Have ideas to improve npm? Join in the discussion! »

@react-native-community/datetimepicker TS

3.0.9 • Public • Published a month ago

Readme

Explore BETA

1 Dependency

90 Dependents

30 Versions

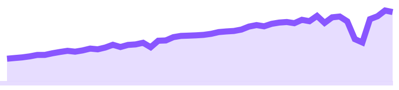
Install

```
npm i @react-native-community/datetimepicker
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Weekly Downloads

139,787



Version	License
3.0.9	MIT
Unpacked Size	Total Files
183 kB	62
Issues	Pull Requests
70	3

Homepage

github.com/react-native-community/datetimepicker#readme

Repository

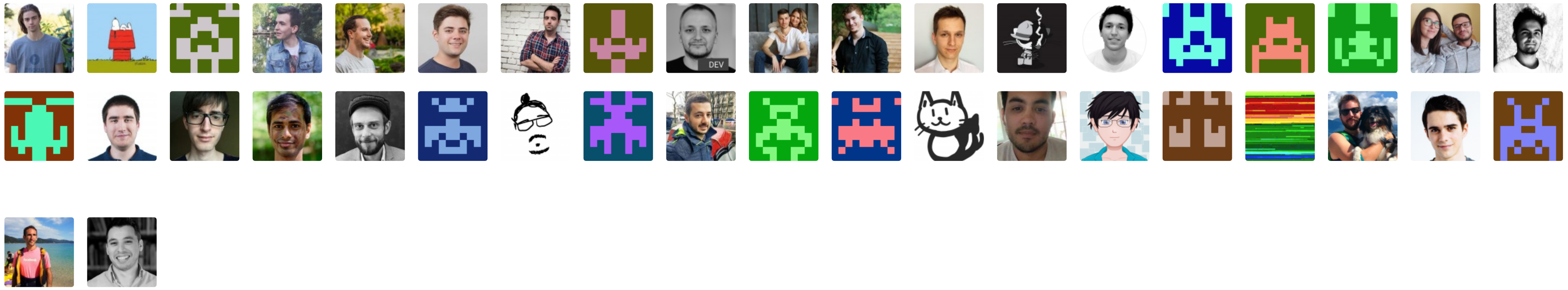
github.com/react-native-community/datetimepicker

Last publish

a month ago

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Collaborators



>_ **Try** on RunKit

Report malware

🚧🚧 Looking for collaborators 🚧🚧

See this **issue**

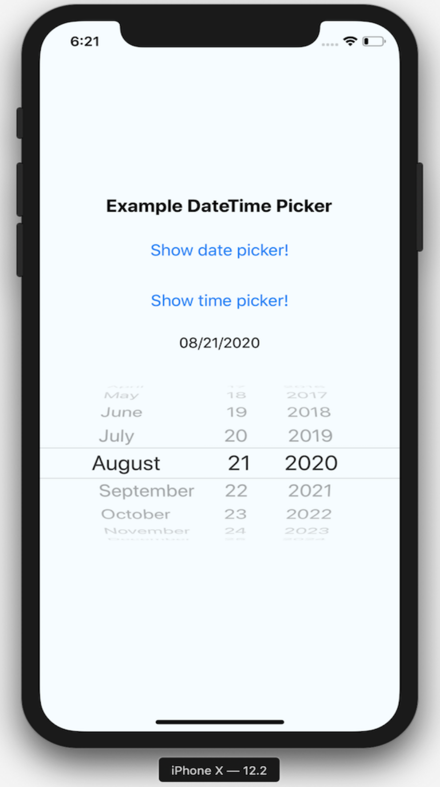
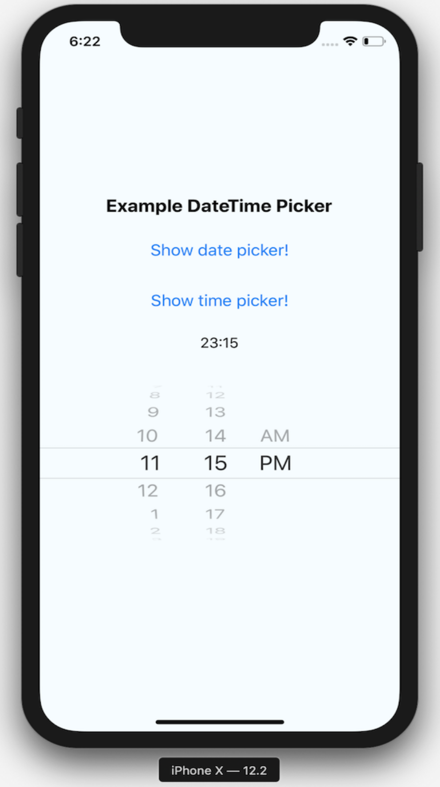
This repository was moved out of the react native community GH organization, in accordance to **this proposal**. The module is still published on `npm` under the old namespace (as documented) but will be published under a new namespace soon, with a major version bump.

React Native DateTimePicker

PASSED | platforms android | ios | windows | license MIT | Lean Core Extracted

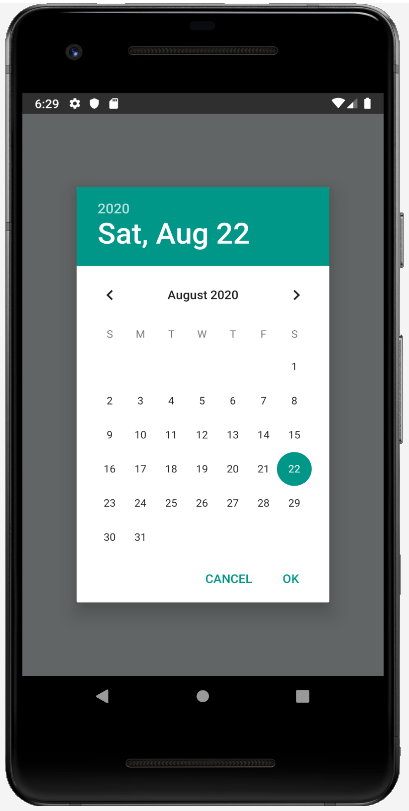
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

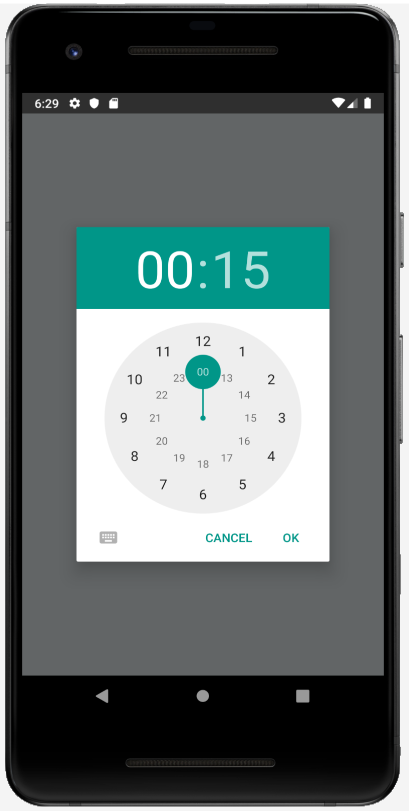
React Native date & time picker component for iOS, Android and Windows.

iOS	
	

Android	

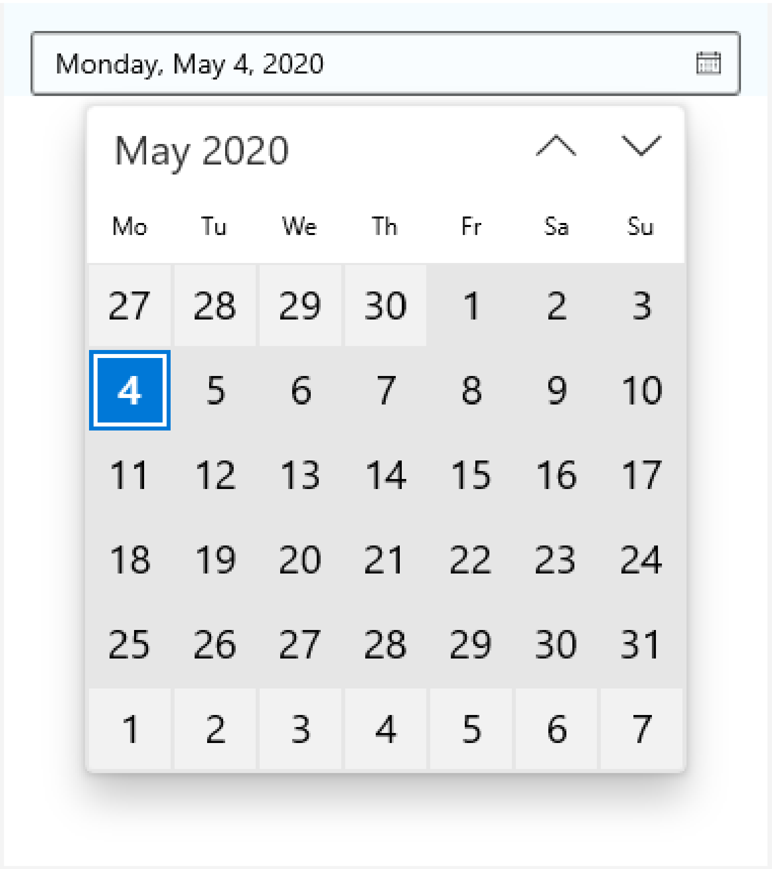
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD





Windows

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



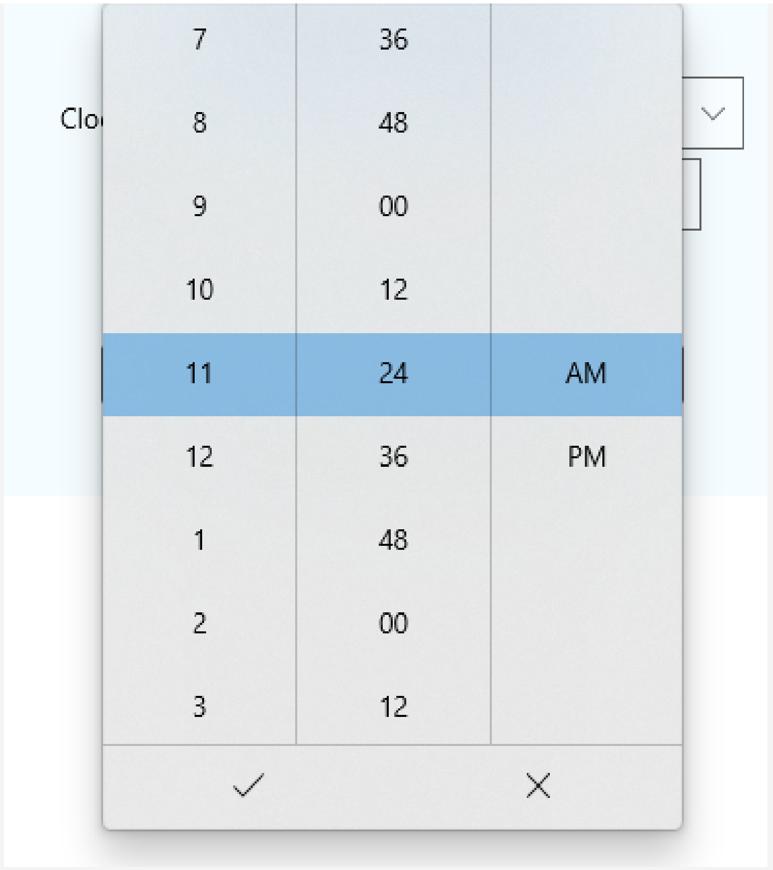


Table of Contents

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Requirements

- Xcode >= 11.6

Expo users notice

This module is part of Expo - **see docs**. However, Expo SDK may not contain the latest version of the module and therefore, the newest features and bugfixes may not be available in Expo. Use the command `expo install @react-native-community/datetimepicker` (not `yarn` or `npm`) to install this module - Expo will automatically install the latest version compatible with your Expo SDK (which may *not* be the latest version of the module available).

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Getting started

```
npm install @react-native-community/datetimepicker --save
```

or

```
yarn add @react-native-community/datetimepicker
```

Autolinking is not yet implemented on Windows, so **Manual installation** is needed.

RN >= 0.60

If you are using RN >= 0.60, only run `npx pod-install`. Then rebuild your project.

RN < 0.60

For RN < 0.60, you need to link the dependency using `react-native link`:

```
react-native link @react-native-community/datetimepicker
```

Then run `npx pod-install` and rebuild your project.

If this does not work, see **Manual installation**.

General Usage

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
import DateTimePicker from '@react-native-community/datetimepicker';
```

or

```
const DateTimePicker = require('@react-native-community/datetimepicker');
```

Basic usage with state

```
import React, {useState} from 'react';
import {View, Button, Platform} from 'react-native';
import DateTimePicker from '@react-native-community/datetimepicker';

export const App = () => {
  const [date, setDate] = useState(new Date(1598051730000));
  const [mode, setMode] = useState('date');
  const [show, setShow] = useState(false);

  const onChange = (event, selectedDate) => {
    const currentDate = selectedDate || date;
    setShow(Platform.OS === 'ios');
    setDate(currentDate);
  };

  const showMode = (currentMode) => {
    setShow(true);
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    setMode(currentMode);
  };

  const showDatepicker = () => {
    showMode('date');
  };

  const showTimepicker = () => {
    showMode('time');
  };

  return (
    <View>
      <View>
        <Button onPress={showDatepicker} title="Show date picker!" />
      </View>
      <View>
        <Button onPress={showTimepicker} title="Show time picker!" />
      </View>
      {show && (
        <DateTimePicker
          testID="dateTimePicker"
          value={date}
          mode={mode}
          is24Hour={true}
          display="default"
          onChange={onChange}
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
        />
      )}
    </View>
  );
};
```

Props

> Please note that this library currently exposes functionality from `UIDatePicker` on iOS and DatePickerDialog + TimePickerDialog on Android, and `CalendarDatePicker` +TimePicker on Windows.
>
> These native classes offer only limited configuration, while there are dozens of possible options you as a developer may need. It follows that if your requirement is not supported by the backing native views, this library will *not* be able to implement your requirement. When you open an issue with a feature request, please document if (or how) the feature can be implemented using the aforementioned native views. If those views do not support what you need, such feature requests will be closed as not actionable.

`mode` (`optional`)

Defines the type of the picker.

List of possible values:

- `"date"` (default for `iOS` and `Android` and `Windows`)
- `"time"`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- `"datetime"` (`iOS` only)
- `"countdown"` (`iOS` only)

```
<RNDateTimePicker mode="time" />
```

`display` (`optional`)

Defines the visual display of the picker. The default value is `"default"`.

List of possible values for Android

- `"default"` - Show a default date picker (spinner/calendar/clock) based on `mode` and Android version.
- `"spinner"`
- `"calendar"` (only for `date` mode)
- `"clock"` (only for `time` mode)

List of possible values for iOS (maps to preferredDatePickerStyle)

- `"default"` - Automatically pick the best style available for the current platform & mode.
- `"spinner"` - the usual appearance with a wheel from which you choose values
- `"compact"` - Affects only iOS 14 and later. Will fall back to "spinner" if not supported.
- `"inline"` - Affects only iOS 14 and later. Will fall back to "spinner" if not supported.

```
<RNDateTimePicker display="spinner" />
```

`onChange` (`optional`)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Date change handler.

This is called when the user changes the date or time in the UI. It receives the event and the date as parameters.

```
setDate = (event, date) => {};

<RNDateTimePicker onChange={this.setDate} />;
```

`value` (`required`)

Defines the date or time value used in the component.

```
<RNDateTimePicker value={new Date()} />
```

`maximumDate` (`optional`)

Defines the maximum date that can be selected. Note that on Android, this only works for `date` mode because **TimePicker** does not support this.

```
<RNDateTimePicker maximumDate={new Date(2300, 10, 20)} />
```

`minimumDate` (`optional`)

Defines the minimum date that can be selected. Note that on Android, this only works for `date` mode because **TimePicker** does not support this.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
<RNDateTimePicker minimumDate={new Date(1950, 0, 1)} />
```

`timeZoneOffsetInMinutes` (`optional`, `iOS only`)

Allows changing of the timeZone of the date picker. By default it uses the device's time zone.

```
// GMT+1
<RNDateTimePicker timeZoneOffsetInMinutes={60} />
```

`timeZoneOffsetInSeconds` (`optional`, `Windows only`)

Allows changing of the time zone of the date picker. By default it uses the device's time zone.

```
// UTC+1
<RNDateTimePicker timeZoneOffsetInSeconds={3600} />
```

`dayOfWeekFormat` (`optional`, `Windows only`)

Sets the display format for the day of the week headers. Reference: https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.calendarview.dayofweekformat?view=winrt-18362#remarks

```
<RNDateTimePicker dayOfWeekFormat={'{dayofweek.abbreviated(2)}'} />
```

`dateFormat` (`optional`, `Windows only`)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Sets the display format for the date value in the picker's text box. Reference: https://docs.microsoft.com/en-us/uwp/api/windows.globalization.datetimeformatting.datetimeformatter?view=winrt-18362#examples

```
<RNDateTimePicker dateFormat="dayofweek day month" />
```

`firstDayOfWeek` (`optional`, `Windows only`)

Indicates which day is shown as the first day of the week.

```
<RNDateTimePicker firstDayOfWeek={DAY_OF_WEEK.Wednesday} />
// The native parameter type is an enum defined in defined https://docs.microsoft.com/en-us/uwp/api/w
```

`textColor` (`optional`, `iOS only`)

Allows changing of the textColor of the date picker. Has effect only when `display` is `"spinner"`.

```
<RNDateTimePicker textColor="red" />
```

`locale` (`optional`, `iOS only`)

Allows changing of the locale of the component. By default it uses the device's locale.

```
<RNDateTimePicker locale="es-ES" />
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

`is24Hour` (`optional`, `Windows and Android only`)

Allows changing of the time picker to a 24 hour format. By default, this value is decided automatcially based on the user's chosen locale and other preferences.

```
<RNDateTimePicker is24Hour={true} />
```

`neutralButtonLabel` (`optional`, `Android only`)

Allows displaying neutral button on picker dialog. Pressing button can be observed in onChange handler as `event.type === 'neutralButtonPressed'`

```
<RNDateTimePicker neutralButtonLabel="clear" />
```

`minuteInterval` (`optional`)

The interval at which minutes can be selected. Possible values are: `1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30`

(On Windows, this can be any number between 0-59.)

on iOS, this in only supported when `display="spinner"`

```
<RNDateTimePicker minuteInterval={10} />
```

`style` (`optional`, `iOS only`)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Sets style directly on picker component. By default, the picker height is fixed to 216px.

Please note that by default, picker's text color is controlled by the application theme (light / dark mode). In dark mode, text is white and in light mode, text is black.

This means that eg. if the device has dark mode turned on, and your screen background color is white, you will not see the picker. Please use the `Appearance` api to adjust the picker's background color so that it is visible, as we do in the **example App** or **opt-out from dark mode**.

```
<RNDateTimePicker style={{flex: 1}} />
```

Migration from the older components

`RNDateTimePicker` is the new common name used to represent the old versions of iOS and Android.

On Android, open picker modals will update the selected date and/or time if the prop `value` changes. For example, if a HOC holding state, updates the `value` prop. Previously the component used to close the modal and render a new one on consecutive calls.

DatePickerIOS

- `initialDate` is deprecated, use `value` instead.

  ```
  // Before
  <DatePickerIOS initialValue={new Date()} />
  ```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
// Now
<RNDateTimePicker value={new Date()} />
```

- `date` is deprecated, use `value` instead.

```
// Before
<DatePickerIOS date={new Date()} />
```

```
// Now
<RNDateTimePicker value={new Date()} />
```

- `onChange` now returns also the date.

```
// Before
onChange = (event) => {};
<DatePickerIOS onChange={this.onChange} />;
```

```
// Now
onChange = (event, date) => {};
<RNDateTimePicker onChange={this.onChange} />;
```

- `onDateChange` is deprecated, use `onChange` instead.

```
// Before
setDate = (date) => {};
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
<DatePickerIOS onDateChange={this.setDate} />;
```

```
// Now
setDate = (event, date) => {};
<RNDateTimePicker onChange={this.setDate} />;
```

DatePickerAndroid

- `date` is deprecated, use `value` instead.

```
// Before
try {
  const {action, year, month, day} = await DatePickerAndroid.open({
    date: new Date(),
  });
} catch ({code, message}) {
  console.warn('Cannot open date picker', message);
}
```

```
// Now
<RNDateTimePicker mode="date" value={new Date()} />
```

- `minDate` and `maxDate` are deprecated, use `minimumDate` and `maximumDate` instead.

```
// Before
try {
  const {action, year, month, day} = await DatePickerAndroid.open({
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    minDate: new Date(),
    maxDate: new Date(),
  });
} catch ({code, message}) {
  console.warn('Cannot open date picker', message);
}
```

```
// Now
<RNDateTimePicker
  mode="date"
  minimumDate={new Date()}
  maximumDate={new Date()}
/>
```

- `dateSetAction` is deprecated, use `onChange` instead.

```
// Before
try {
  const {action, year, month, day} = await DatePickerAndroid.open();
  if (action === DatePickerAndroid.dateSetAction) {
    // Selected year, month (0-11), day
  }
} catch ({code, message}) {
  console.warn('Cannot open date picker', message);
}
```

```
// Now
setDate = (event, date) => {
  if (date !== undefined) {
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    // timeSetAction
  }
};
<RNDateTimePicker mode="date" onChange={this.setDate} />;
```

- `dismissedAction` is deprecated, use `onChange` instead.

```
// Before
try {
  const {action, year, month, day} = await DatePickerAndroid.open();
  if (action === DatePickerAndroid.dismissedAction) {
    // Dismissed
  }
} catch ({code, message}) {
  console.warn('Cannot open date picker', message);
}
```

```
// Now
setDate = (event, date) => {
  if (date === undefined) {
    // dismissedAction
  }
};
<RNDateTimePicker mode="date" onChange={this.setDate} />;
```

TimePickerAndroid

- `hour` and `minute` are deprecated, use `value` instead.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
// Before
try {
  const {action, hour, minute} = await TimePickerAndroid.open({
    hour: 14,
    minute: 0,
    is24Hour: false, // Will display '2 PM'
  });
  if (action !== TimePickerAndroid.dismissedAction) {
    // Selected hour (0-23), minute (0-59)
  }
} catch ({code, message}) {
  console.warn('Cannot open time picker', message);
}
```

```
// Now
// It will use the hour and minute defined in date
<RNDateTimePicker mode="time" value={new Date()} />
```

- `timeSetAction` is deprecated, use `onChange` instead.

```
// Before
try {
  const {action, hour, minute} = await TimePickerAndroid.open();
  if (action === TimePickerAndroid.timeSetAction) {
    // Selected hour (0-23), minute (0-59)
  }
} catch ({code, message}) {
  console.warn('Cannot open time picker', message);
}
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
// Now
setTime = (event, date) => {
  if (date !== undefined) {
    // Use the hour and minute from the date object
  }
};
<RNDateTimePicker mode="time" onChange={this.setTime} />;
```

- `dismissedAction` is deprecated, use `onChange` instead.

```
// Before
try {
  const {action, hour, minute} = await TimePickerAndroid.open();
  if (action === TimePickerAndroid.dismissedAction) {
    // Dismissed
  }
} catch ({code, message}) {
  console.warn('Cannot open time picker', message);
}
```

```
// Now
setTime = (event, date) => {
  if (date === undefined) {
    // dismissedAction
  }
};
<RNDateTimePicker mode="time" onChange={this.setTime} />;
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Contributing to the component

Please see **CONTRIBUTING.md**

Manual installation

iOS

1. Install CocoaPods, here the **installation guide**.

2. Inside the iOS folder run `pod init`, this will create the initial `pod` file.

3. Update your `pod` file to look like the following (Remember to replace `MyApp` with your target name):

```
# Allowed sources
source 'https://github.com/CocoaPods/Specs.git'

target 'MyApp' do
  # As we use Swift, ensure that `use_frameworks` is enabled.
  use_frameworks!

  # Specific iOS platform we are targetting
  platform :ios, '8.0'

  # Point to the installed version
  pod 'RNDateTimePicker', :path => '../node_modules/@react-native-community/datetimepicker/RNDateT
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
  # React/React-Native specific pods
  pod 'React', :path => '../node_modules/react-native', :subspecs => [
    'Core',
    'CxxBridge',      # Include this for RN >= 0.47
    'DevSupport',     # Include this to enable In-App Devmenu if RN >= 0.43
    'RCTText',
    'RCTNetwork',
    'RCTWebSocket',   # Needed for debugging
  ]

  # Explicitly include Yoga if you are using RN >= 0.42.0
  pod 'yoga', :path => '../node_modules/react-native/ReactCommon/yoga'

  # Third party deps podspec link
  pod 'DoubleConversion', :podspec => '../node_modules/react-native/third-party-podspecs/DoubleCon
  pod 'glog', :podspec => '../node_modules/react-native/third-party-podspecs/glog.podspec'
  pod 'Folly', :podspec => '../node_modules/react-native/third-party-podspecs/Folly.podspec'

end
```

4. Run `pod install` inside the same folder where the `pod` file was created

5. `npm run start`

6. `npm run start:ios`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Android

1. Add the following lines to `android/settings.gradle`:

```
include ':@react-native-community_datetimepicker'
project(':@react-native-community_datetimepicker').projectDir = new File(rootProject.projectDir, '
```

2. Add the compile line to the dependencies in `android/app/build.gradle`:

```
dependencies {
    ...
    implementation project(':@react-native-community_datetimepicker')
}
```

3. Add the import and link the package in `MainApplication.java`:

```
+ import com.reactcommunity.rndatetimepicker.RNDateTimePickerPackage;

public class MainApplication extends Application implements ReactApplication {

  @Override
  protected List<ReactPackage> getPackages() {
    @SuppressWarnings("UnnecessaryLocalVariable")
    List<ReactPackage> packages = new PackageList(this).getPackages();
    // Packages that cannot be autolinked yet can be added manually here, for example:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
+    packages.add(new RNDateTimePickerPackage());
     return packages;
   }
 }
```

Windows

Add the DateTimePickerWindows project to your solution

1. Open the solution in Visual Studio 2019
2. Right-click solution icon in Solution Explorer > Add > Existing Project Select 'D:\pathToYourApp\node_modules@react-native-community\datetimepicker\windows\DateTimePickerWindows\DateTimePickerWindows.vcxproj'

windows/myapp.sln

Add a reference to `DateTimePickerWindows` to your main application project. From Visual Studio 2019:

Right-click main application project > Add > Reference... Check 'DateTimePickerWindows' from the 'Project > Solution' tab on the left.

pch.h

Add `#include "winrt/DateTimePicker.h"`.

app.cpp

Add `PackageProviders().Append(winrt::DateTimePicker::ReactPackageProvider());` before `InitializeComponent();`.

Running the example app

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

1. Install required pods in `example/ios` by running `npx pod-install`
2. Run `npm start` to start Metro Bundler
3. Run `npm run start:ios` or `npm run start:android` or `npm run start:windows` (or `yarn run start:windows`)

Keywords

react-native-component react-native ios android windows datepicker timepicker datetime

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Advisories

Press

Status

Contact npm

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD