Ez Scheduler API Documentation - version 1.1.0

Getting Started

Once you inform us that you would like to provide this service to your patients, our team will create a unique API key that will be sent to your practice, as a part of the short script, that you can easily implement to your website on the page that you would like patients to schedule appointments from.

API key is bound to one or more domains, so the scheduling widget can be initialized only on registered domain(s).

Ez Scheduler initialization script sample:

```
(function (window, document) {
 window.__EzSchedulerConfig__ = {
   apiKey: "ZWU4...NGE0",
   primaryColor: "rgb(50, 124, 247)",
 };

 var s = document.createElement("script");
 s.type = "text/javascript";
 s.async = true;
 var int = (window.__EzSchedulerConfig__.internal = {
   s: "https://ezschedulerdev.ezinfra.net",
 });
 s.src = int.s + "/bootstrap.js";
 var before = document.getElementsByTagName("script")[0];
 before.parentNode.insertBefore(s, before);
})(window, document);
```

In order to install Ez Scheduler widget, just add the provided code snippet to your website. After initialization, by default, the blue floating button will appear in the bottom right corner of the parent web page. We will show you in this document how you can customize the widget appearance.

Config parameters

Here are listed available config parameters that could be added to the `window.__EzSchedulerConfig__` object in initialization code snippet:

apiKey

Type: string
Required: true
Example: "ZWU4ZDEwNjUtNzIx1234NzI0LWI3NmQtN2YzZGNkZjFjNGE0"

apiKey is your identifier. Based on this key we present correct information about your practice to patients. apiKey is restricted only to domains requested during the key issuing process. If the key is not valid or the domain on which the widget is initialized is not supported, the widget won't show up in the web page and it will log errors in the browser console.

primaryColor

Type: string
Required: false
Default value: "rgb(50, 124, 247)"
Example: "#565656"

Theme primary color. Supports both hex and rgb format.

gradientColor

Type: string
Required: false
Default value: "linear-gradient(
 90deg,
 rgba(50, 124, 247, 1) 0%,
 rgba(50, 154, 247, 1) 75%,
 rgba(50, 144, 247, 0.8) 100%
)"

If you want to color backgrounds with gradients (where applicable), set this parameter to a value compatible with `linear-gradient` css. When using `gradientColor`, also set a `primaryColor` parameter, because it will be used in places where gradientColor could not be used.

initiallyOpened

Type: boolean
Required: false
Default value: false
Example: true
You can set this parameter to true, in order to open the widget straight away after the initialization process.

hideFloatingButton

Type: boolean
Required: false
Default value: false
Example: true

This option might be useful if you plan to open/close the widget in some other way, other than the default floating button. See EzScheduler API

hideInsuranceStep

Type: boolean
Required: false
Default value: false
Example: true

Set this option to true in order to completely remove “Insurance Information” step

position

Type: string
Required: false
Default value: “right”
Example: “left” or “right”

Set this option to “left” in order to position the widget in the bottom-left corner instead of the default bottom-right position.

containerId

Type: string
Required: false

Default value: none
Example: "ez-scheduler-container"

If the default widget positioning doesn't work for you, you have an option to render Ez Scheduler widget inside any HTML element, which is identified by `containerId` config parameter. Be aware that the widget will adapt it's height and width to the given container, so now it's in your hands to correctly position and size the widget container.

onLoad

Type: function
Required: false
Default value: none
Example: `() => {console.log("Widget is initialized, you can call EzScheduler APIs now")}`

Callback function, called when the widget is initialized. After this event, you can call `EzScheduler.open()` and `EzScheduler.close()` APIs. See EzScheduler API

EzScheduler API

When initialized, the `window.EzScheduler` global object is available. It contains two API functions, `open` and `close`. Widget is considered initialized when `onLoad` callback function is called.

open

Type: function
Example: `window.EzScheduler.open()`

You can use this api to implement your custom logic for opening/closing the widget. For example, if the case is that the widget should be opened by clicking some standard button from the host web page, then you could hide the default floating button (`hideFloatingButton`) and use this API to open the widget when some other button is clicked.

close

Type: function
Example: `window.EzScheduler.close()`