

ion-loading

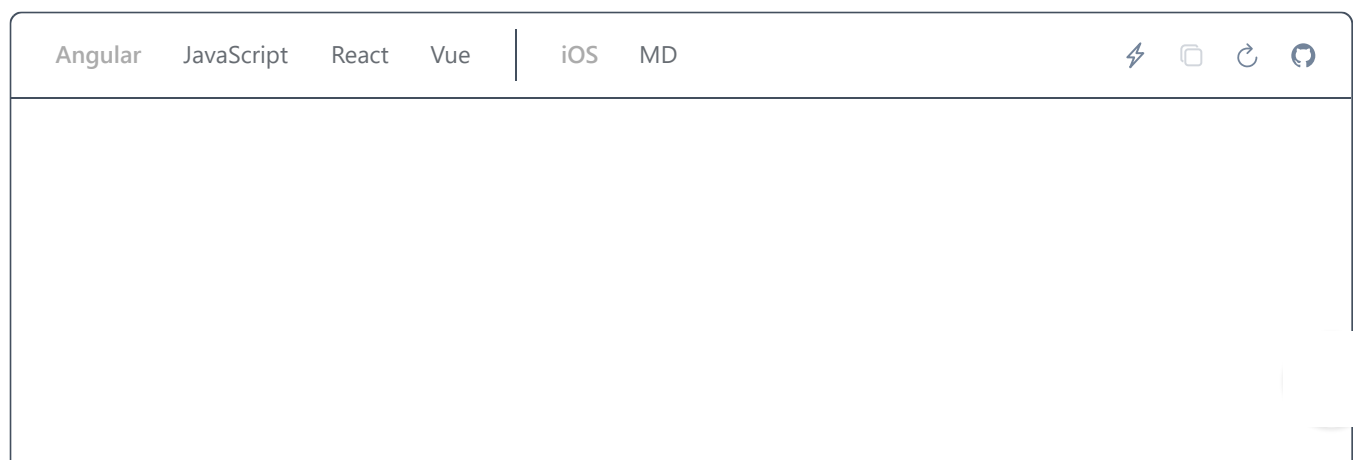
An overlay that can be used to indicate activity while blocking user interaction. The loading indicator appears on top of the app's content, and can be dismissed by the app to resume user interaction with the app. It includes an optional backdrop, which can be disabled by setting `showBackdrop: false` upon creation.

Basic Usage

Once presented, the loading indicator will display indefinitely by default. Developers can manually dismiss the loading indicator after creation by calling the `dismiss()` method on the component. The `onDidDismiss` function can be called to perform an action after the loading indicator is dismissed.

Alternatively, developers can configure the loading indicator to dismiss automatically after a specific amount of time by passing the number of milliseconds to display it in the `duration` of the loading options.

Controller



Show Loading

<> src/app/example.component.html TS src/app/example.component.ts

Copy

```
import { Component } from '@angular/core';

import { LoadingController } from '@ionic/angular';

@Component({
  selector: 'app-example',
  templateUrl: 'example.component.html',
})
export class ExampleComponent {
  constructor(private loadingCtrl: LoadingController) {}

  async showLoading() {
    const loading = await this.loadingCtrl.create({
      message: 'Dismissing after 3 seconds...',
      duration: 3000,
    });

    loading.present();
  }
}
```

Inline

Ionic React and Ionic Vue users also have the option to use the `ion-loading` component directly in their template.

React Vue

```
import React, { useState } from 'react';
import { IonLoading, IonButton, IonContent } from '@ionic/react';

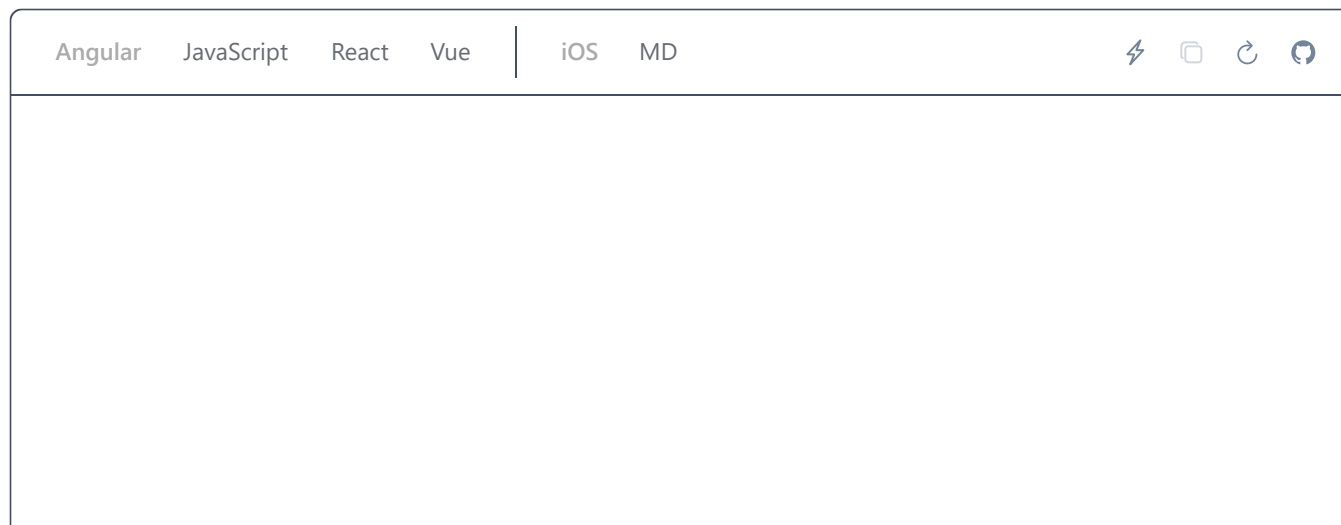
export const LoadingExample: React.FC = () => {
  const [showLoading, setShowLoading] = useState(false);

  return (
    <IonContent>
      <IonButton onClick={() => setShowLoading(true)}>Show Loading</IonButton>
      <IonLoading
        cssClass="my-custom-class"
        isOpen={showLoading}
        onDidDismiss={() => setShowLoading(false)}
        message={'Please wait...'}
        duration={5000}
      />
    </IonContent>
  );
};
```

Customization

Spinners

The spinner that is used can be customized using the `spinner` property. See the [spinner property documentation](#) for a full list of options.



Show Loading

 src/app/example.component.html  src/app/example.component.ts

```
<ion-button (click)="showLoading()">Show Loading</ion-button>
```

Theming

Loading uses scoped encapsulation, which means it will automatically scope its CSS by appending each of the styles with an additional class at runtime. Overriding scoped selectors in CSS requires a [higher specificity](#) selector.

We recommend passing a custom class and using that to add custom styles to the host and inner elements.

[Angular](#)[JavaScript](#)[React](#)[Vue](#)[iOS](#)[MD](#)

Show Loading

<> src/app/example.component.html TS src/app/example.component.ts  src/global.css

```
import { Component } from '@angular/core';

import { LoadingController } from '@ionic/angular';

@Component({
  selector: 'app-example',
  templateUrl: 'example.component.html',
})
export class ExampleComponent {
  constructor(private loadingCtrl: LoadingController) {}

  async showLoading() {
    const loading = await this.loadingCtrl.create({
      message: 'Loading...',
      duration: 3000,
      cssClass: 'custom-loading',
    });

    loading.present();
  }
}
```

Note

ion-loading is presented at the root of your application, so we recommend placing any ion-loading styles in a global stylesheet.

Interfaces

LoadingOptions

```
interface LoadingOptions {
  spinner?: SpinnerTypes | null;
  message?: string | IonicSafeString;
  cssClass?: string | string[];
  showBackdrop?: boolean;
  duration?: number;
  translucent?: boolean;
  animated?: boolean;
  backdropDismiss?: boolean;
  mode?: Mode;
  keyboardClose?: boolean;
  id?: string;
  htmlAttributes?: { [key: string]: any };

  enterAnimation?: AnimationBuilder;
  leaveAnimation?: AnimationBuilder;
}
```

Properties

animated

Description	If true , the loading indicator will animate.
Attribute	animated
Type	boolean

Default	true
---------	------

backdropDismiss

Description	If <code>true</code> , the loading indicator will be dismissed when the backdrop is clicked.
Attribute	<code>backdrop-dismiss</code>
Type	boolean
Default	false

cssClass

Description	Additional classes to apply for custom CSS. If multiple classes are provided they should be separated by spaces.
Attribute	<code>css-class</code>
Type	<code>string</code> <code>string[]</code> <code>undefined</code>
Default	undefined

duration

Description	Number of milliseconds to wait before dismissing the loading indicator.
Attribute	<code>duration</code>
Type	number

Default	0
---------	---

enterAnimation

Description	Animation to use when the loading indicator is presented.
Attribute	undefined
Type	((baseEl: any, opts?: any) => Animation) undefined
Default	undefined

htmlAttributes

Description	Additional attributes to pass to the loader.
Attribute	undefined
Type	undefined { [key: string]: any; }
Default	undefined

keyboardClose

Description	If true , the keyboard will be automatically dismissed when the overlay is presented.
Attribute	keyboard-close
Type	boolean

Default	true
---------	------

leaveAnimation

Description	Animation to use when the loading indicator is dismissed.
Attribute	undefined
Type	((baseEl: any, opts?: any) => Animation) undefined
Default	undefined

message

Description	Optional text content to display in the loading indicator.
Attribute	message
Type	IonicSafeString string undefined
Default	undefined

mode

Description	The mode determines which platform styles to use.
Attribute	mode
Type	"ios" "md"
Default	undefined

showBackdrop

Description	If <code>true</code> , a backdrop will be displayed behind the loading indicator.
Attribute	<code>show-backdrop</code>
Type	<code>boolean</code>
Default	<code>true</code>

spinner

Description	The name of the spinner to display.
Attribute	<code>spinner</code>
Type	<code>"bubbles"</code> <code>"circles"</code> <code>"circular"</code> <code>"crescent"</code> <code>"dots"</code> <code>"lines"</code> <code>"lines-sharp"</code> <code>"lines-sharp-small"</code> <code>"lines-small"</code> <code>null</code> <code>undefined</code>
Default	<code>undefined</code>

translucent

Description	If <code>true</code> , the loading indicator will be translucent. Only applies when the mode is <code>"ios"</code> and the device supports backdrop-filter .
Attribute	<code>translucent</code>
Type	<code>boolean</code>
Default	<code>false</code>

Events

Name	Description
<code>ionLoadingDidDismiss</code>	Emitted after the loading has dismissed.
<code>ionLoadingDidPresent</code>	Emitted after the loading has presented.
<code>ionLoadingWillDismiss</code>	Emitted before the loading has dismissed.
<code>ionLoadingWillPresent</code>	Emitted before the loading has presented.

Methods

dismiss

Description	Dismiss the loading overlay after it has been presented.
Signature	<code>dismiss(data?: any, role?: string) => Promise<boolean></code>

onDidDismiss

Description	Returns a promise that resolves when the loading did dismiss.
Signature	<code>onDidDismiss<T = any>() => Promise<OverlayEventDetail<T>></code>

onWillDismiss

Description	Returns a promise that resolves when the loading will dismiss.

Signature	<code>onWillDismiss<T = any>() => Promise<OverlayEventDetail<T>></code>
-----------	--

present

Description	Present the loading overlay after it has been created.
Signature	<code>present() => Promise<void></code>

CSS Shadow Parts

No CSS shadow parts available for this component.

CSS Custom Properties

Name	Description
<code>--backdrop-opacity</code>	Opacity of the backdrop
<code>--background</code>	Background of the loading dialog
<code>--height</code>	Height of the loading dialog
<code>--max-height</code>	Maximum height of the loading dialog
<code>--max-width</code>	Maximum width of the loading dialog
<code>--min-height</code>	Minimum height of the loading dialog
<code>--min-width</code>	Minimum width of the loading dialog
<code>--spinner-color</code>	Color of the loading spinner

Name	Description
--width	Width of the loading dialog

Slots

No slots available for this component.

[Edit this page](#)