# Clarity client API

You can quickly get started with Clarity without coding but by interacting with the Clarity client API. This API can help you access advanced features as described in this reference. Add the following calls to Clarity APIs to the HTML or JavaScript of your webpage to access these features.

**Note**

Your Clarity ID serves as your API key. No other client API key is necessary, and there is no cost for using Clarity client APIs.

## Overview

### JavaScript APIs

| Purpose | Syntax | Parameters | Required? |
|---|---|---|---|
| Cookie consent | `window.clarity(&#39;consent&#39;)` | String | Yes |
| Custom identifiers | `window.clarity(&quot;identify&quot;, <customuserid>,<customsessionid>,<custompageid>`) | Strings | `<customuserid>`: Yes, `<customsessionid>`: No, `<custompageid>;`: No |
| Custom tags | `window.clarity(&quot;set&quot;, <key>, <value>`) | key: string, value: string or an array of strings | Yes |
| Upgrade session | `window.clarity(&quot;upgrade&quot;, <upgrade reason`>) | Strings | "upgrade": Yes, `<upgrade reason>`: Yes |

### HTML APIs

| Purpose | Syntax | Parameters | Required? |
|---|---|---|---|
| Mask content | `data-clarity-mask=&quot;true&quot;` | Boolean | Yes |
| Unmask content | `data-clarity-unmask=&quot;true&quot;` | Boolean | Yes |

# API Reference

## Specify if cookies should be set

If your project is configured to require cookie consent, Clarity uses a unique first-party cookie to track each user with a browser cookie. If cookie consent is required, you must call the consent API to indicate that you have consent from the user to track them using a cookie. Otherwise, Clarity gives each page a unique ID instead of a cookie when cookie consent is required. Clarity will automatically set the first-party cookie by default, and you don't need to call this API.

| Syntax | Parameters | Required? |
|---|---|---|
| `window.clarity(&#39;consent&#39;)` | String | Yes |

### Example

In this example, a `consentGranted` event is fired when the user accepts cookies:

```
window .addEventListener("consentGranted", () => window.clarity('consent'));
```

When the `consent` argument is passed to `clarity`, cookies will be set.

### Tip

Learn more about how to manage **cookie consent on Clarity**.

## Customize your identifiers

Clarity automatically generates various types of identifiers needed for its normal functioning. However, you can assign other names for `user id`, `session id`, and `page id` if you want to have custom features on your site that requireS it.

| Syntax | Parameters | Required? |
|---|---|---|
| `window.clarity(&quot;identify&quot;, <customuserid>, <customsessionid>, <custompageid>)` | Strings | `<customuserid>`: Yes, `<customsession id>`: No, `<custompageid>`: No |

### Examples

JavaScriptCopy

```
window.clarity("identify","myuserid")
window.clarity("identify","myuserid", "mysessionid")
window.clarity("identify", "myuserid", "mysessionid", "mypageid")
```

## Add custom tags

Clarity offers many pre-defined ways to filter and analyze website data. Still, you may want to track things that are specific to your site or user experience. With custom tags, you can apply arbitrary tags to your Clarity session.

To use custom tags, pass the `set` argument along with a key-value pair to define a tag in your JavaScript. When Clarity collects data for that tag, it will appear in the Filters options.

| Syntax | Parameters | Required? |
|---|---|---|
| `window.clarity(&quot;set&quot;,` "key", "value") | key: string, value: string or an array of strings | Yes |

**Note**

You can call this API multiple times. There is no limit to the number of custom tags you can have.

### Examples

```
window.clarity("set", "experiment", "experiment1")
window.clarity("set", "flight",["flight1", "flight2"])
```

Note: The last call has the same effect as:

```
window.clarity("set", "fight", "flight1")
window.clarity("set", "flight", "flight2")
```

**Tip**

Learn more about Clarity's **filtering options** and **custom tags**.

## Mask your site content

By default, your users' sensitive content is masked. We classify all input box content, numbers, and email addresses as sensitive content. Masked content isn't uploaded to Clarity.

If you want more control over which content on your site is masked, you can mask content using the Clarity website or add the `data-clarity-mask` property to HTML elements on your site.

**Note**

Setting `data-clarity-mask` to false has no effect. To unmask content, use **data-clarity-unmask**.

| **Syntax** | **Parameters** | **Required?** |
|---|---|---|
| `data-clarity-mask=&quot;true&quot;` | Boolean | Yes |

**Example**

```
<form action="" method="get" data-clarity-mask="true">
    <label for="GET-name">User Name:</label>
    <input id="GET-name" type="text" name="name">
    <input type="submit" value="Submit">
</form>
```

## Unmask your site content

Suppose you want to ensure that specific data items aren't sent to Clarity. In that case, you can unmask them by using `data-clarity-unmask`.

**Note**

Setting `data-clarity-unmask` to false has no effect. To mask content, use **data-clarity-mask**.

| **Syntax** | **Parameters** | **Required?** |
|---|---|---|
| `data-clarity-unmask=&quot;true&quot;` | Boolean | Yes |

**Example**

```
<article class="Movie Review" data-clarity-unmask="true">
    <header>
```

```
        <h2>Star Wars</h2>
    </header>
    <section>
        <p>A classic!</p>
    </section>
</article>
```

## Prioritize specific sessions for recording

Clarity records up to 100,000 sessions per project per day. If your project’s total volume of sessions exceeds the maximum daily limit, some sessions will be throttled for playback. By default, Clarity will use rules to capture the most interesting sessions for its recordings.

You can use the `upgrade` API to prioritize specific types of sessions for recording. This is useful if you have sessions with specific types of events (such as clicks) that you want to look at or interactions with specific parts of your website (such as a shopping cart).

| Syntax | Parameters | Required? |
|---|---|---|
| `window.clarity(&quot;upgrade&quot;, <upgrade reason>)` | Strings | "upgrade": Yes, `<upgrade reason>`: Yes |

### Example

```
window.clarity("upgrade", "button click")
```