

# Platform-Specific Development Lab

We really try to make all of our code work on all platforms. It would be great if we could do that 100% of the time! But we've seen that sometimes it's necessary to detect the type of OS our app is running on at the time and change how it behaves. We'll work on that in this lab by adding a DatePicker that is intelligent enough to change how it displays based on the OS.

## Implementing DatePickerIOS

1. Edit Landing.js. In the JSX, add a DatePickerIOS just above the list of films, setting the date to today and the onChange event to a new method called setDate.
2. Write setDate to console.log() the date chosen.
3. If you're able to debug this on an iPhone device or simulator, run and test. Make sure you can see the date chosen in the console.
4. Make the DatePickerIOS show only the date and not the time of day.

This is great, but it'll only work on iOS. Let's see if we can support it on Android.

## Implementing DatePickerAndroid

5. Add a Button titled "Pick a showing date". The onPress event should call pickDate.
6. pickDate should call DatePickerAndroid.open. You can either use async/await or handle the Promise, but you want to get the date chosen in a local variable.
7. Console.log that local variable.
8. Run and test on an Android device or emulator until you can see the chosen date in the console.

## Using the Platform module

9. At this point we have an app that works on both platforms but also errors on both platforms. Let's get rid of those errors by detecting the OS. We can do that with the Platform module.
10. import Platform at the top of Landing.js.
11. In the JSX, find where you're including the <DatePickerIOS> and use an expression to make this conditional based on us being on an iOS device. Something like this:  

```
{ Platform.OS === "ios" && <DatePickerIOS .... > }
```
12. Then, where you're drawing the button, make it conditional on the Platform.OS being "android"
13. Run and test. Adjust until it is working.

## Extracting the DatePicker

This is great and all but we're going to want to have this functionality on another component. Let's extract it so it can be reused.

14. Create a new component called DatePicker. It should import our Redux store and initialize state in a constructor:

```
this.state = {...store.getState() }
```

15. Also, since we want state to refresh on every re-render, add this:

```
static getDerivedStateFromProps(nextProps, prevState) {  
  return { ...store.getState() };  
}
```

16. Make the JSX return a Button:

```
<Button onPress={this.showModal}  
  title="Showing times for {this.state.selected_date.toDateString()}" />
```

17. Write the showModal method. If we're on Android, it should DatePickerAndroid.open(). (Hint: Take out the logic from Landing.js and put it here to set a date on Android).
18. In that success callback, go ahead and ...  
`store.dispatch(  
 {type: "SET_SELECTED_DATE", date:new Date(date.year, date.month, date.day)})`
19. And of course you'll need to handle that action in reducers.js. Put in another case sort of like this:  
`case "SET_SELECTED_DATE":  
 return { ...state, selected_date: action.date };`
20. Run and test. On iOS nothing works yet. But on Android, you should be able to press the button and then pick a date which is saved in our application state and displayed on the button.

## Now for iOS

21. In showModal if we're on iOS, set a variable called showIOSPicker in state:  
`this.setState({ showIOSPicker = !this.state.showIOSPicker });`
22. Back in the JSX, conditionally show the DatePickerIOS control. If this.state.showIOSPicker is truthy, show it. If not, don't.
23. Set your DatePickerIOS's date property to this.state.selected\_date.
24. Make its onChange event set showIOSPicker to false and dispatch the "SET\_SELECTED\_DATE" action to the store. (Hint: it will be kind of like the Android dispatch, but not exactly like the Android dispatch. Watch out!)
25. Run and test on iOS. You should be able to hit the button and see an iOS date picker show. When the user chooses a date, the selector should disappear and the date displayed in the button should change.