSTE SCHIJF	=	300
BOVENGRENS REISDUUR TWEDE SCHIJF	=	600
PERCENTAGE REISDUUR EERSTE SCHIJF	=	100%
PERCENTAGE REISDUUR TWEDE SCHIJF	=	50%
PERCENTAGE REISDUUR DERDE SCHIJF	=	0%

de volgende situatie:

Vlucht KL1002 met

passagier = {Jaap}

is belaste reis = waar

reisduur per trein in minuten = 150

Natuurlijk persoon Jaap met

belasting op basis van afstand = 100

moet het volgende resultaat hebben:

Natuurlijk persoon Jaap met

belasting op basis van reisduur = 100



Debuggen



Executielijst

Nadat een test is uitgevoerd wordt een lijst getoond met regels die tot een toekenning hebben geleid.

✓ Test 002

de volgende situatie:

Vlucht KL1002 met
 passagier = {Jaap}
 is belaste reis = waar
 afstand tot bestemming in kilometers = 100

Natuurlijk persoon Jaap met
 is passagier van 65 jaar of ouder = waar

moet het volgende resultaat hebben:

Natuurlijk persoon Jaap met
 belasting op basis van afstand = 168

> Parameterset: Parameters 2018

[1] belasting op basis van afstand 04 vanaf 2018
 Jaap.Natuurlijk persoon.belasting op basis van afstand := 168

Door middel van Ctrl-click op de regel of door te klikken op het vraagteken achter een resultaat, kan worden gesprongen naar de uitgevoerde regel voor verdere analyse.

✓ Test 001

de volgende situatie:

Vlucht KL0001 met
 passagier = {Jaap, Joap}
 afstand tot bestemming in kilometers = 630
 bereikbaar per trein = waar
 datum van de vlucht = dd. 1-7-2018
 reisduur per trein in minuten = 451
 gebruik fossiele brandstof minder dan 50 procent = waar

Natuurlijk persoon Jaap met
 geboortedatum = dd. 1-5-1980

Natuurlijk persoon Joop met
 geboortedatum = dd. 5-12-1932

moet het volgende resultaat hebben:

Vlucht KL0001 met
 totale belasting op basis van afstand = 270,9
 totale belasting op basis van reisduur = 135
 totaal te betalen belasting = 385
 er zijn niet-voorspelde eigenschappen

Contingent treinmiles ? <- constructie van vastgesteld contingent treinmiles 01 met
 aantal treinmiles op basis van aantal passagiers = 200
 totaal aantal treinmiles voor evenredige verdeling = 2700
 restant na evenredige verdeling = 0
 er zijn niet-voorspelde eigenschappen

Natuurlijk persoon Jaap met
 belasting op basis van afstand = 116,1
 belasting op basis van reisduur = 58
 te betalen belasting = 164
 treinmiles op basis van evenredige verdeling = 1350
 er zijn niet-voorspelde eigenschappen

Natuurlijk persoon Joop met
 belasting op basis van afstand = 154,8
 belasting op basis van reisduur = 77
 te betalen belasting = 221
 treinmiles op basis van evenredige verdeling = 1350
 er zijn niet-voorspelde eigenschappen

> Parameterset: Parameters 2018

✓ [1] Flow: Belasting en treinmiles vanaf 01-01-2018

[1] aantal passagiers 01 vanaf 2018
 KL0001.Vlucht.aantal passagiers := 2

[1] constructie van vastgesteld contingent treinmiles 01 vanaf 2018
 KL0001.Vlucht.vastgesteld contingent treinmiles := [constructie van vastgesteld contingent treinmiles 01]
 constructie van vastgesteld contingent treinmiles 01.reis met treinmiles := [KL0001.Vlucht]
 constructie van vastgesteld contingent treinmiles 01.aantal passagiers op basis van aantal passagiers := 200

[1] feitcreatie van passagier met recht op treinmiles 01 vanaf 2018
 constructie van vastgesteld contingent treinmiles 01.passagier met recht op treinmiles := [Jaap.Natuurlijk persoon]
 Jaap.Natuurlijk persoon.te verdelen contingent treinmiles := [constructie van vastgesteld contingent treinmiles 01]

[2] feitcreatie van passagier met recht op treinmiles 01 vanaf 2018
 constructie van vastgesteld contingent treinmiles 01.passagier met recht op treinmiles := [Joop.Natuurlijk persoon]
 Joop.Natuurlijk persoon.te verdelen contingent treinmiles := [constructie van vastgesteld contingent treinmiles 01]

[1] totaal aantal treinmiles vanaf 2018
 constructie van vastgesteld contingent treinmiles 01.totaal aantal treinmiles voor evenredige verdeling := 2700

[1] leeftijd 01 vanaf 2018
 Jaap.Natuurlijk persoon.leeftijd := 38 jaren

[2] leeftijd 01 vanaf 2018
 Joop.Natuurlijk persoon.leeftijd := 85 jaren

[1] belaste reis 01 vanaf 2018
 KL0001.Vlucht.belaste reis := waar

[1] passagier van 25 tot en met 64 jaar vanaf 2018
 Jaap.Natuurlijk persoon.passagier van 25 tot en met 64 jaar := waar

[1] belasting op basis van afstand 03 vanaf 2018
 Jaap.Natuurlijk persoon.belasting op basis van afstand := 116,1

[2] passagier van 65 jaar of ouder vanaf 2018
 Joop.Natuurlijk persoon.passagier van 65 jaar of ouder := waar

[2] belasting op basis van afstand 05 vanaf 2018
 Joop.Natuurlijk persoon.belasting op basis van afstand := 154,8

[1] belasting op basis van reisduur 01 vanaf 2018
 Rij 2 van belasting op basis van reisduur 01(instance of Geldigheidsperiode) = waar
 Jaap.Natuurlijk persoon.belasting op basis van reisduur := 58

[2] Rij 2 van belasting op basis van reisduur 01(instance of Geldigheidsperiode) = waar
 Joop.Natuurlijk persoon.belasting op basis van reisduur := 77

[1] leeftijd numeriek 01 vanaf 2018
 Jaap.Natuurlijk persoon.leeftijd numeriek := 38

[2] leeftijd numeriek 01 vanaf 2018
 Joop.Natuurlijk persoon.leeftijd numeriek := 85

f11 maximaal aantal te ontvangen treinmiles 01



Symbolen

De waarde van de toekenning of waarmee evaluatie van de regel is uitgevoerd

Regel belasting op basis van afstand 04
geldig vanaf 2018

De belasting op basis van afstand [168] van een passagier van een reis moet berekend worden als het HOGE BASISTARIEF EERSTE SCHIJF [170]? min (het HOGE TARIEF VERMINDERING EERSTE SCHIJF [0,02]? maal de afstand tot bestemming in kilometers [100]?)

indien ✓ aan a[1] volgende voorwaarden wordt voldaan:

- ✓ de reis is een belaste reis?
- ✓ hij voldoet aan ten minste één van de volgende voorwaarden:
 - ✗ hij is een passagier van 18 tot en met 24 jaar?
 - ✓ hij is een passagier van 65 jaar of ouder?
- ✓ de afstand tot bestemming in kilometers [100]? is kleiner dan de BOVENGRENS AFSTAND EERSTE SCHIJF [500]?.

[?] = link naar oorsprong van waarde

✓ = de conditie is waar / ✗ = de conditie is niet waar / ? = de conditie is niet geëvalueerd



Demo Testspraak in ALEF



Opdracht ALEF30 – Valideren rekenregel,
opdracht ALEF31 – Valideren consistentieregel en
opdracht ALEF32 – Valideren attributen