

## Customerlabs Assessment - 1

### 1. General Understanding:

What is the primary purpose of the Customer Labs JavaScript API documented at the given link?

CustomerLabs APIs enable to collection and sending of website data to CustomerLabs to unify and sync the data across multiple platforms. This data unification and synchronization across multiple platforms allow businesses to gain insights into, such as page views, clicks, and form submissions, and utilize this information for various analytics and marketing purposes.

In a practical example, if a user visits an e-commerce site, views a dress, and leaves without purchasing, the API enables targeted notifications, like a WhatsApp message with a special offer, to re-engage and encourage them to buy.

Summarize the key features and functionalities provided by the Customer Labs JavaScript API.

The key features and functionalities provided by the Customer Labs JavaScript API include:

- **Pageview Tracking:** Automatically tracks page views by default when the Customer Labs JavaScript is enabled. It also allows users to customize page view events, such as

tracking category views or specific product views.

- **Click Tracking:** Enables tracking of links or button clicks on the website, providing details about the clicked elements and associated properties.
- **Form Submission Tracking:** Allows tracking of form submissions, capturing information such as first name, phone number, email, and the page from which the form was submitted.

Explain the importance of website event tracking in the context of Customer Labs.

Website event tracking is crucial for Customer Labs as it provides valuable insights into user interactions and behaviors on the website. This data allows businesses to understand how users engage with their site, enabling them to make data-driven decisions for marketing strategies, user experience improvements, and personalized content delivery. Customer Labs helps businesses create a comprehensive view of customer journeys and enhance overall user satisfaction by tracking page views, clicks, and form submissions.

## **2. Initialization and Setup:**

Describe the steps involved in initializing the Customer Labs JavaScript API on a website:

- Include the Customer Labs script in your HTML file. This script is typically provided by Customer Labs and is loaded asynchronously or as part of your main script.

```
<script src="customerlabs.js"></script>
```

- After loading the script, initialize the Customer Labs API with the necessary configuration, such as the API key and project ID.

CODE:

```
var _cl = _cl || []; _cl.push(['init', { apiKey: 'your-api-key',  
projectId: 'your-project-id' }]);
```

- Load and execute the Customer Labs script after initialization.

CODE:

```
(function () { var cl = document.createElement('script');  
cl.type = 'text/javascript'; cl.async = true; cl.src =  
'customerlabs.js'; var s =  
document.getElementsByTagName('script')[0];  
s.parentNode.insertBefore(cl, s); })();
```

[What Parameters required when initializing the Customer Labs JavaScript API:](#)

- `apiKey`: A unique identifier associated with your Customer Labs account.
- `projectId`: Identifier for the specific project or account within Customer Labs.

## Event Tracking:

How does the Customer Labs JavaScript API facilitate event tracking on a website

- The Customer Labs JavaScript API provides functions like `pageview`, `trackClick`, and `trackSubmit` to send various types of events to the Customer Labs system. These functions take event names and properties as parameters and allow you to track user interactions on the website.

Example of tracking a custom event using the Customer Labs JavaScript API:

CODE:

```
_cl.pageview('Custom Event', {  "customProperties": {  
  "page_url": { "t": "string", "v": window.location.href },  
  "clicked_from": { "t": "string", "v": "header section" }  } });
```

## Custom Properties:

Explain the concept of custom properties in the context of event tracking with the Customer Labs JavaScript API:

- Custom properties allow you to attach additional information to an event, providing more context and details. For example, in a `pageview` event, you can include custom properties like `page_url` and `clicked_from` to capture specific details about the event.

How to include custom properties when tracking events:

CODE:

```
_cl.pageview('Custom Event', {  "customProperties": {
```

```
"page_url": { "t": "string", "v": window.location.href },  
"clicked_from": { "t": "string", "v": "header section" } } }));
```

### **User Identification:**

Describe the methods available for identifying users with the Customer Labs JavaScript API:

- User identification is crucial for tracking individual user actions. Methods for user identification include automatic identification by the Customer Labs script or manual identification using the identify function.

CODE:

```
_cl.push(['identify', 'user123']);
```

Purpose of user identification and its contribution to analytics:

- User identification allows tracking and analyzing the actions of individual users over time. It contributes to personalized analytics, providing insights into individual user behavior on the website.

I have acquired additional knowledge through this prework.

I learned the content from the JavaScript API Documentation, ChatGPT, and Google.

Thank you!