

Widget en Widget-API

Zoals we in de bovenstaande secties hebben vastgesteld, zijn alle nieuwe elementen van onze oplossingen onderling gerelateerd de tabbladinstellingen. Deze omvat Widgets (Core en Custom). De aangepaste Widgets kunnen ingesteld worden in de sectie “Widgetbeheer”. De eigenschappen kunnen dan verder worden geregeld in de sectie “Widgetinstellingen”. Deze worden vervolgens doorgegeven aan de klant als client-side widget-gegevens.

Bovendien kunnen alle widgets, Core en Custom, onderdeel worden gemaakt van een aangepaste lay-out of een tabblad, die kan worden weergegeven in de nieuwe Applicatie voor Vertegenwoordigers.

Verder publiceren we met de laatste release een Widget-API, die methoden en gebeurtenissen blootlegt voor het ophalen van configuratie-eigenschappen op de klant en ook voor widget-naar-widget communicatie.

Dit wordt gedemonstreerd in de volgende schermafbeelding, die laat zien hoe eigenschappen die in de Beheertool zijn ingesteld, beschikbaar worden gemaakt op voor de klant:

The image shows two side-by-side screenshots. The left screenshot is the 'Widget Configuration' interface. It features a search bar, a table with columns for 'Property', 'Inherit', and 'Value', and a list of widget categories on the left. A red box highlights the 'Property1' row, which has a value of '123'. Below this, the 'uri' field is set to 'https://demo.puzzel.com/dev/'. The right screenshot is the 'External Widget Test' tool. It shows the configuration for an API call, including the widget name 'widget options' and the arguments. The 'Widget Api answer' section displays a JSON object:

```
{  "name": "widget options",  "value": {    "uri": "https://demo.puzzel.com/dev/widgets/external/demo",    "type": "result"  }}
```

 A red box highlights the 'uri' value in the JSON output, which matches the value from the configuration tool. A red arrow points from the configuration tool's 'Property1' value to the JSON output.

Widget API-implementatie

De externe widgets worden over het algemeen gehost op een ander domein en hebben geen directe toegang tot de Widget API. Om de domeinbarrière te omzeilen zal de Core van de applicatie een berichtenkanaal openen met de Messaging API van de browser. In essentie bouwt dit op de `Window.postMessage()` methode, om gegevens te verzenden en te ontvangen.

Meer informatie hierover is beschikbaar op de volgende webpagina: <https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage>

Net als deze: <https://developer.mozilla.org/en-US/docs/Web/API/MessagePort/postMessage>