BCIO

Branded Content Inventory Optimization

FINAL USER GUIDE

Functional documentation

Introduction to the big three components of the tool:

This tool provides a solution for the creation and management of sponsored content for a final website. The main objective is the injection of sponsored offers on a web into a given space, also including a solution for content management and results analysis.

The injection of offers is based on an analysis of the interests of the user who visits the final website. Depending on the interests registered in the user data, the application will select the offer that best suits those interests and will inject it on the front web page replacing/rotating other offers of the same active campaign.

For these purposes, a cloud system has been designed that provides the content creator a web environment as a backoffice to see campaigns, see the clicks and impressions as the main goals of the associated offers and manage it to adapt to the branded content needs.

This design is divided into three main components:

- Backoffice.
- Recommendation API.
- Module Web (script).

Backoffice:

Backoffice web application is designed to provide to the BCIO content creator a web environment where create and manage sponsored content campaigns with associated offers.

Once a campaign is created and a valid offer is associated, the tool will send the offers to the final page as long as the campaign is still active on dates.

Backoffice also provides a logic to create and manage campaigns and offers. We detail that logic below.

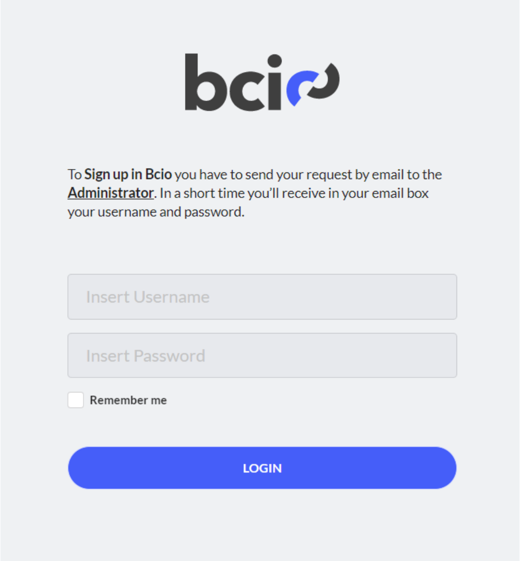
- **Campaign**: Campaign is the main unit of the tool. The user creates a campaign with some dates to start injecting associated offers to the final page.

- **Offers**: A campaign has associated offers. An offer contains all data related to the sponsored content. This is, description, url, kicker data, headline, subtitle, main image, associated tags and offer goal. A campaign can have as many offers as required.

Pages:

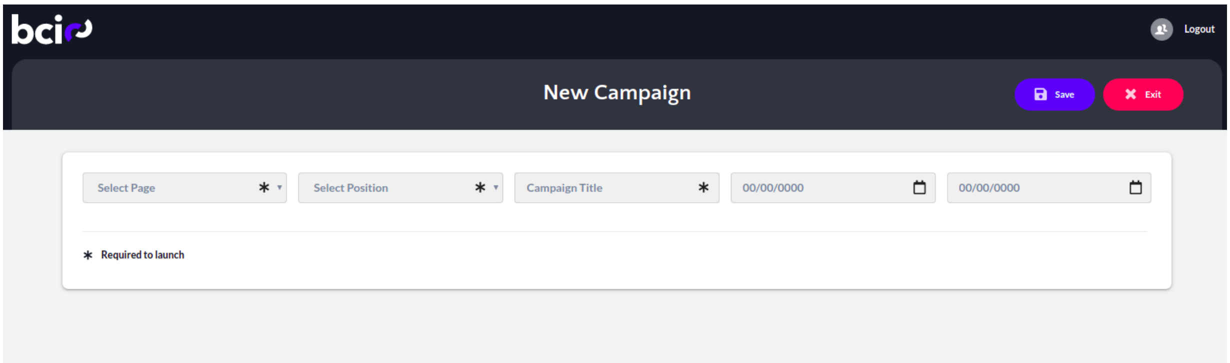
Here is a list of the pages of the backoffice tool.

Login:

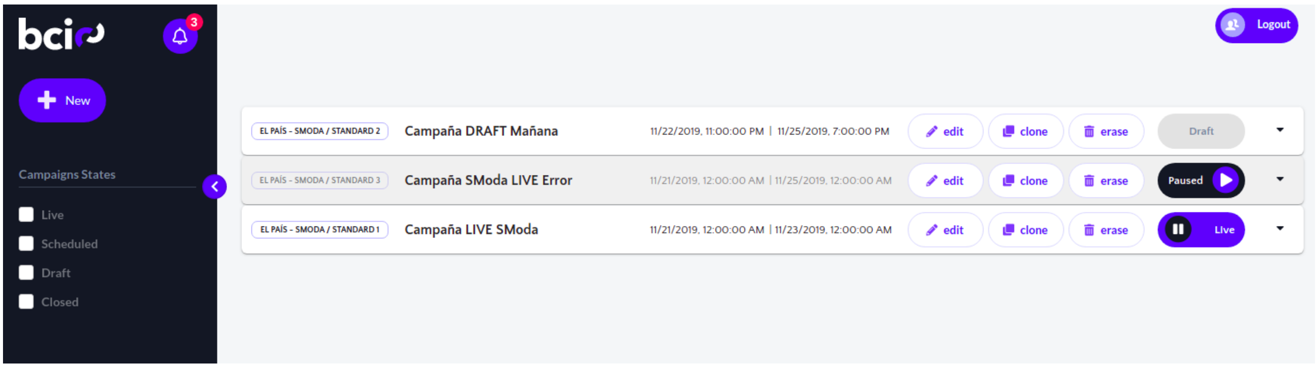


To **Sign up in Bcio** you have to send your request by email to the Administrator. In a short time you'll receive in your email box your username and password.

Create Campaign:



Home (Campaign list page):

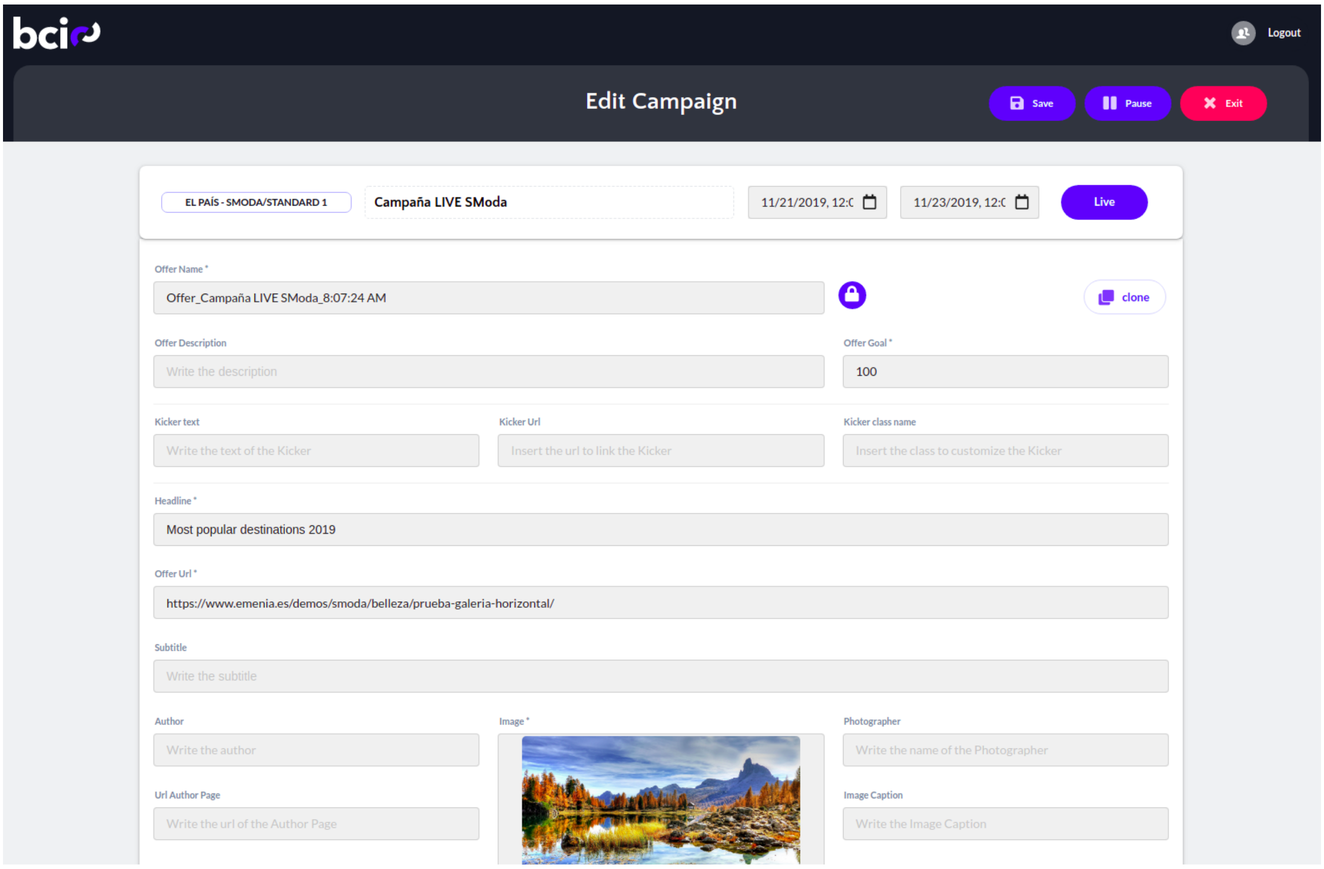


Campaigns States

- □ Live
- □ Scheduled
- □ Draft
- □ Closed

EL PAÍS - SMODA / STANDARD 2	Campaña DRAFT Mañana	11/22/2019, 11:00:00 PM | 11/25/2019, 7:00:00 PM	edit	clone	erase	Draft	▾
EL PAÍS - SMODA / STANDARD 3	Campaña SModa LIVE Error	11/21/2019, 12:00:00 AM | 11/25/2019, 12:00:00 AM	edit	clone	erase	Paused	▾
EL PAÍS - SMODA / STANDARD 1	Campaña LIVE SModa	11/21/2019, 12:00:00 AM | 11/23/2019, 12:00:00 AM	edit	clone	erase	Live	▾

Modify Offer/Campaign:



EL PAÍS - SMODA/STANDARD 1 | Campaña LIVE SModa | 11/21/2019, 12:(| 11/23/2019, 12:(

Offer Name *: Offer_Campaña LIVE SModa_8:07:24 AM

Offer Description: Write the description

Offer Goal *: 100

Kicker text: Write the text of the Kicker

Kicker Url: Insert the url to link the Kicker

Kicker class name: Insert the class to customize the Kicker

Headline *: Most popular destinations 2019

Offer Url *: https://www.emenia.es/demos/smoda/belleza/prueba-galeria-horizontal/

Subtitle: Write the subtitle

Author: Write the author

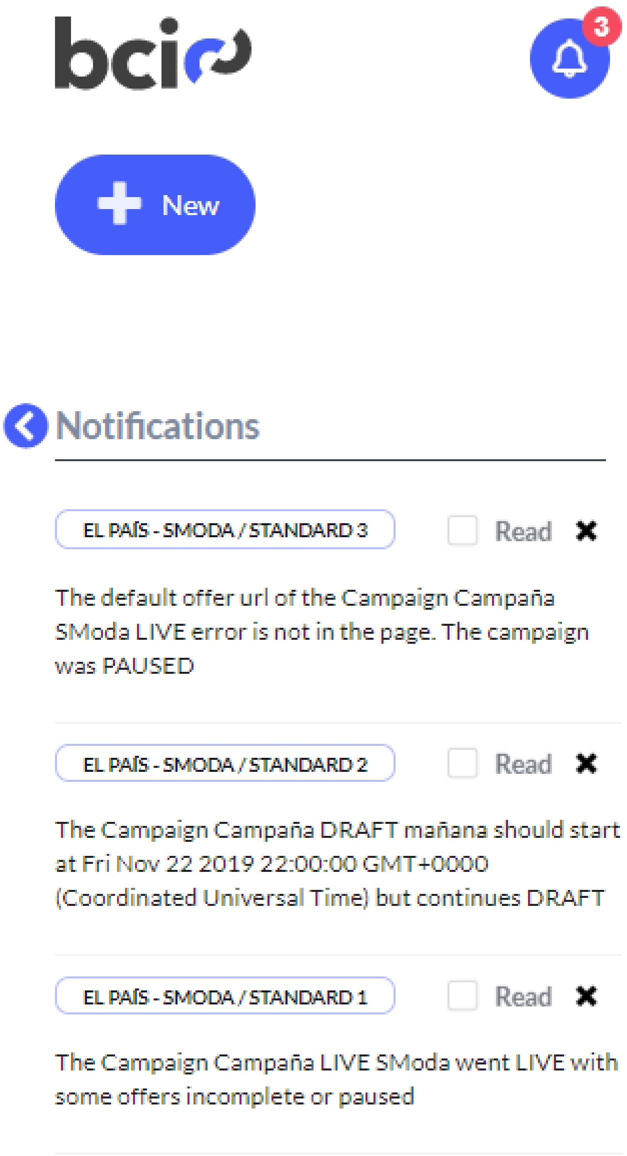
Image *

Photographer: Write the name of the Photographer

Url Author Page: Write the url of the Author Page

Image Caption: Write the Image Caption

Notifications Panel (from Home Page):



EL PAÍS - SMODA / STANDARD 3 □ Read ✖

The default offer url of the Campaign Campaña SModa LIVE error is not in the page. The campaign was PAUSED

EL PAÍS - SMODA / STANDARD 2 □ Read ✖

The Campaign Campaña DRAFT mañana should start at Fri Nov 22 2019 22:00:00 GMT+0000 (Coordinated Universal Time) but continues DRAFT

EL PAÍS - SMODA / STANDARD 1 □ Read ✖

The Campaign Campaña LIVE SModa went LIVE with some offers incomplete or paused

Campaign functional logic:

A campaign is created with mandatory fields and optional dates.

Fields list:

- **Page and Position**:
  Mandatory field. Final page where offers will be injected and the associated position where the offer has to match.
  Page and position values are included via insert into the database. These fields are selected by the main user and set on the installation.

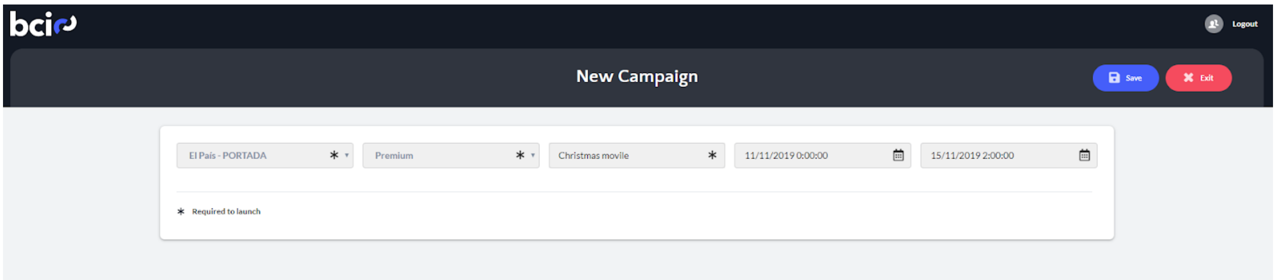
  These fields are designed to apply a control over the final page and its layout of the content of the offers. Page field indicates the identifier of the web page and position its associated template. Example:

  - Page 1 has two associated positions, Premium and Standard.
    This is, each position has one or more associated sizes for the offer image, kicker colors and an html template that will be used to find the html article/class on the final page and replace the original offer with the new one.

    The html template is set on the installation tool as the page and positions are but the template is indicated in a file instead of database. These templates must be prepared for variable substitution. This is, indicate a {{variable}} field related to each field of the offer. This code will be replaced by the value of the offer field and injected as plain html on the final page and replacing the initial offer.

- **Campaign title:**
  Mandatory field filled by the campaign creator. Indicates the title/description of the campaign.

- **Dates. Start and End:**
  Optional fields. Determines start and end of the campaign. Campaigns can be created without these two fields and be set in next steps but then they will be mandatory to launch.



New Campaign

El País - PORTADA * | Premium * | Christmas movile * | 11/11/2019 0:00:00 | 15/11/2019 2:00:00

* Required to launch

Campaign states:

Depending on dates, associated offers status or user decisions campaigns have a state/status attribute which determines its activity. States can be *DRAFT*, *LIVE*, *PAUSED*, *SCHEDULED* and *CLOSED* and campaign behavior varies depending on their status.

- **DRAFT**

  Innital and default status of a campaign of new creation. In this states the campaign is not able to be launched because of missing or incomplete default associated offer. Will stay as DRAFT as long as the user wants or until the default offer is filled and launch the campaign.

- **LIVE**

  Main state of a campaign. In this state a campaign has at least a valid default offer and an active/current date. A campaign passes to LIVE status when it gets saved and the user clicks on the 'Launch' button. Then data is saved, validated and status is marked as LIVE because of current dates.

  At this point the system is running and sending the offers to the page and position of the campaign.

- **SCHEDULED**

  Same behavior as LIVE state but at this point campaign is launched but its start date has not been reached. Campaign will pass to LIVE automatically when its start date arrives.

- **PAUSED**

  Only LIVE or SCHEDULED campaigns can be paused. This campaign status prevents offers from being sent in case of LIVE campaign or pass to LIVE from SCHEDULED ones.

  A paused campaign can be resumed but this state implies certain controls:

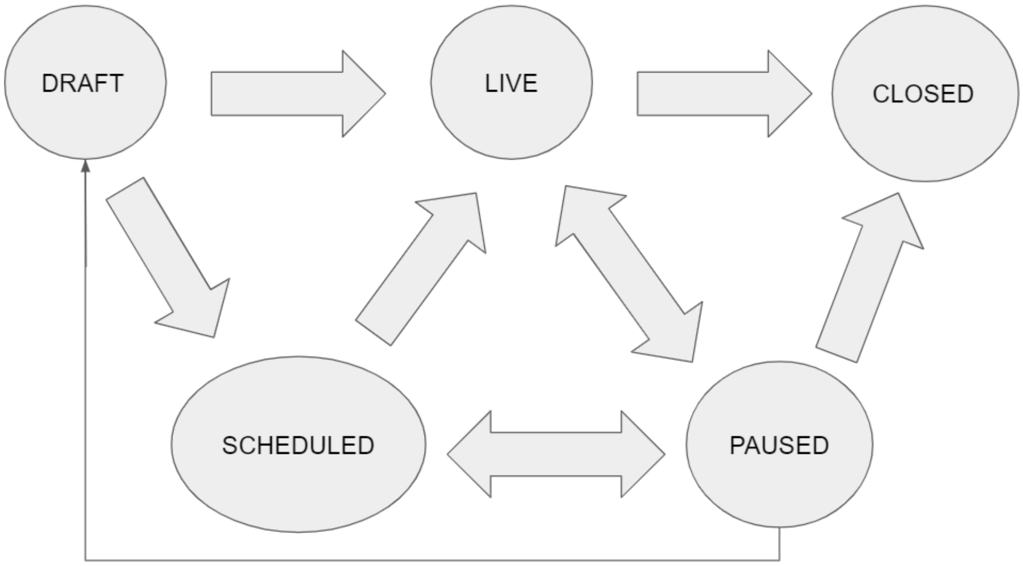
  - A campaign can only be resumed by the user.
  - When the user resumes a paused campaign, data is validated and the campaign passes to LIVE or SCHEDULED. Default offers must be valid to resume a campaign but if there is some other offer with invalid data the campaign will resume and the invalid offer will not be injected.
  - A paused campaign can not be resumed if its dates overlaps with some LIVE or SCHEDULED campaigns.
  - If a paused campaign exceeds its end date automatically passes to CLOSED.

  To pause a campaign click on Pause toggle on the campaign list page or on the Pause button on modify campaign page (replaces Launch button).

- **CLOSED**

  Final state. When a campaign reaches its end date automatically its marked as closed and users cannot launch or modify it anymore. Its campaign and offer data can be consulted but not modified.

Campaign state diagram:



Campaign state diagram explanation:

From	To	Operative
DRAFT	LIVE	Campaign created, saved and launched.
DRAFT	SCHEDULED	Campaign created, saved and launched if its beginning is in some date after this very moment.
LIVE	CLOSED	Campaign which has reached its end date.
LIVE	PAUSED	Campaign gets deactivated and will send no offers till resumed. Even if its end date is overpassed paused status remains active.
PAUSED	LIVE	After clicking the resume button data gets validated and campaign is relaunched.
PAUSED	SCHEDULED	If a paused campaign gets resumed. Data is validated and status gets updated as scheduled due to dates.
PAUSED	DRAFT	Special transition in case of a scheduled campaign tries to pass to live due to dates but its default offer is paused.
SCHEDULED	LIVE	Reached start date on a valid campaign. This change is automatic made by a background process.
SCHEDULED	PAUSED	Clicking on the pause button in a scheduled campaign changes its status and the campaign gets freezed.
PAUSED	CLOSED	Reaches end date on a paused campaign.

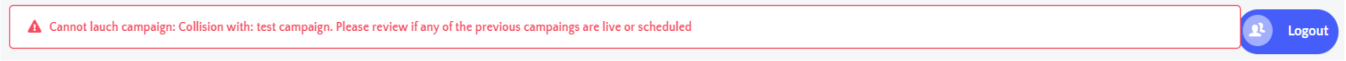
Campaign filter by status:

On the home page there is an option to filter the list of campaigns by it's status. Paused campaigns will be shown with the LIVE check even if the resumed status is LIVE or SCHEDULED.

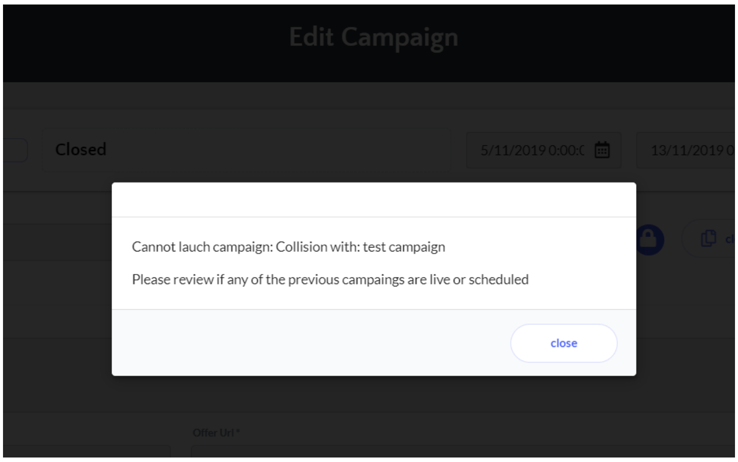
Collisions on campaigns:

Two campaigns are in collision when it's dates overlap each other and it's final page and position are the same. This means a live campaign can not be injecting offers to the same page and same position at the same time as another campaign.

To handle this behaviour backoffice will not allow to launch/resume a campaign when a collision exists with another LIVE/SCHEDULED campaign.
When this happens an error/warning message will appear on the top of the page or a popup indicating the overlap.



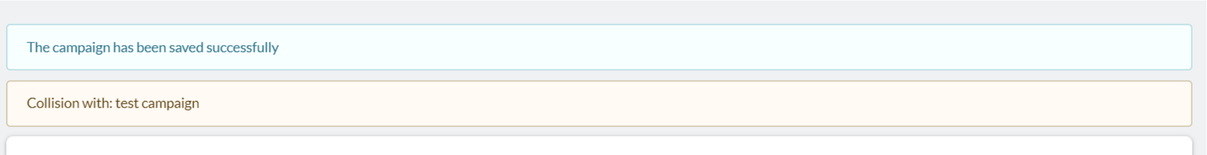
Cannot lauch campaign: Collision with: test campaign. Please review if any of the previous campaings are live or scheduled



Cannot lauch campaign: Collision with: test campaign

Please review if any of the previous campaings are live or scheduled

Warning messages will always appear indicating the conflicting campaign when saving data if two campaigns share page, position and dates interval even if the collisioned campaign is not in LIVE/SCHEDULED status.



The campaign has been saved successfully

Collision with: test campaign

Notifications:

The notification system will notify the user of certain situations related to a campaign that requires their attention. This system is designed following these basic features:

- Notifications will be independent per user. Each user will have all notifications and will mark them as read or deleted as they consider but that will not affect other users, who will continue to have those notifications on their panel.
- Clicking the notifications button at the homepage will display a side panel with the list. There will be a Read button to mark as read and an X in to clear the notification.
- A limited number of notifications will be displayed. Suppose there are 100 unread notifications, as a maximum of X will be displayed (eg 30) that will appear scrolling. No paging or loading of previous notifications.

There are three types of notifications implemented:

- Notification in the app when a campaign pass to live and some offer is incomplete:
  - The system detects that a campaign will pass to live (from SCHEDULED or PAUSED) and it has invalid or paused offers (not the default one).
- Notification in the app when the start date is near and the campaign is in DRAFT state:
  - The system detects that a draft campaign has saved dates and the start date will be soon.
- Notification when the default offer is retirated of the cover page (from the widget):
  - The system cannot find the default offer of the page to inject the offers of the campaign.
  - The campaign will be set as PAUSED.

Offer functional logic:

An offer can be created with mandatory and optional fields. An offer is the single branded content unit which belongs to a single campaign and represents the content that will be injected into the page and position of its father campaign indicates.
On modify campaign page campaign basic data can be updated. On the same page each offer data is loaded as a section of the page and can be modified too. By default the first offer is set as default offer. This can be changed by switching on the lock icon of the offer.

Fields list:

- **Offer Name:**
  Auto generated by default when a user creates a campaign or when an offer is cloned.
- **Offer Description:** Optional and 20-250 characters long.
- **Brand Name:** Name of the branded content company.
- **Offer Url:**
  Mandatory. Url of the default offer of the destination page. BCIO tool will use this url to find the offer into the html code of the final page. Then will replace the original offer with the new one injecting its data and the html template.
- **Headline:** Mandatory. Main title input.
- **Subtitle:** Subtitle input.
- **Kicker Url:** Associated url of the kicker.
- **Kicker class name:** Name of the .class associated with the kicker style. On the final page this class will be styled with css.
- **Kicker text:** This text will be shown on the kicker image.
- **Offer Goal:**
  Mandatory. Number of clicks marked as offer main goal. If some offer goal is reached, the offer will not be a priority and the system will show other sibling offer. If every offer has reached the goal but campaign remains as live, the system will send a random offer.
- **Image:**
  Mandatory. Main image of the offer content. Must be loaded with every size that campaign position marks on the load modal.
- **Tags:**
  Mandatory. Documentation or segmentation tags. It's a way of classifying the offer. Tags must be loaded in the installation inserting it into the database.
- **Author:** Author of the linked content.
- **URL Author Page:** Link to portfolio/personal space of the author.
- **Photographer:** Author of the offer image.
- **Image Caption:** Text associated to the image.
- **Image Copyright:** Copyright of the image if any.

Offer states:

Offers have its own status attribute. There are three states and each one determines if the offer is valid or ready to be sent.

- **DRAFT**

  Initial state of every offer. Once an offer is created it is in draft state when it's required data is empty. When the user fills the required data with valid information (p.e url field needs to be valid via GET call) the offer passes to LIVE even of the state of it's campaign. An offer in DRAFT state will never be sent to the final page.

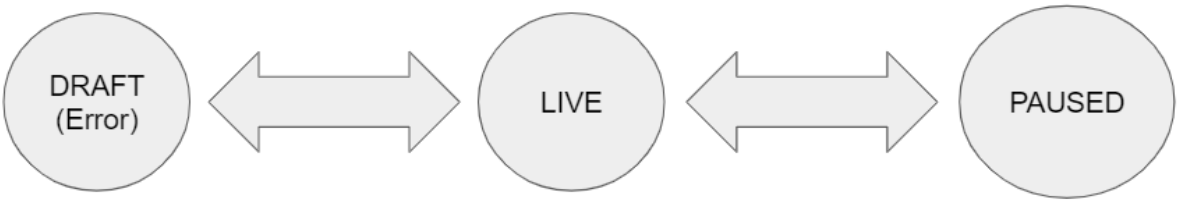
- **LIVE**

  An offer is in live state when every mandatory field is filled, valid and saved independently of the campaign status.

- **PAUSED**

  Only a LIVE offer can be paused. When an offer is in PAUSED status it will not be sent to the final page on LIVE campaign.
  Campaigns default offer can not be paused if campaign is in LIVE status. If the campaign is in SCHEDULED status it's default offer can be paused but when campaign reaches the start date will pass to DRAFT if default offer is still as paused.

Offer state diagram:



From	To	Operative
DRAFT	LIVE	Required offer data filled, validated and saved.
LIVE	DRAFT	Saved data has some invalid (empty mandatory) inputs. Shown as ERROR on a live campaign list. If the user fills a field of a live offer with invalid data its offer won't be saved but if the mandatory field is empty theres is no validation and the state passes to DRAFT
LIVE	PAUSED	Offer paused and not running into the campaign.
PAUSED	LIVE	Offer reactivated after successful offer data validation.

Recommendation API

The recommendation api is the module of the solution that serve the content to the widget. In the output side, this service will provide the required information of the campaigns and the information of the user to choose the best offer for each user. In the input side the widget will send events and errors to this API.

Endpoints

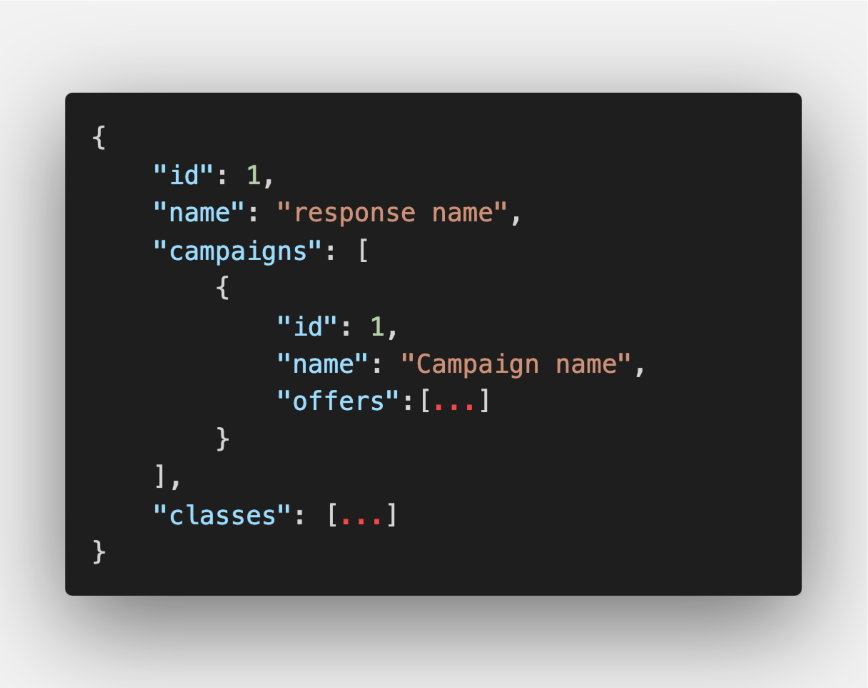
- **Get active campaigns by page.** This service is used when the webpage is loaded, the widget will request the active campaigns with offers and the templates. This information is read from the redis database.
- **Get user custom data.** When the widget is found a known user will make a request of the custom order for the offers for that user. This information is also in the redis database and is loaded by the data team.
- **New impression (post event).** Each impression of any offer by the widget will send an event to the recommendation API. This event will go to BigQuery. And will use to update the impression count of each offer by the back office.
- **New click (post event).** Each click in any offer by the widget will send an event to the recommendation API. This event will go to BigQuery. And will use to update the click count of each offer by the back office. The clicks amount reach the goal defined for the offer the widget will stop to impress that offer.
- **New error.** If the widget can not find the default offer's url in the webpage will send an error to the recommendation api. The back office will check the errors of each campaign. If one campaign has 10 or more errors by day will notificate the user and pause the campaign.

Widget Script (Module Web)

The Bcio widget is one of the two software parts which allows to rate posted/published ads and replace them accordingly.

How does it work?

Once the widget is initialized, it will make a request to the api URL passed in its params, getting all the ads available for the pageID set.



```
{
    "id": 1,
    "name": "response name",
    "campaigns": [
        {
            "id": 1,
            "name": "Campaign name",
            "offers":[...]
        }
    ],
    "classes": [...]
}
```

Right after it gets the response mentioned above it will check if the user has visited the site previously checking a cookie that was set before in order to make another request asking which ads are more relatable for this user.

In this cookie evaluation there are several conditionals to contemplate:

1. If a cookie is found the user goes to a Smart Assignation process:
    a. It evaluates the first offer obtained, since they are return in order the first one would be always the more related to the user, if the first offer its completed (has already achieved its goal) or there is not user data stored the widget will render a random offer:

    b. If the user data is received successfully and the offers has not achieved its goal the widget will render the first more related to the user.

2. On the contrary, if a cookie has not been set previously, it evaluates if the user has the first visit cookie and when it was set.

a. If the cookie's date is the day before the widget will save the cookie and send the user to the smartAssignation process explained before.
b. If the cookie has been set the same day the user will go to the fastAssignation process. Where if all offers have achieved their goal the widget will render a random offer.

On the contrary, if there are incomplete offers it will use the compensation Index configured to show a random offer or the one that needs more clicks to reach its goal.

There are several steps to render the selected offers.

The next step is filtering the offers which have been gotten in the first request and match which one with the most relatable in the second request. Once it gets the offer for a campaign it will look into the page for the default offer through its url, once it finds it, it will replace it with the offer selected, getting first all the offer values to be rendered, like image and template.

Finally, to render the offer it will match all the templates params and replace them in the corresponding spot: