

# Hantera datum och tid i JavaScript med Moment.js

## Inledning

Moment.js är ett JavaScript bibliotek som kan validera, analysera och manipulera datum och tid. Det fungerar i de flesta webbläsare och i Node.js (Sanders, L., Cambron, I., Johnson-Pint, M., Marwaha, K., Johnson-Pint, M. and Wood, T., 2019, /docs/#/use-it). Det är relativt enkelt att använda Moment, man behöver inte skriva speciellt mycket kod för att få ut det man har tänkt sig när det gäller datum och tid. Man sparar tid och får en mycket kortare kod jämfört med om man inte skulle använda sig av biblioteket. Det är alltså många fördelar med att använda sig av Moment.

## Bakgrund

För ca 9 år sedan, inspirerad av enkelheten och användbarheten av Underscore.js och Underscore.string, skapade Tim Wood en datumformaterare och ett timeago plugin och kallade det för Underscore.date. Efter att ha arbetat med projektet under en tid så utvecklades det till Moment.js (Wood, 2016).

Med hjälp av ett hundratal committers, på GitHub, och en stor mängd bugrapporter, plugins och idéer har Moment växt långt bortom vad Tim Wood hade förväntat sig när han startade projektet. Det är väl testat och dokumenterat och har blivit ett av de 10 mest populära JavaScript biblioteken. I början av 2013 lämnade Tim Wood över projektledarrollen till Iskren Chernev för att spendera mer tid med sin nyfödda dotter(Wood, n.d.).

## Användning

Datum och tid är vanligt förekommande i applikationer. Man kan exempelvis använda datum för att spåra skapandet av ett objekt, man kan ta reda på hur länge sedan det var ett event ägde rum eller spara datumet för ett kommande event. Eftersom datum inte är så enkelt att jobba med så är det viktigt att ha ett JavaScript bibliotek som är exakt och har ett enkelt gränssnitt (Robinson, 2016).

Moment har ett starkt stöd för internationalisering vilket gör att programvaran kan anpassas till olika språk. Som standard används engelska (USA) men det finns många olika språk och tidszoner att välja mellan (Sanders et al., 2019).

Det går som jag tidigare nämnt att använda Moment både i webbläsare och i Node.js. Nedan kan man se olika kodexempel för att introducera ett nytt projekt (Sanders et al., 2019, #/use-it).

## Webbläsare

```
<script src="moment.js"></script>
<script>
    moment().format();
</script>
```

## CDN-länk:

```
<script>https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.24.0/moment-with-locales.min.js</script>
```

## Node.js

```
Npm install moment
var moment = require('moment');
moment().format();
```

För att välja språk använder man sig av `moment.locale`. Om man exempelvis väljer svenska som språk ser koden ut så här (Sanders et al., 2019):

```
moment.locale('sv');
moment().format('LT'); // 13:26
moment().format('LL'); // 23 februari 2020
moment().format('ll'); // 23 feb 2020
moment().format('LLLL'); // söndag 23 februari 2020 kl. 13:27
moment().format('llll'); // sön 23 feb 2020 13:27
```

Moment erbjuder flera sätt att visa tid på exempelvis den grundläggande momentfunktionen **moment()**, **moment.utc** och **moment.parseZone** (Sanders et al., 2019, /guides/#/parsing).

Om man vill visa ett datum eller en tidsangivelse som stämmer överens med användarens lokala tid använder man följande kod:

```
moment('2020-01-01T23:35:01').format('LLL'); //1 januari 2020 kl. 23:35
```

Vill man använda sig av Koordinerad universell tid (UTC) ser koden ut så här:

```
moment.utc('2020-01-01T23:35:01').format('LLL'); // 1 januari 2020 kl. 23:35
```

Vid en tidszonsförskjutning så används:

```
moment.parseZone("2020-01-01T00:00:00-13:00").format('LLL'); // 1 januari 2020  
kl. 00:00
```

Nedan är exempel på hur man kan skriva ut datum och tid på olika sätt beroende på hur mycket information man vill ha med (Sanders et al., 2019).

```
moment().format('MMMM Do YYYY, h:mm:ss a'); // februari 23e 2020, 1:20:51 pm  
moment().format('dddd'); // söndag  
moment().format("MMM Do YY"); // feb 23e 20
```

Det går även att addera och subtrahera med tid vilket man kan se i koden nedan (Sanders et al., 2019).

```
moment("20120620", "YYYYMMDD").fromNow(); // för 8 år sedan  
moment().startOf('day').fromNow(); // för 13 timmar sedan  
moment().endOf('day').fromNow(); // om 11 timmar  
moment().startOf('hour').fromNow(); // för 24 minuter sedan  
moment().calendar(); // Idag 13:25  
moment().subtract(10, 'days').calendar(); // 2020-02-13
```

Det finns mycket mer man kan göra med tid och datum genom att använda Moment, jag har plockat ut det mest grundläggande. Om man går in på **momentjs.com** kan man lära sig mer.

## Diskussion

Det finns många fördelar med ett JavaScript bibliotek som hanterar tid och datum. Exempelvis sparar det tid och man får en mycket kortare och enklare kod, det är en fördel att det går att använda både i webbläsare och i Node.js och det är bra att det stödjer internationalisering och olika tidszoner.

Det finns såklart även nackdelar med Moment. Enligt (Inventi.studio, 2019) är Moment tungt och långsamt om man jämför det med det med Day.js, Luxon, JS-Joda, Date-fns och Native-API som är liknande JavaScript bibliotek.

Skillnaden på React och Moment är ganska stor. Moment manipulerar tid och datum genom att man skriver några rader kod medan man med React kan skapa en applikation som man kan utveckla exempelvis hemsidor med. Det går bra att kombinera Moment med olika ramverk som exempelvis React. Man kan relativt enkelt inkludera Moment i sin Reactapp.

## Framtid

Jag tror att JavaScript bibliotek och Ramverk kommer att fortsätta att vara väldigt uppskattat eftersom det underlättar så mycket för användarna. Moment kommer säkert att fortsätta utvecklas. Det finns redan många plugin som man kan ladda ner och det kommer säkert att bli fler med tiden. Eftersom Moment är så enkelt att använda tror jag att det kommer att förbli populärt.

## Referenslista

Wood, T. (2016). *moment().endOf('term')*. [online] Medium. Available at: <https://medium.com/@timrwood/moment-endof-term-522d8965689> [Accessed 19 Feb. 2020].

Wood, T. (2011). *moment/moment*. [online] GitHub. Available at: <https://github.com/moment/moment/blob/develop/CONTRIBUTING.md> [Accessed 19 Feb. 2020].

Sanders, L., Cambron, I., Johnson-Pint, M., Marwaha, K., Johnson-Pint, M. and Wood, T. (2019). *Moment.js | Home*. [online] Available at: <https://momentjs.com/> [Accessed 19 Feb. 2020].

Inventi.studio. (2019). *Why you shouldn't use Moment.js....* [online] Available at: <https://inventi.studio/en/blog/why-you-shouldnt-use-moment-js> [Accessed 21 Feb. 2020].

Wood, T. (n.d.). *Moment.js | by Tim Wood*. [online] Timwood.work. Available at: <https://timwood.work/work/moment/> [Accessed 22 Feb. 2020].