



How-To Guides

OAuth

Fullscript.js

What Is Fullscript.js?

Before you get started

Using Fullscript.js

Setup

OAuth And Backend Work

Add Embeddable Compone...

Handle Events

Go Live

Embeddable features

Reference

Fullscript Redirect

Webhooks

Events

Handle events

Your integration could be considered complete now that you have the feature loaded on the page. However, there's more you can do to create a seamless user experience.

Fullscript.js fires events when actions happen that your app may want to know about. For example, an event is fired when a practitioner finishes creating a treatment plan for a patient. The event includes data about the treatment plan that your app can store. When receiving this event, you'll also know the user is done with the feature and you can choose to unmount it.

FYI

Unmounting is optional. It removes the feature from your page, but you can also leave it there, or navigate away if that makes sense for your use case.

Take a moment to review the [available events for your feature](#).

Begin listening for events

To listen for the events fired by the embedded feature, subscribe with `treatmentPlanFeature.on()`.

Listen for Treatment Plan: patient.selected (event)

Create a function to call when the `patient.selected` event is triggered. Connect this to the event with `treatmentPlanFeature.on()`.

```
function handlePatientSelected(data) {  
  console.log("data", data);  
  // Use this as an opportunity to save the Fullscript patient id,  
  // unmount the feature, change page etc  
}  
  
treatmentPlanFeature.on("patient.selected", handlePatientSelected);
```

Or, use TypeScript (if using NPM version) to add the EventListener:

```
const handlePatientSelected = (data: EventListenerPayload<"patient.se"  
  console.log(data);  
  // Use this as an opportunity to save the Fullscript patient id,  
  // unmount the feature, change page etc  
};
```

Listen for Treatment Plan: activated (event)

Create a function to call when the `treatmentPlan.activated` event is triggered. Connect this to the event with `treatmentPlanFeature.on()`.

```
function handleTreatmentPlanActivated(data) {  
  console.log("data", data);  
  // Use this as an opportunity to store data about the plan,  
  // unmount the feature, change page etc  
}  
  
treatmentPlanFeature.on("treatmentPlan.activated", handleTreatmentPlan
```

Alternately, use TypeScript (if using NPM version):

```
const handleTreatmentPlanActivated = (data: EventListenerPayload<"tre"  
  console.log("data", data);  
  // Use this as an opportunity to store data about the plan,  
  // unmount the feature, change page etc  
};
```

Stop listening for events

If desired, you can also stop listening to an event:

```
treatmentPlanFeature.off("treatmentPlan.activated", handleTreatmentPlan  
treatmentPlanFeature.off("patient.selected", handlePatientSelected);
```

On this page

Begin listening for events

Listen for Treatment Plan: patient.selected (event)

Listen for Treatment Plan: activated (event)

Stop listening for events

Related Topics

Fullscript.js events

credentialing.com/submit/011/patient-selected/management-selected/

< Add embeddable compo...

Go live >