**IONIC**

**Introduction**

Ionic is an open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies — HTML, CSS, and JavaScript — with integrations for popular frameworks like [Angular](https://ionicframework.com/docs/angular/overview), [React](https://ionicframework.com/docs/react), and [Vue](https://ionicframework.com/docs/vue/overview).

**Installing Ionic**

Ionic apps are created and developed primarily through the Ionic [command-line](https://ionicframework.com/docs/reference/glossary#cli) utility. The Ionic CLI is the preferred method of installation, as it offers a wide range of dev tools and help options along the way. It is also the main tool through which to run the app and connect it to other services, such as Appflow.

**Install the Ionic CLI**[**​**](https://ionicframework.com/docs/intro/cli#install-the-ionic-cli)

Before proceeding, make sure your computer has [Node.js](https://ionicframework.com/docs/reference/glossary#node) installed. See [these instructions](https://ionicframework.com/docs/intro/environment) to set up an environment for Ionic.

Install the Ionic CLI with npm:

**npm install -g @ionic/cli**

If there was a previous installation of the Ionic CLI, it will need to be uninstalled due to a change in package name.

$ npm uninstall -g ionic  
$ npm install -g @ionic/cli  
  
**Note**

The -g option means *install globally*. When packages are installed globally, EACCES permission errors can occur. Consider setting up npm to operate globally without elevated permissions. See [Resolving Permission Errors](https://ionicframework.com/docs/developing/tips#resolving-permission-errors) for more information.

**Start an App**[**​**](https://ionicframework.com/docs/intro/cli#start-an-app)

Create an Ionic app using one of the pre-made app templates, or a blank one to start fresh. The three most common starters are the blank starter, tabs starter, and sidemenu starter. Get started with the ionic start command:

**ionic start**



**Run the App**[**​**](https://ionicframework.com/docs/intro/cli#run-the-app)

The majority of Ionic app development can be spent right in the browser using the ionic serve command:

**$ cd myApp**

**$ ionic serve**

There are a number of other ways to run an app, it's recommended to start with this workflow. To develop and test apps on devices and emulators.

# UI Components

# ion-action-sheet

An Action Sheet is a dialog that displays a set of options. It appears on top of the app's content, and must be manually dismissed by the user before they can resume interaction with the app. Destructive options are made obvious in ios mode. There are multiple ways to dismiss the action sheet, including tapping the backdrop or hitting the escape key on desktop.

**Buttons**[**​**](https://ionicframework.com/docs/api/action-sheet#buttons)

A button's role property can either be destructive or cancel. Buttons without a role property will have the default look for the platform. Buttons with the cancel role will always load as the bottom button, no matter where they are in the array. All other buttons will be displayed in the order they have been added to the buttons array. Note: We recommend that destructive buttons are always the first button in the array, making them the top button. Additionally, if the action sheet is dismissed by tapping the backdrop, then it will fire the handler from the button with the cancel role.

A button can also be passed data via the data property on ActionSheetButton. This will populate the data field in the return value of the onDidDismiss method.

**Customization**[**​**](https://ionicframework.com/docs/api/action-sheet#customization)

Action Sheet 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 to cssClass in the create method and using that to add custom styles to the host and inner elements. This property can also accept multiple classes separated by spaces. View the Usage section for an example of how to pass a class using cssClass.

/\* DOES NOT WORK - not specific enough \*/  
.action-sheet-group {  
 background: #e5e5e5;  
}  
  
/\* Works - pass "my-custom-class" in cssClass to increase specificity \*/  
.my-custom-class .action-sheet-group {  
 background: #e5e5e5;  
}

Any of the defined CSS Custom Properties can be used to style the Action Sheet without needing to target individual elements:

.my-custom-class {  
 --background: #e5e5e5;  
}