

# How to manage date and time in React using Moment.js

Moment.js is a great library for efficient date and time management in React apps. Here are detailed instructions.

Picture 1 of How to manage date and time in React using Moment.js

**Managing date & time in React** can be a challenge for beginners. Fortunately, there are several libraries that help you manage dates and times in React. One of them is Moment.js.

Moment.js is a lightweight library that provides a simple solution for manipulating and formatting dates & times in JavaScript.

## Install the Moment.js . library

The Moment.js library is a package that manages date & time operations in JavaScript. Installing the Moment.js library is the first step to using it in a React app. You can do this by running the following command in the terminal.

```
npm install moment
```

Once the installation is complete, you can use Moment.js in React elements.

## Use Moment.js to display date and time

You can use **Moment.js** to display dates and times in a specific format in your React app. To use this library, import **moment** from the installed package.

```
import React from 'react'; import moment from 'moment'; function App() { ??
const date = moment().format("MMMM DD YYYY"); ??
const time = moment().format("HH mm ss"); ??return ( ????? ???????
```

```
Today's date is { date }
```

```
??????
```

```
The time is { time }
```

```
???? ??) } export default App
```

**The moment()** method creates a new moment object, representing a specific point in time. **The format()** method formats a moment object as a string.

**The format()** method takes a string argument that specifies the desired format for the moment object. String arguments can include a combination of letters and special characters. Each character has its own meaning.

Examples of some special characters:

1. YYYY: Four-digit year
2. YY: Two-digit year
3. MM: Two-digit month
4. M: One- or two-digit month
5. MMMM: Month in words
6. DD: Day of the month with two digits
7. D: Day of the month with one or two digits
8. Do: Day of the month with ordinal
9. HH: Two-digit hour
10. Q: One or two digit hour
11. mm: Two-digit minute
12. m: One- or two-digit minute
13. ss: Two-digit number of seconds
14. s: Number of seconds with one or two digits

**When the App** element in the previous code block is output, the current date & time will appear in the specified format on the screen.

**The moment()** method can take a date or time argument string. When moment() has this argument, it creates a moment object representing the date or time. This string can be in various formats such as ISO 8601, RFC 2822 or Unix timestamps.

For example:

```
const date = moment('1998-06-23').format("MMMM DD YYYY");
```

## Use Moment.js to manipulate dates and times

The Moment.js library also provides several methods for manipulating month and time snoring. They allow you to add or subtract time periods, set specific values ??for date and time elements, and perform other applicable operations.

For example:

```
import React from 'react'; import moment from 'moment'; function App() { ??
const tomorrow = moment().add(1, 'day').format("Do MMMM,YYYY"); ??
const time = moment().subtract(1, 'hour').format("HH:mm:ss"); ??
const lastYear = moment().set('year', 2022).set('month', 1).format("Do MMMM,YYYY");
??const hour = moment().get('hour'); ??return ( ???? ??????)
```

Tomorrow's date is { tomorrow }

??????

This was the time: { time }, an hour ago

```
??????
```

```
{ lastYear }
```

```
??????
```

```
{ hour }
```

```
???? ??) } export default App
```

In this example, you declare the following JavaScript variables: **tomorrow** , **time** , **lastYear** , and **hour** . These 4 variables use different methods of the Moment.js library to manipulate dates and times.

**The tomorrow** variable uses **the add()** method to add a date to the current date. **Add()** adds the time to the Moment object. This function takes two arguments: a time value and a string representing the added time unit. The units can be **years** , **months** , **weeks** , **days** , **hours** , **minutes** and **seconds** .

You can also reverse time with **subtract()** . In this case, the time variable uses **subtract()** to subtract 1 hour from the current time.

**Using the set()** method , the Last year variable sets the year to 2022 and the month to February. **set()** assigns a specific time unit to the Moment object.

With the **get()** method, you can retrieve a specific time. **Get()** computes one argument - the time unit.

## Use Moment.js to parse date and time

Another useful feature of Moment.js is the ability to parse dates and times from strings. This can be useful when working with data from external sources.

To parse a date or time from a string, you need to use **moment()** . Here, **moment()** computes two arguments, a date string and a format string.

Example of using **moment()** to parse date and time:

```
import React from 'react'; import moment from 'moment'; function App() { ??  
const date = moment('2920130000', 'Do-MMMM-YYYY').toDate(); ??  
console.log( date ); ??return ( ????? ??) } export default App
```

In the above code block, **moment()** will parse this string '2920130000' using the format string "Do-MMMM-YYYY". **toDate()** converts the moment object to a standard JavaScript date object.

**toDate()** takes no arguments and returns a JavaScript Date object representing the same date and time as the moment object.

Overall, **Moment.js** is a powerful library that provides a simple way to manipulate and format dates and times in JavaScript. Hope the article is useful to you.

You finished reading the article "**How to manage date and time in React using Moment.js**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

---