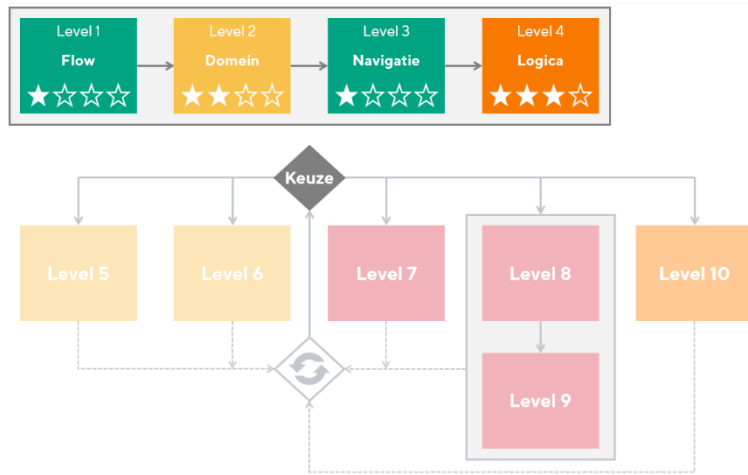


LEVEL 4: LOGICA

NIVEAU: ★ ★ ★ ☆

De applicatie geeft nu pagina's weer en je kunt ertussen navigeren. De informatie die je op de aanvraagpagina invult, wordt echter nog nergens voor gebruikt. In dit level voegen we logica toe aan de applicatie.



Hieronder staan de specificaties van de logica zoals aangeleverd door de fictieve zorgverzekeraar. Deze gaan we in kleine stappen modelleren in Blueriq. **Je hoeft die dus nu niet in één klap te modelleren!**

De premie van de polis wordt berekend met behulp van de volgende formule:

**PremiePerMaand =
PremieBasisVerzekering + PremieAanvullendeVerzekering - Korting**

De premie voor de basisverzekering is altijd € 99,-.

Klanten die kiezen voor een eigen risico van meer dan € 250,- krijgen 5% korting op de basispremie.

De premie voor aanvullende tandartsverzekering is € 10,- en voor aanvullende fysiotherapie € 12,-

Een aanvrager wordt gezien als gezond als hij of zij **dagelijks** meer dan 1 uur beweegt of **wekelijks** meer dan 5 uur.

1. Aangezien de aanvullende premie niet altijd hetzelfde is, maar afhangt van de waarde die je kiest als type aanvullende verzekering, is een **beslistabel (decision table)** het element dat deze beslissing het beste kan modelleren.

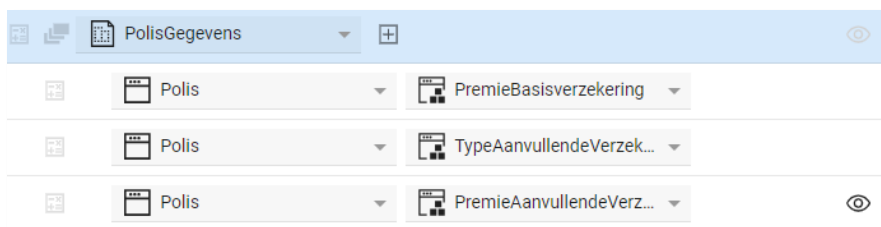
Maak de beslistabel zoals weergegeven aan **en maak hem verder af**. Deze beslistabel (kies een zinvolle naam) bepaalt de waarde van de premie van de aanvullende verzekering voor alle mogelijke aanvullende verzekeringen: Tandarts, Fysiotherapie, beide (het toverwoord is **AND**) of geen enkele (**unknown** is een **?** in Blueriq).

Polis.TypeAanvullendeVerzekering	"Tandarts"
Polis.PremieAanvullendeVerzekering	10
Justifications	

Een kolom toevoegen aan een beslistabel kan door in een cel te gaan staan en dan op de hangende druppel met een plus erin **+** te klikken. De volgorde van kolommen doet er doorgaans niet toe, maar omdat Blueriq stopt met afleiden als er een waarde is bepaald én de tabel van links naar rechts afgewerkt wordt, geldt voor deze tabel dat de volgorde wel van belang is. In ieder geval dient de kolom voor beide aanvullende verzekeringen vooraan te staan!

Koppel de beslistabel **niet** met een default value aan een attribuut! Het modelleren van de beslistabel zelf is genoeg.

Als je wilt kijken in de Runtime, is de kans aanwezig dat deze tabel "het nog niet doet". Dit komt omdat we in de container of in die container op de pagina nog even moeten vertellen dat het attribuut **Polis.TypeAanvullendeVerzekering** bij wijziging aanleiding is voor een herberekening en dus een **Refresh** moet veroorzaken.



Is de aanvullende premie die berekend wordt ook **Read only**? ☹️ Dat is noodzakelijk, want anders berekent Blueriq dat attribuut niet, hoeveel logica je er ook op loslaat.

2. De berekening voor het attribuut voor de premie van de basisverzekering is altijd hetzelfde, het is een constante voor dit attribuut! Daarvoor gebruiken we de **default value (constant)** bij het attribuut. Zet de **default value** van het attribuut **Polis.PremieBasisVerzekering** op een **Constant** van **99**.

Attribute		Default value
Entity Polis	Name PremieBasisverzekering	Default value Constant
Data type Currency	Value list	Constant 99
<input type="checkbox"/> Multivalued		

3. De waarde van het attribuut **Polis.Korting** is
- ofwel 0 euro,
 - ofwel 5% van de basispremie.

Dat hangt af van je eigen risico. We kunnen een **business rule** maken in combinatie met een **default value (constant)** van 0 euro om deze korting af te leiden.

Een **beslistabel** met een kolom voor 0 euro en een kolom 5% had ook gekund.

Business rule
If $Polis.EigenRisico > 250$
Then Polis.Korting
Is $Polis.PremieBasisverzekering * 0.05$

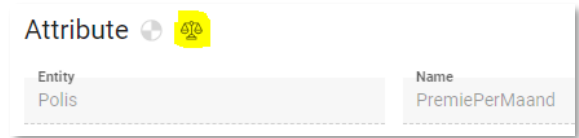
4. De berekening van het attribuut **Polis.PremiePerMaand** is een eenvoudige optel- en aftreksom van drie attributen. Maak daarvoor een **default value (expression)** bij het attribuut.

Nu we de logica elementen hebben gemaakt om alle attributen te berekenen en dus bepaald hebben wat de maandelijkse premie zal zijn, kun je je applicatie opnieuw openen in de Runtime en de berekeningen bekijken. Heb je alle **Read only's** goed staan (voor berekende velden) en de **Refresh** op invoervelden die bijdragen aan de berekening?

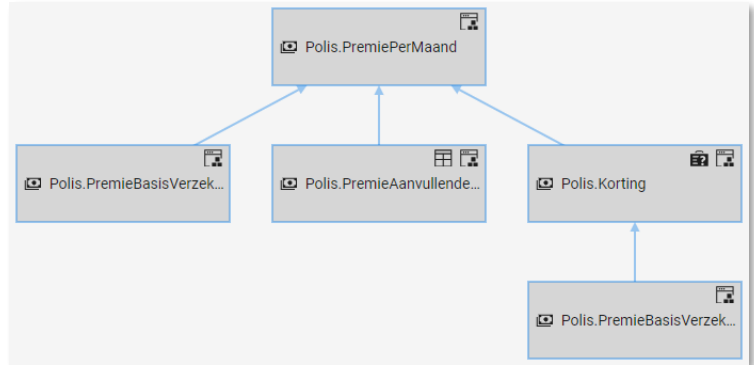
5. Maak een **beslistabel** om te bepalen of iemand een gezonde of ongezonde levensstijl heeft. Je zult twee rijen nodig hebben in deze tabel (eentje met de frequentie van bewegen en eentje met de aantallen uren beweging). Extra rijen toevoegen doe je door helemaal linksboven te gaan staan en daar op de liggende druppel te klikken.

Denk eraan dat de gezondheid **Read only** moet zijn en de invoer-attributen met beweging een **Refresh** krijgen op de pagina! Test je applicatie weer in de Runtime.

6. Blueriq heeft de mogelijkheid om beslissingen te visualiseren met een zogenaamd **Decision Requirements Diagram**. Dat diagram is te zien door op het weegschaaltje te klikken en is te bekijken (en doorklikbaar!) voor elk afleidbaar attribuut, zoals [Polis.PremiePerMaand](#).



Het DRD voor de premie per maand ziet er uit als hiernaast weergegeven. Kun je beredeneren waarom dat klopt? Het diagram is doorklikbaar, dus met Ctrl-klik kun je navigeren naar alle elementen van het diagram en weer terug met het kleiner teken linksboven dat element.



Als je op een rechthoek klikt, klapt hij open voor extra informatie en nog meer navigatiemogelijkheden.

