

# Tracking Dentli Booking Widget Events with Google Tag Manager

## Introduction

Dentli's online booking widget provides a seamless way for patients to book appointments on a website. To gain insights into this booking flow, we can track specific booking events through **Google Tag Manager (GTM)** and send them into **Google Analytics 4 (GA4)**. Dentli's widget automatically pushes these events into the website's **dataLayer**, which is a JavaScript object used by GTM to pass information (like events or variables) to tags ([developers.google.com](https://developers.google.com)). By configuring GTM to listen for these custom events and trigger GA4 tags, you can measure user behavior at each step of the booking process.

## Dentli DataLayer Events Overview

Dentli's booking widget pushes a series of custom events to the dataLayer as the user progresses. The key events to track (and their meanings) are:

- **PickTreatment** – Fired when a patient selects a treatment from the list (service selection).
- **PickDateTime** – Fired when a patient clicks on a date/time slot (appointment time selection).
- **PatientDetails** – Fired after the patient fills in all personal details and clicks "Continue" (moving past the info form).
- **ConfirmBooking** – Fired when the patient reaches the confirmation/review step (just before finalizing the booking).
- **online\_rebooking\_completed** – Fired when a booking change (rebooking) is successfully completed by the user.
- **online\_booking\_completed** – Fired when a new booking has been successfully placed (final booking confirmation).

Each of these events is pushed to the dataLayer with an `event` key (e.g., `dataLayer.push({'event': 'PickTreatment'})`). Our goal is to capture these events in GTM and send them to GA4 as event hits.

## Prerequisites

Before proceeding, make sure:

- **GTM is installed** on your website (e.g., via the GTM container snippet in your WordPress site's `<head>` and `<body>`).
- The Dentli booking widget script is installed and running on your site (so it pushes the events to the dataLayer).
- You have a GA4 property set up, and a GA4 Configuration tag in GTM (with your GA4 Measurement ID) firing on all pages. This ensures any GA4 event tags will use the correct GA4 settings.

## Setting Up Custom Event Triggers in GTM

To have GTM “listen” for Dentli’s booking events, we will create a **Custom Event trigger** for each event name. This tells GTM to fire tags whenever the specified dataLayer event occurs. Follow these steps for each Dentli event:

*Example of creating a Custom Event trigger in GTM (using a sample event name). Each trigger listens for a specific `event` pushed to the dataLayer.*

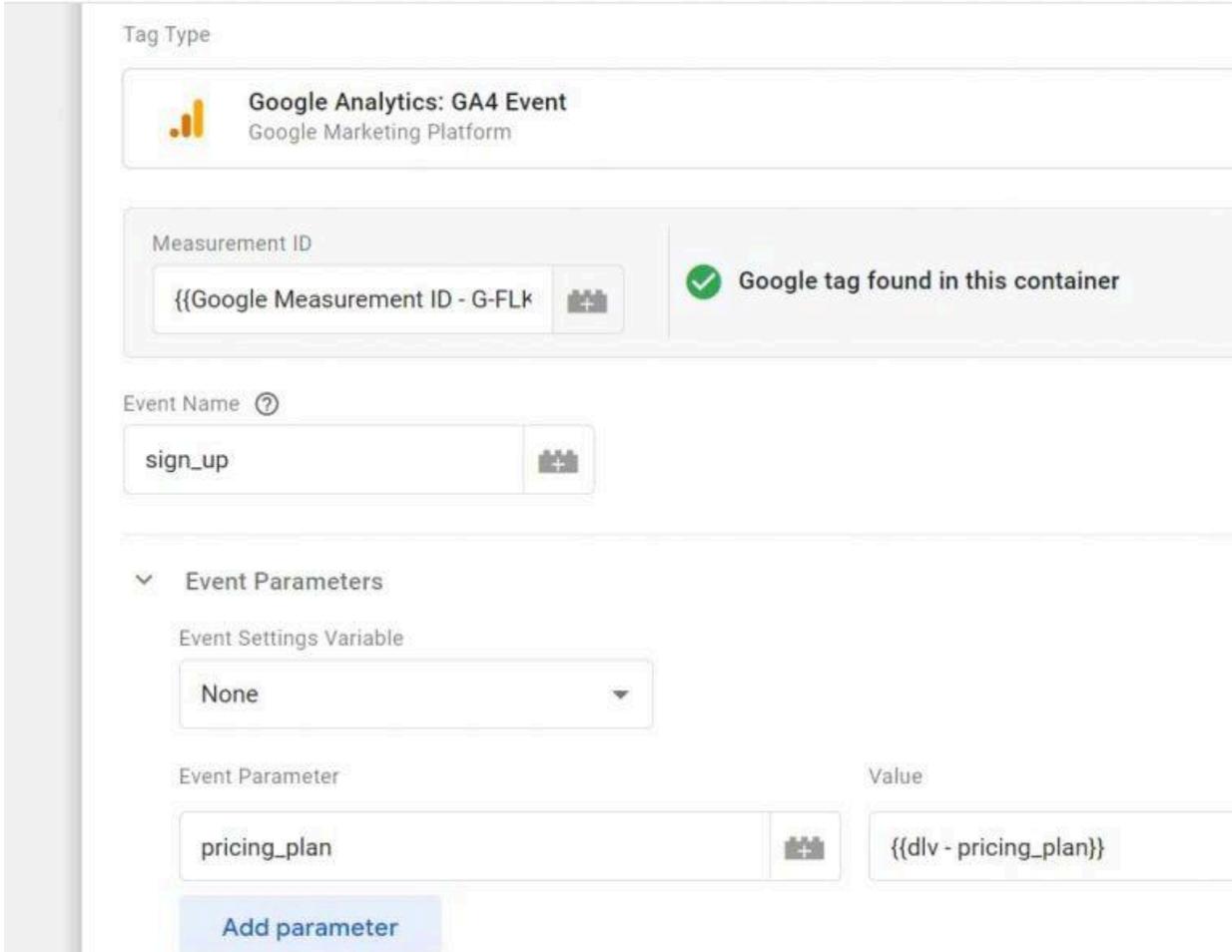
1. **Create a new Trigger:** In your GTM workspace, navigate to **Triggers** and click **New**.
2. **Choose Trigger Type:** Name the trigger (e.g., “Event – PickTreatment”), then click **Trigger Configuration** and select **Custom Event**.
3. **Specify Event Name:** In the Event Name field, enter the exact name of the Dentli event (for example, `PickTreatment`). Event names are case-sensitive, so they must match exactly.
4. **Configure Trigger Conditions:** Usually, you can leave the trigger set to fire on “All Custom Events” when the event name matches. (No additional conditions are needed unless you want to narrow down the pages or other criteria for firing.)
5. **Save the Trigger:** Click **Save**. Repeat the above to create triggers for each of the Dentli events (`PickDateTime`, `PatientDetails`, etc.) using their respective names. Make sure each trigger’s Event Name matches one of the Dentli dataLayer event strings.

After this, GTM will be aware of these custom events. In Preview mode, when you perform a booking action on the site, you should see the corresponding event appear in the GTM debug event stream (e.g., an event named "PickTreatment" when you pick a treatment). This confirms that the triggers are set up correctly to catch the dataLayer pushes.

## Configuring GA4 Event Tags for Dentli Events

With triggers in place, we need to send the event data to Google Analytics 4. We will create a **GA4 Event tag** for each event (or you can use one tag with a variable if you prefer a dynamic approach). Each tag will fire on its respective custom event trigger and send an event to GA4. In GTM:

GA4 - Event - Sign Up 



The screenshot shows the configuration for a Google Analytics 4 Event tag in GTM. The tag is titled "GA4 - Event - Sign Up". The tag type is "Google Analytics: GA4 Event" from the Google Marketing Platform. The measurement ID is set to "{{Google Measurement ID - G-FLk}}", and a green checkmark indicates "Google tag found in this container". The event name is "sign\_up". Under "Event Parameters", the "Event Settings Variable" is set to "None". One event parameter is defined: "pricing\_plan" with a value of "{{dlv - pricing\_plan}}". An "Add parameter" button is visible at the bottom.

*Example GA4 Event tag configuration in GTM. In this example, an event named "sign\_up" is being sent to GA4 with an additional parameter. For Dentli events, you would use the Dentli event name (or a mapped name) as the GA4 Event Name.*

1. **Create a GA4 Event Tag:** Go to **Tags** and click **New**. Name it descriptively (e.g., “GA4 – PickTreatment Event”).
2. **Tag Configuration:** Choose **Google Analytics: GA4 Event** as the tag type. If you have a GA4 Configuration tag set up, select it under *Configuration Tag* (or enter your Measurement ID manually if not using a config tag).
3. **Event Name:** Enter an event name to send to GA4. You can use the same name as the dataLayer event (e.g., “PickTreatment”), or map it to a more GA4-friendly name. For simplicity, using the same name is fine for custom events. *(If Google has a recommended event that matches the intent, you could use that. For example, for a completed booking you might use a name like “generate\_lead” or “purchase” depending on how you classify bookings. But a custom name is acceptable.)*
4. **Event Parameters (optional):** If the Dentli dataLayer push includes additional details (like treatment name, appointment date, etc.), you can capture those via **Data Layer Variables** and send them as parameters. For example, if `dataLayer.push({event: 'PickTreatment', treatmentName: 'Cleaning'})` is present, you could add an event parameter like `treatment_name: {{DLV - treatmentName}}`. This step is optional and requires creating those Data Layer Variables in GTM.
5. **Triggering:** Click the Triggering section and select the corresponding Custom Event trigger you created (e.g., the “Event – PickTreatment” trigger for the PickTreatment tag). This ensures the GA4 tag fires only when that specific event occurs.
6. **Save the Tag.** Repeat the process for each Dentli event you want to track (you can create separate GA4 event tags for each, or reuse one tag with a variable event name). Be sure to link each tag to the matching Custom Event trigger.

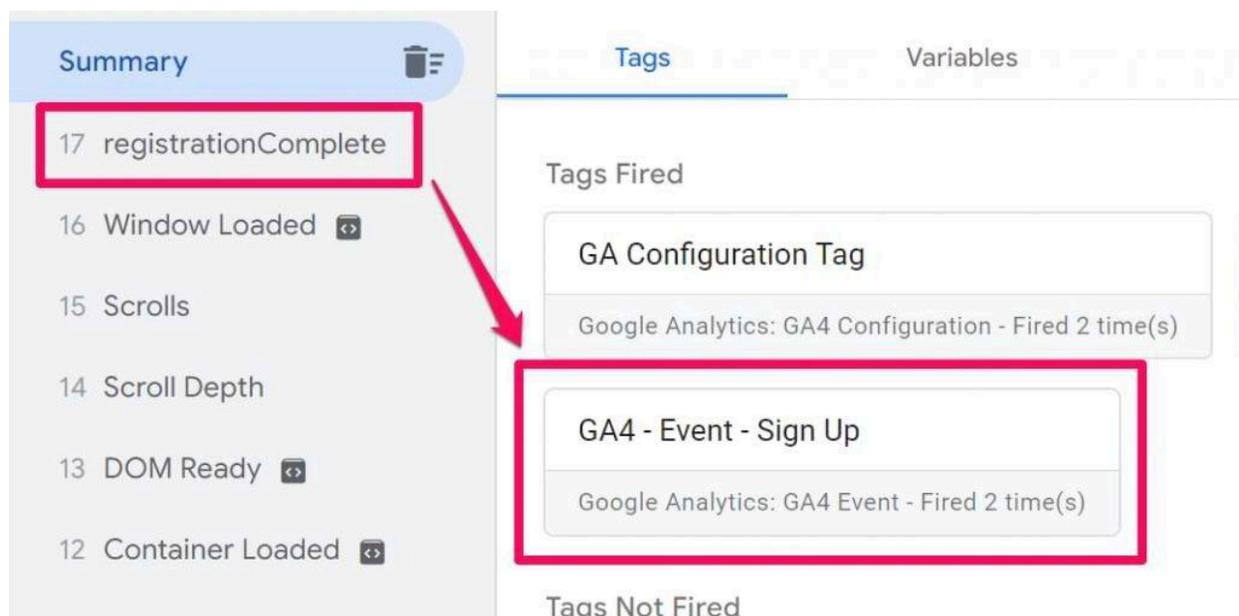
With this setup, every time a Dentli event is pushed to the dataLayer, GTM will fire the GA4 Event tag and send the hit to your GA4 property. For example, when a booking is successfully completed and `online_booking_completed` is pushed, your GA4 tag can send an event (perhaps named “online\_booking\_completed” or a custom name like “booking\_complete”) to GA4.

**Note:** *Ensure your GA4 configuration tag (or the GA4 event tag settings) has `debug_mode` enabled if you want to see these events in GA4 DebugView. In GA4, you might also want to mark the final booking completion events (e.g., `online_booking_completed` and `online_rebooking_completed`) as **Conversions** for easier tracking of completed bookings in GA4 reports.*

## Testing and Verification

After implementing the triggers and tags, it's crucial to test that everything works as expected before publishing the GTM container live.

1. **Use GTM Preview Mode:** Click **Preview** in GTM to open the Tag Assistant preview. Navigate to your website's booking page and perform a test booking (you don't need to finalize an actual appointment, but go through the motions of selecting a treatment, time, etc., up to confirmation).
2. **Observe DataLayer Events:** As you interact with the booking widget, watch the GTM debug panel. You should see the custom events appear in the **Event stream** on the left (e.g., an event named "PickTreatment" when you pick a treatment). Click on each event and check the **Data Layer** tab to confirm the event name and any parameters are present.



*GTM Preview mode showing a custom dataLayer event (`registrationComplete` in this example) in the event stream. You should see Dentli's events like "PickDateTime", "ConfirmBooking", etc., listed similarly when you trigger them.*

3. **Verify Tag Firing:** Still in Preview mode, click on an event like "PickTreatment" in the left panel and look at the **Tags** tab to ensure your GA4 Event tag fired on that event. GTM will indicate which tags fired (or not) for each event. Each custom event should trigger its corresponding GA4 tag.
4. **Check GA4 DebugView:** (Optional) Open your GA4 DebugView to see incoming events. If debug mode is on, you should see events with the names you configured (e.g., PickTreatment, PickDateTime, etc.) arriving in GA4. This confirms GA4 is receiving the

data.

5. **Publish Container:** Once everything looks good in preview, submit and publish your GTM container so the changes go live on the website. Monitor the real-time GA4 reports during actual bookings to double-check that events are logging from real users.

## Conclusion

By setting up custom event triggers and GA4 tags in Google Tag Manager, agencies and Dentli clients can effectively track each step of the online booking funnel. Whenever a patient interacts with the Dentli booking widget, the corresponding events (treatment selected, time chosen, details entered, booking confirmed, etc.) will be recorded in GA4. This data is invaluable for understanding user drop-off points and measuring the success of your online booking system. With the above configuration, you'll have a robust GTM container that captures Dentli's booking events in real-time, providing you actionable insights to optimize your patient scheduling process.