Kinto
DataLayer & tagging specifications



V1.0 - 05/04/2021

TABLE OF CONTENT

About this document **4**

Tag Manager script implementation **4**

Overview 4

Adding Google Tag Manager code 5

DataLayer overview **6**

Event triggers **7**

DataLayer generic specifications **8**

General overview 8

General rules for object names & values 9

Page object 9

Event object 11

Product object 12

Form object 12

DataLayer detailed specifications **13**

Page templates 13

Components 13

Support **15**

HISTORY CHANGE

Date	Description	Who
05/03/2021	**Initial version of the dataLayer and tagging specifications**	Christophe Hendrick (FlyingFish)
26/03/2021	**dataLayer finalisation (compliance with AEM - dataLayer)**	Amaury Dessy (FlyingFish)
05/04/2021	**Added form iframe case. Same domain and cross domain**	Christophe Hendrick (FlyingFish)

1. About this document

This document describes the technical details of the dataLayer and tagging mechanisms to be implemented in the new Kinto Platform.

These elements will allow to measure user interactions and behaviour on the new sites.

The dataLayer and tagging mechanism will be used not only to feed digital analytics platforms (Google Analytics but also possible Adobe Analytics) but also to possibly implement other tools such as Usabilla, Content Square and more.

The document is intended for development teams and it covers the following topics:

- **Tag Manager script implementation:** details about the tag manager script to be added in all pages and where it should be located
- **DataLayer overview:** principle of the dataLayer and high level view on its structure
- **Event triggers:** how events will used to track specific user interactions and how these will be made
- **DataLayer detailed specifications:** detailed description of the various objects that compose the dataLayer
- **Detailed content tagging specifications:** rules & values of the dataLayer objects and events for all content elements (pages, components, tools,...)

This document will be elaborated progressively as new elements will be added in dataLayer detailed specifications and content tagging specifications.

2. Tag Manager script implementation

Overview

Google Tag Manager 360 will be used as the primary tag manager platform for the Kinto sites.

The tag manager will be used:

- To deploy and manage globally (i.e. across all sites) the primary analytics solution: Google Analytics 360 and other Kinto pan-EU tools.

  These elements will be managed through the Kinto **Global container** i.e container that is unique and physically added in all pages of all AEM sites.

  .

Global variables and triggers will be also defined in the global container and exported to local containers at creation time.

- To deploy and manage local tags and scripts: advertising tags, social media tags, local tools (surveys, local analytics platform,...).

  There will be one **local container** by market. These local containers will be injected dynamically by the Kinto global container using the Google Tag Manager zone functionality.

  These containers will be managed by NMSC's users and local agency users.

Working with separated containers allows to facilitate proper governance and access management across the TME and NMSC organization.

Adding Google Tag Manager code

The following code must be present in all pages of the Kinto sites that are created in the AEM platform. This must be automatic.

Only one GTM script will be present "physically" in the page and injected directly by the CMS - this will be the Kinto **global snippet** that is used to inject TME global analytics tag and other global scripts.

It is the global container that will inject the local containers via the zone feature to allow local markets / agencies to integrate advertising tags.

The below code must be added in the <HEAD> section of the page if possible or as high as possible in the <BODY> in order to guarantee better data precision.

```
<!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.s
tart':
new Date().getTime(),event:'gtm.js'});var
f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=tr
ue;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentN
ode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-NCJ8QMB');</scr
ipt>
<!-- End Google Tag Manager -->
```

3. DataLayer overview

- The Data layer is a Javascript object where all contextual information will be set including content and product information, user attributes, event description and more
- It is a global information object that will allow the tag manager to retrieve the required parameters and to send it to the analytics tools and other platforms
- The data layer **must be set initially on page load** (page & context information) and it will get updated whenever a tracked event occurs.
- It must be placed in the HTML code **BEFORE** the tag manager snippet so all contextual information is available when the tag manager code is initialized.
- Here is a simplified example of the data layer object for illustrative purpose

```
<script>
window.tngaDataLayer = {

     "page": {
          "category": "service",
          "subCategory": "overview",
          "sectionlv1": "service",
          "sectionlv2": "kinto-share",
          "sectionlv3": "",
          "language": "it",
          "country": "it"
          },

     "product": {
          "category": "mobility",
          "subCategory": "share"
      },

     "event": {
          "name": "formevent",
          "componentName": "contact",
          "componentGroup": "forms",
          "action": "forms-step",
          "label": "form-success"
     },
    …
}
</script>
```

For detailed specifications of the various objects and their attributes, please see section 5.

Important note:

- The data layer is based on “standard” Javascript as currently implemented in Toyota sites. We don’t propose to use the Google Tag Manager dataLayer object to keep the implementation “universal” i.e. it can be used by other applications or tag manager systems (e.g. Adobe Launch).

4. Event triggers

- Some key user interactions within a page (e.g. click on a CTA) will need to be tracked. This will be done using the data layer to provide contextual information about the event (action, component being clicked, item value,...) and event triggers to inform the tag manager

- Basically, whenever an action that needs to be tracked (list of tracked events are described in the component detailed specifications), the following will happen
    - The **tngaDataLayer object will be updated** with event related properties
    - A **custom event trigger** will be made after the data layer update. The value of the trigger (= event name) will vary depending on the type of event (see detailed component specifications)

- The tag manager event listener will detect the event, re-evaluate the data layer state and update corresponding variables. It will also evaluate any triggers / firing rules to decide if some tags need to be fired.

- The mechanism is illustrated in the below example: in this example, the user clicks on a CTA pointing to a contact form.

```
window.tngaDataLayer = {

     "page": {
          // page attributes
          },

     "product": {
          // product attributes
               },

     "event": {
          "name": "ctaevent",
          "componentName": "cta",
          "componentGroup": "cta",
```

```
            "action": "click-cta",
            "label": "dealerfinder-toyota"
      },
    …
}
```

After updating the event object (in red), the following trigger is executed

```
dataLayer.push({'event': 'ctaevent'});
```

- When a tracked event occurs, it is **important to maintain the state of the data layer object** i.e.
    - new objects related to the event (e.g. the tngaDataLayer.event object) are added (if not yet presented) or updated if already present in the DataLayer.
    - Other contextual objects (e.g. the page object) that are not affected by the event must remain in the dataLayer and must not be altered.

5. DataLayer generic specifications

This section is describing all the objects that can be present in the data layer object depending on the page template, user actions, tools or forms being used, etc.

Important: The global structure is work in progress and will be refined / enriched as detailed guidelines are provided.

General overview

Here is a global overview of the different properties & attributes that may be present in the data layer.

```
<script>
window.tngaDataLayer = {

      "page": {
            "category": "$pageCategory",
            "subCategory": "$pageSubCategory",
            "sectionLv1": "$p1stLevelSection",
            "sectionLv2": "$2ndLevelSection",
            "sectionlv3": "$3rdLevelSection",
            "language": "$languageISOCode",
            "country": "$countryISOCode"
            },

      "event": {
```

```
            "name": "$eventName",
            "componentName": "$componentName",
            "componentGroup": "$componentGroup",
            "action": "$eventAction",
            "label": "$eventLabel",
      },

      "product": {
            "category": "$productCategory",
            "subCategory": "$kintoProductSubCategory",
                  },

      "form": {
            "name": "$formName",
            "step": "$formStep",
            "conversion": "$formConversion",

      },
    …
}
</script>
```

General rules for object names & values

- All object & attribute names must be in *lower camel case.* No space or separator should be used between words.
    - OK: subCategory, productInfo
    - NOK: sub-Category, product_info
- All string / text values should be
    - lowercase by default except if explicitly mentioned.
    - By default, space should be replaced by “-” (dash)
- All currency / float values should apply the following format i.e.
    - No thousand separator
    - Decimal must be represented by a “.” (dot) as in American format
    - For currency value, no currency symbol

Page object

The page object contains detailed information about the page being viewed such as the page template name, category and subcategory as well as information about its position in the site structure.

This will enable reporting by content type, primary sections and other.

Object	tngaDataLayer.**page**
Scope	Page - set on page load and should not be modified afterward

Attribute	*Description*	*Values / examples*
`category`	Mandatory Main template category	homepage promotions content products services ...
`subCategory`	Optional - depends on page template Template sub-type	overview detail-page article ...
`sectionLv1`	Mandatory English value based on the primary level of the sitemap / url	homepage about-us news jobs support
`sectionLv2`	Optional Correspond to the 2nd level of the sitemap / url	
`sectionLv3`	Optional Correspond to the 3rd level of the sitemap / url	
`language`	Mandatory. 2-letter ISO code of the content language *Note: some countries like BE, CH, EE, LT, LV support multiple languages.*	en de fr nl it ...
`country`	Mandatory. 2-letter ISO code of the content country	de fr it no ...

Event object

The event object contains event related details such as the component from which the event was triggered, the action performed, the related element,...

This will enable user interactions reporting within pages, to understand what components are used and what features in particular.

Object	tngaDataLayer.**event**
Scope	Event. Object is added or updated whenever a tracked event is taking place. Additional objects may be added or updated depending on the event type.

Attribute	***Description***	***Values / examples***
`name`	Mandatory The type of event: CTA, download, general component click,... See component detailed specifications	ctaevent downloadevent componentevent formevent ... .
`componentName`	Mandatory. Name of the component that generate the event	cta contact newsletter …
`componentGroup`	Mandatory Corresponding group of the component See component detailed specifications	cta forms
`action`	Mandatory. Description of the action performed See component detailed specifications	click-cta form-step ...
`label`	Mandatory. Description / contextual information about the performed action e.g. link being click, target tool of a CTA button, filename of the file being downloaded See component detailed specifications	E.g. URL of link being clicked Target tool of a CTA button Filename of the file being downloaded ...

Product object

The product object will contain information about the Kinto service being viewed or linked to the interaction made by the user.

Object	tngaDataLayer.**product**
Scope	Page and event. Object array is added on page load if applicable and can be updated / enriched on specific events.

Attribute	***Description***	***Values / examples***
`category`	Always equal to “mobility”.	
`subCategory`	The type of kinto service related to the page / interaction.	E.g: share one join flex go ride

Form object

The form object applies specifically for forms and may evolve over time. It contains information such as the form name, the step name or description, a conversion flag,...

Note: If the form is embedded in an Iframe, please read the **Forms embedded in Iframes** section.

It is present in forms.

Object	tngaDataLayer.**form**
Scope	Page or event. Object is added or updated on page load (if page level) or on specific events.

Attribute	***Description***	***Values / examples***
`name`	Mandatory Name of the form. Lead generation form name will be prefixed by “lead-”	E.g. contact lead-offer-request ...

`step`	Mandatory. Description or title of the step of the form (e.g. start, contact-details,...) = “form-success” when form is successfully submitted	E.g. contact-details form-success …
`conversion`	Mandatory Indicates if a conversion occurred (=1) or not (=0, default)	0 | 1

6. DataLayer detailed specifications

You will find a detailed view of the expected dataLayer values per type of interaction in the following tagging spreadsheet

Page templates

Tagging specifications for the general page object can be found in the tagging spreadsheet under the “Page object” tab.

Definitions & examples of variable values are also reminded in the “Value dictionary” tab.

Components

Tagging specifications for components and events can be found in the tagging spreadsheet under the “Components & events” tab.

Definitions & examples of variable values are also reminded in the “Value dictionary” tab.

7. Forms embedded in Iframes

The requirement depend the domain under which the iframe is located:

The source page and the form iframe are located under the same domain

The form frame must contain the global GTM snippet (GTM-NCJ8QMB)

The source page and the form iframe are located under the different domains

Note: This requirement is not compatible with Onetrust (or similar consent management system) since there is no page reload after consent.
In that case the Iframe form must be located under the same hostname as the main page.

The form frame must contain the global GTM snippet (GTM-NCJ8QMB)

In order to bind sessions across domains, we have to pass the Google Analytics clientId to the Iframe url. When the Iframe is loaded we must append the Google Analytics clientId to the original Iframe URL.

Here is the process:

1. Check the presence of the Google Analytics cookie in the source page
    a. Cookie name =" _ga"
2. If exist, extract its value and parse it to keep only the clientId part:
    a. Cookie value example: GA1.2.1621912702.1612898204
    b. The script will extract the part after the 2nd "."
    c. The clientId in this case is: 1621912702.1612898204
3. Load the form Iframe and append the clientId as a parameter of the page.
    a. https://www.toyota-france.fr/kinto/kinto-one.asp becomes then https://www.toyota-france.fr/kinto/kinto-one.asp?gacid=1621912702.1612898204

This id helps us to pass the clientId over domains, and maintain the session active.

8. Support

For any question or remark about this document, please contact=

- Christophe Hendrick (Lead Digital Strategist and Analyst, Flying Fish): christophe.hendrick@flyingfish.agency or
- Amaury Dessy (Digital Analyst, Flying Fish): amaury.dessy@flyingfish.agency

----End of Document----