



Data Layer Documentation

Overview

This document provides a comprehensive outline for implementing a data layer on www.kingston.gov.uk using Google Tag Manager (GTM) for the new Drupal website.

The data layer is critical to enabling the tracking of user interactions, as it is a hard coded method of tracking that is better than what is currently implemented via event tracking. By implementing a consistent data layer, we're ensuring that insights are systematically gathered across various user interactions without the need to attach events to elements within the DOM.

More information about what data layers are and how they work can be found here:

<https://developers.google.com/tag-platform/tag-manager/datalayer>

There is also this module for drupal based sites:

<https://www.drupal.org/project/datalayer>

Implementation Approach

- **Google Tag Manager (GTM)** will be used to deploy tracking scripts, manage event triggers, and ensure that all tracking configurations are consistent.
- A **JavaScript-based data layer** (**`window.dataLayer`**) is used to store structured data about user interactions dynamically.



Data Layer Documentation	1
Overview	1
Implementation Approach	1
Page-Level Data Layer	3
Implementation:	3
Events Triggering Page-Level Data Layer	3
Navigation Events	4
Button Clicks	4
Mega Menu Navigation Interactions	4
Task Clicks	4
Breadcrumb Link Click	5
Click Table of Contents	5
Search Interaction Tracking	6
Main Site Search	6
Click Open/Close Filters	6
Refine Search Filter	6
Click Search Results	7
Zero Search Results	7
Specific Link Clicks	9
File Downloads	9
Language Changes	9
Print / Share Content	9
Video / Audio Start	10
Video / Audio Stop	10
Contact Us Click	11
Email Click	11
External Link Click	11
Non-interaction & timed events	13
404 Event	13
Scroll Depth Tracking	13



Page-Level Data Layer

Each page should include a structured data layer object that provides essential details about the page, such as its title, URL, category, and user status. This helps in creating standardized reports across different sections of the website.

Implementation:

JavaScript

```
window.dataLayer = window.dataLayer || [];  
window.dataLayer.push({  
  'event': 'page_view',  
  'page_title': '{{Page Title}}',  
  'page_path': '{{Page URL}}',  
  'page_category': '{{Page Category}}',  
  'content_type': '{{News/Article/Directory etc}}',  
  'page_author': '{{Author}}',  
  'first_published': '{{Original Publish date}}',  
  'last_updated': '{{Last Modified date}}',  
  'robots_tag': '{{allowed/disallowed}}'  
});
```

Events Triggering Page-Level Data Layer

- **Page Load:** Each page load should fire a `page_view` event



Navigation Events

Button Clicks

Purpose: Monitor interactions with all key buttons across the site to track engagement.

Trigger: Fires when a user clicks any button element on site.

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'button_click',
  'button_text': '{{Click Text}}',
  'button_id': '{{Click ID}}',
});
```

Mega Menu Navigation Interactions

Purpose: Capture interactions with menu navigation and breadcrumbs to understand user navigation behavior.

Trigger: Fires when a user navigates with the mega menu

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'navigation_click',
  'menu_item': '{{Click Text}}',
  'menu_id': '{{Click ID}}',
  'menu_level': '{{Menu Depth Level}}',
});
```

Task Clicks

Purpose: Track interactions with key tasks or services links on the website.

Trigger: Fires when a user clicks on a task type button on the main homepage, or sub pages

Implementation:



JavaScript

```
window.dataLayer.push({  
  'event': 'task_click' // or 'sub_task_click',  
  'task_name': '{{Task Name}}',  
  'page_path': '{{Page URL}}'  
});
```

Breadcrumb Link Click

Purpose: Monitor clicks on breadcrumb navigation links.

Trigger: When a user clicks on a breadcrumb link within the page as a form of navigation

Implementation:

JavaScript

```
window.dataLayer.push({  
  'event': 'breadcrumb_click',  
  'breadcrumb_url': '{{Breadcrumb URL}}'  
  'breadcrumb_item': '{{Breadcrumb Name}}',  
  'full_breadcrumb': '{{Entire Breadcrumb}}'  
});
```

Click Table of Contents

Purpose: Track when users click on a Table of Contents link.

Trigger: When a user clicks on a table of contents item within a guide page

Implementation:

JavaScript

```
window.dataLayer.push({  
  'event': 'contents_table_click',  
  'toc_item': '{{Table of Contents Item}}',  
  'toc_position': '{{Number}}',  
  'page_path': '{{Page URL}}'  
});
```



Search Interaction Tracking

Main Site Search

Purpose: Capture search queries entered into the primary search bar and their corresponding results.

Trigger: Fires when a user submits a query in the main site search bar, or if the search reloads post filter refinement or application

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'main_site_search',
  'search_term': '{{Search Term}}',
  'results_count': '{{Results Count}}',
  'page_path': '{{Page URL}}'
});
```

Click Open/Close Filters

Purpose: Track when users open or close search filters.

Trigger: Fires when a user opens or closes a filter menu in search

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'click_search_filters',
  'filter_name': '{{Filter Name}}',
  'action': '{{Open/Close}}',
  'search_term': '{{Search Term}}',
});
```

Refine Search Filter

Purpose: Track when users refine their search query after receiving results.

Implementation: Track when a user filters actively using the filter menu



JavaScript

```
window.dataLayer.push({
  'event': 'refine_search',
  'search_term': '{{Search Term}}',
  'page_path': '{{Page URL}}'
  'filter_name': '{{Filter Name}}',
  'refinement': '{{Refinement Name}}',

});
```

Click Search Results

Purpose: Monitor which search results users click to assess relevance and engagement.

Trigger: Fire when a user clicks any of the results within the search page

Implementation:

JavaScript

```
window.dataLayer.push({
  'event': 'click_search_results',
  'search_term': '{{Search Term}}',
  'result_url': '{{Clicked Result URL}}',
  'result_text': '{{Clicked Result Text}}',
  'result_number' : '{{Rank of item}}'
});
```

Zero Search Results

Purpose: Identify searches that returned no results to improve content and search functionality.

Trigger: Fires when a user receives a search that has zero results

Implementation:

JavaScript

```
window.dataLayer.push({
  'event': 'zero_search_results',
  'search_term': '{{Search Term}}',
});
```





Specific Link Clicks

File Downloads

Purpose: Track when users download PDFs, Excel files, Word documents, and other resources.

Trigger: Fires when a user clicks on a downloadable link which opens either as a browser window (PDF) or initiates a download on their machine

Implementation:

JavaScript

```
window.dataLayer.push({  
  'event': 'file_download',  
  'file_name': '{{Click Text}}',  
  'file_url': '{{Click URL}}',  
  'file_type': '{{File Type}}',  
});
```

Language Changes

Purpose: Capture when users switch the website language.

Trigger: Fires when a user clicks on a switch language link

Implementation:

JavaScript

```
window.dataLayer.push({  
  'event': 'language_change',  
  'language': '{{Selected Language}}',  
  'page_path': '{{Page URL}}'  
});
```

Print / Share Content

Purpose: Monitor when users print or share content via social media or email.

Implementation: When users click a viable print or share button - if available on a page



JavaScript

```
window.dataLayer.push({
  'event': 'print_share_content',
  'action': '{{Print/Share Type}}',
  'page_path': '{{Page URL}}'
});
```

Video / Audio Start

Purpose: Track when users start playing a video or audio file.

Trigger: Fires when a user starts a video or audio file on the website

Implementation:

JavaScript

```
window.dataLayer.push({
  'event': 'video_audio_start',
  'media_type': '{{Media Type}}', //video / audio
  'media_name': '{{Media Name}}',
  'page_path': '{{Page URL}}'
});
```

Video / Audio Stop

Purpose: Track when users stop playing a video or audio file

Trigger: Fires when a user either stops a video / audio file, the video finishes playing, or when a user leaves the page.

Implementation:

JavaScript

```
window.dataLayer.push({
  'event': 'video_audio_stop',
  'media_type': '{{Media Type}}', //video / audio
  'media_name': '{{Media Name}}',
  'playback_point': '{{00:01:23}}', // timestamp
  'page_path': '{{Page URL}}'
```



```
});
```

Contact Us Click

Purpose: Track when users click to contact

Trigger: Fires when a user interacts with a contact button, clicking a phone number to contact the council

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'contact_us_click',
  'phone number': '{{Phone Number}}',
  'page_path': '{{Page URL}}'
});
```

Email Click

Purpose: Capture when users click on an email link.

Trigger: Fires when a user interacts with a contact button, clicking an email address to contact the council

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'email_click',
  'email_address': '{{Email Address}}',
  'page_path': '{{Page URL}}'
});
```

External Link Click

Purpose: Track clicks on outbound links leading to external websites.

Trigger: Fires when a user interacts with an external link click



Implementation:

JavaScript

```
window.dataLayer.push({  
  'event': 'external_link_click',  
  'link_text': '{{Link Text}}',  
  'open_in': '{{Opens in new URL or same window}}',  
  'link_url': '{{Link URL}}',  
  'page_path': '{{Page URL}}'  
});
```



Non-interaction & timed events

404 Event

Purpose: Track when users encounter a 'Page Not Found' error

Trigger: Fires when a page cannot be retrieved and does not exist. Only the event needs to fire, as we should capture all relevant information in GA4.

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'error_404',
});
```

Scroll Depth Tracking

Purpose: Measure how deeply users engage with content by tracking scroll depth at various thresholds (e.g., 25%, 50%, 75%, 100%).

Trigger: Fires when a user is at different proportions down the page.

Implementation:

```
JavaScript
window.dataLayer.push({
  'event': 'scroll_depth',
  'scroll_percentage': '{{Scroll Depth}}',
});
```