

Analytics Tracking Specification Document for OTT on Web platform(s)

Introduction

This document provides a detailed technical specification for implementing event tracking on the OTT web platform. The purpose of this document is to ensure consistent and accurate tracking of user interactions while maintaining user privacy and data protection in compliance with GDPR and CCPA regulations. The events covered in this specification are video_played and video_paused.

Dependencies

To implement event tracking on the web platform, the following dependencies are required:

- Amplitude JavaScript SDK: Install the Amplitude JavaScript SDK by including the following script tag in the `<head>` section of your HTML file:

```
<script src="https://cdn.amplitude.com/libs/amplitude-8.5.0-min.gz.js"></script>
```

- Initialize the Amplitude SDK with your project API key:

```
amplitude.getInstance().init("YOUR_API_KEY");
```

Implementation Guidelines

Event: video_played

Description: Triggered when a user starts playing a video.

Property	Type	Required	Example
video_id	string	Yes	"abc123"

video_title	string	Yes	"Example Video"
video_duration	number	Yes	120.5
device_type	string	Yes	"desktop"
device_manufacturer	string	Yes	"Apple"
device_model	string	Yes	"MacBook Pro"
device_platform	string	Yes	"web"
user_type	string	Yes	"registered"

Implementation:

```
amplitude.getInstance().logEvent('video_played', {
  video_id: 'abc123',
  video_title: 'Example Video',
  video_duration: 120.5,
  device_type: 'desktop',
  device_manufacturer: 'Apple',
  device_model: 'MacBook Pro',
  device_platform: 'web',
  user_type: 'registered'
}).
```

Trigger: Fire this event when the user starts playing a video.

Reporting Guidelines:

- In Amplitude, navigate to the 'Events' tab and search for the 'video_played' event.
- Analyze the event properties to gain insights into video playback behavior.

Event: video_paused

Description: Triggered when a user pauses a video.

--	--	--	--

Property	Type	Required	Example
video_id	string	Yes	"abc123"
video_title	string	Yes	"Example Video"
pause_time	number	Yes	60.2
device_type	string	Yes	"desktop"
device_manufacturer	string	Yes	"Apple"
device_model	string	Yes	"MacBook Pro"
device_platform	string	Yes	"web"
user_type	string	Yes	"registered"

Implementation:

```
amplitude.getInstance().logEvent('video_paused', {
  video_id: 'abc123',
  video_title: 'Example Video',
  pause_time: 60.2,
  device_type: 'desktop',
  device_manufacturer: 'Apple',
  device_model: 'MacBook Pro',
  device_platform: 'web',
  user_type: 'registered'
}).
```

Trigger: Fire this event when the user pauses a video.

Reporting Guidelines:

- In Amplitude, navigate to the 'Events' tab and search for the 'video_paused' event.
- Analyze the event properties to understand user engagement and identify potential drop-off points

drop-off points.

User Identity Handling

To handle user identity, use the Amplitude SDK's `setUserId` method for registered users and `setDeviceId` for anonymous users.

```
// For registered users
amplitude.getInstance().setUserId('user@example.com');

// For anonymous users
amplitude.getInstance().setDeviceId('anonymous-user-id');
```

Consent & Privacy Safeguards

To comply with GDPR and CCPA regulations, ensure that you obtain user consent before tracking any events. Use the Amplitude SDK's `setOptOut` method to honor user opt-out preferences.

```
// Check user consent
const hasConsent = checkUserConsent();

// Set opt-out if user has not given consent
amplitude.getInstance().setOptOut(!hasConsent);
```

Testing Guidelines

To verify that events are being tracked correctly:

1. Implement the tracking code as specified in the implementation guidelines.
2. Use browser developer tools or a network monitoring tool to check that the events are being sent to Amplitude with the correct properties.
3. Verify that the events appear in the Amplitude dashboard and can be queried using the reporting guidelines.

Product Manager Usage

Product managers can use the tracked events to create reports and funnels in Amplitude to gain insights into user behavior and engagement. For example:

- Create a funnel to analyze the drop-off points between video_played and video_paused events.
- Segment users based on device_type, device_manufacturer, or user_type to understand differences in video consumption patterns.
- Analyze video_duration and pause_time to identify potential areas for improving user engagement.

Documentation Links

- [Amplitude JavaScript SDK Documentation](#)
- [Amplitude Developer Documentation](#)

Best Practices

- Ensure that event naming and property naming conventions are consistent across the organization.
- Regularly review and update the tracking specification document as new events or properties are added.
- Collaborate with the data team to ensure that the tracked events are being used effectively for analysis and decision-making.