Catalog Service API - User Guide

Motivation

NoviSign allows customers to present dynamic content on players, either by integrating with NoviSign catalog API, which is responsible for consuming and storing JSON objects in the database. Or player can fetch JSON data directly from any thirdparty API.

Use case

A restaurant wants to display on a screen the orders that are being prepared and the ones that are ready for delivery. The catalog API allows the restaurant POS to send data about current orders details and it will be saved on NoviSign end. The player will fetch the data from the catalog service and will inject data to screen.

Manual have three parts:

1. API documentation.
2. Studio - How to create dynamic content.
3. Players - How to configure API details.

Part 1 - API documentation.

API authentication

For the catalog API before making API calls, client should first authenticate and fetch the Bearer token used for all API calls:

POST https://auth.onsignage.com/auth/realms/{{partner_id}}/protocol/openid-connect/token

Headers:

```
Content-Type: application/x-www-form-urlencoded
```

Body

```
grant_type=password&username={{username}}&password={{password}}&client_id={{client_id}}&client_secret={{client_secret}}
```

- Tokens should be generated and shared by NoviSign team

Catalog API

The full API can be found here:
https://test.novisign.com/catalog/webjars/swagger-ui/index.html?configUrl=%2Fnotifications%2Fv3%2Fapi-docs%2Fswagger-config&urls.primaryName=api

For saving the dynamic data, the following API call should be used:
`POST https://test.novisign.com/catalog/{{account_id}}/items/{{item_id}}`

Example of JSON payload (some "orders" grouped by status):
https://test.novisign.com/catalog/3b9add8b-81ab-48eb-91f6-ad21184e25ba/items/mockup

Part 2 - Studio end

Studio configuration

In order to define a dynamic contentt on the Studio, you will need to user variable expressions inside of the creative's widgets. The dynamic variables will be replaced at run time by the data received from the API.

Supported widgets

Text, label, website, web video, web image.

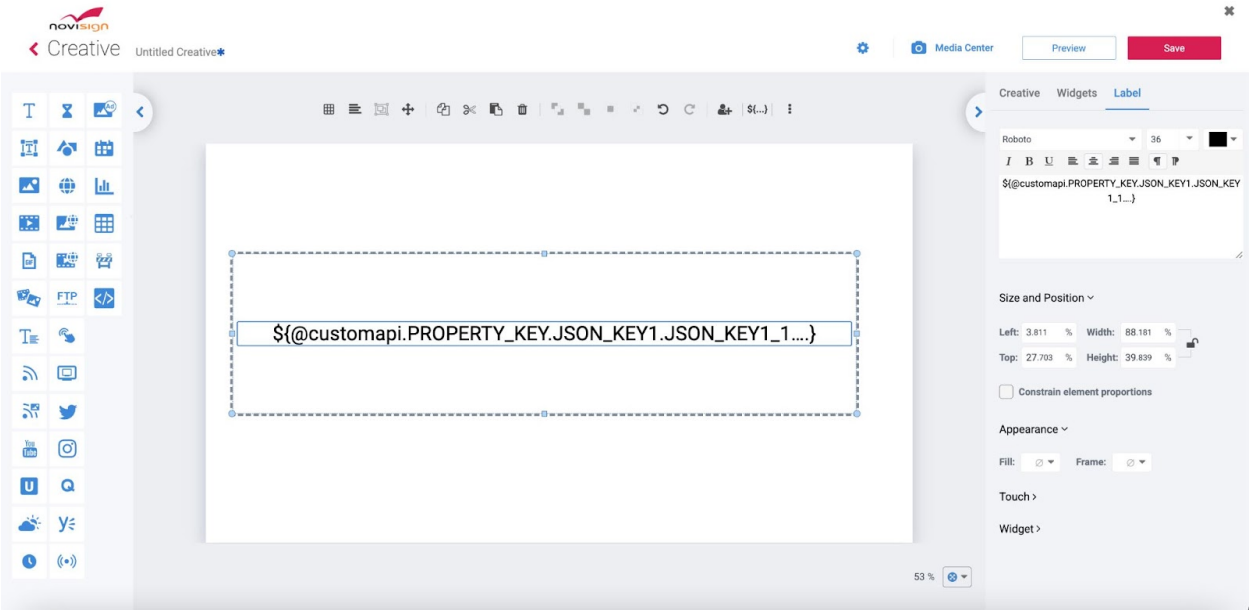
How dynamic expressions looks like - ${}

As mentioned above we support JSON format responses. The JSON will be saved on the player and the expression will refer to the JSON inner path.

${@customapi.PROPERTY_KEY.JSON_KEY1.JSON_KEY1_1….}

@customapi - reserved syntex to refer to JSON
PROPERTY_KEY - each API integration response will be saved under different property key. This prop key will be used to know from where to fetch the data from.
JSON_KEY1,JSON_KEY1_1 - The inner JSON path.

E.g (from mockup JSON ^)
${@customapi.myresturant.InProcess.0.orderIDShort} => “1031”



- After setting the dynamic expression you will see an error message as preview currently not available. Its under development and coming soon. On player it can be presented and shown.

Part 3 - Player

Last phase is to set configuration on the player end to fetch API JSON response data.

Android player

Navigate to: Settings -> advanced section -> external interfaces -> custom API integration
custom API integration is where you add new integrations:

Name - Set the integration name, not used.

API URL - The API direct URL - E.g

https://test.novisign.com/catalog/3b9add8b-81ab-48eb-91f6-ad21184e25ba/items/mockup

Refresh - the interval in seconds the player will call the API - when using NoviSign APIs we ask to set above 30 secs.

Request type - The API method

Property Key - the the key used in the dynamic expression - ${@customapi.PROPERTY_KEY…}

Once all set you can click “TEST” button on the bottom. If clicked player will try to fetch data as set and display the response.

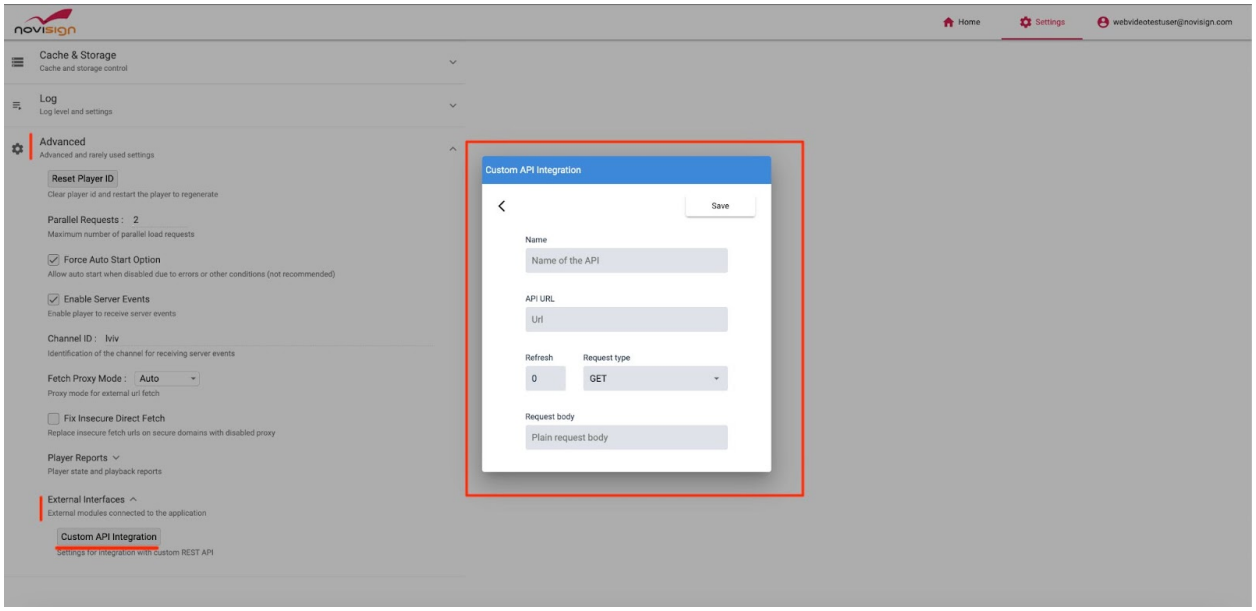
If all is set and you Test and get JSON response, save API.

Last thing is go back to main page and run screen key with dynamic content.

HTML Player

https://app.novisign.com/wplayer/#/pages/settings

Navigate to: Settings -> Advanced -> External interfaces -> Custom API integration



The same configuration as Android player.