



Express Body Parser

#Node.js Express Notes

Express Body Parser



What is bodyParser ?

- When setting up a HTTP web server, one common need is to be able to receive data sent by the server clients.
- Body-parser is the Node.js body parsing middleware.
- It is responsible for parsing the incoming request bodies in a middleware before you handle it.



JSON body

- HTTP POST requests and then access the parsed JSON body from a request sent by a client.
- <https://expressjs.com/en/4x/api.html#express.json>
- `application.use(express.json());`



URL encoded body

- <https://expressjs.com/en/4x/api.html#express.urlencoded>
- `application.use(express.urlencoded({extended:true}));`
- `console.log(req.body);`



Install NPM


- Install bcrypt Package
- <https://www.npmjs.com/package/bcrypt>
- `npm install --save body-parser`



Code

```
const express = require('express')  
const bodyParser = require('body-parser');  
const app = express()  
  
// support parsing of application/json type post data  
app.use(bodyParser.json());  
  
//support parsing of application/x-www-form-urlencoded post data  
app.use(bodyParser.urlencoded({ extended: true }));
```

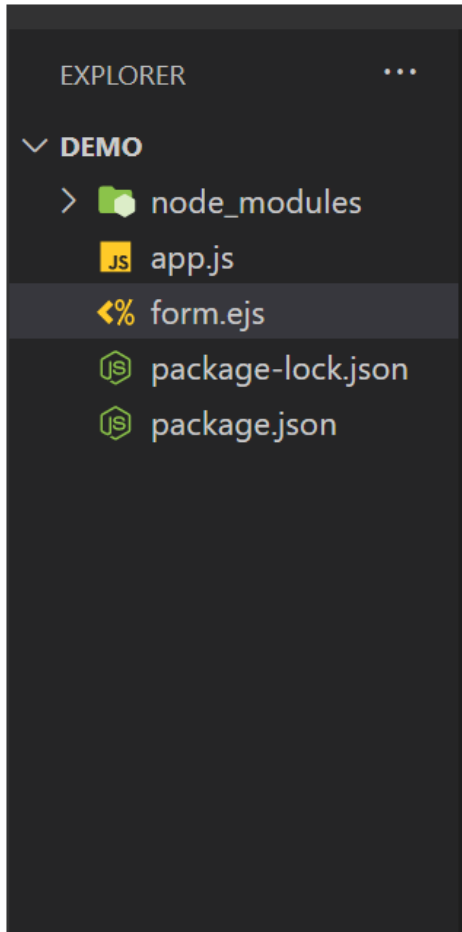




```
File Edit Selection View Go Run Terminal Help app.js - demo - Visual Studio Code

EXPLORER
  DEMO
    > node_modules
    app.js
    package-lock.json
    package.json

app.js
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    |   res.send('Hello World!')
7    | })
8
9  app.listen(port, () => {
10   |   console.log(`Example app listening at http://localhost:${port}`)
11   | })
12
13
14
```

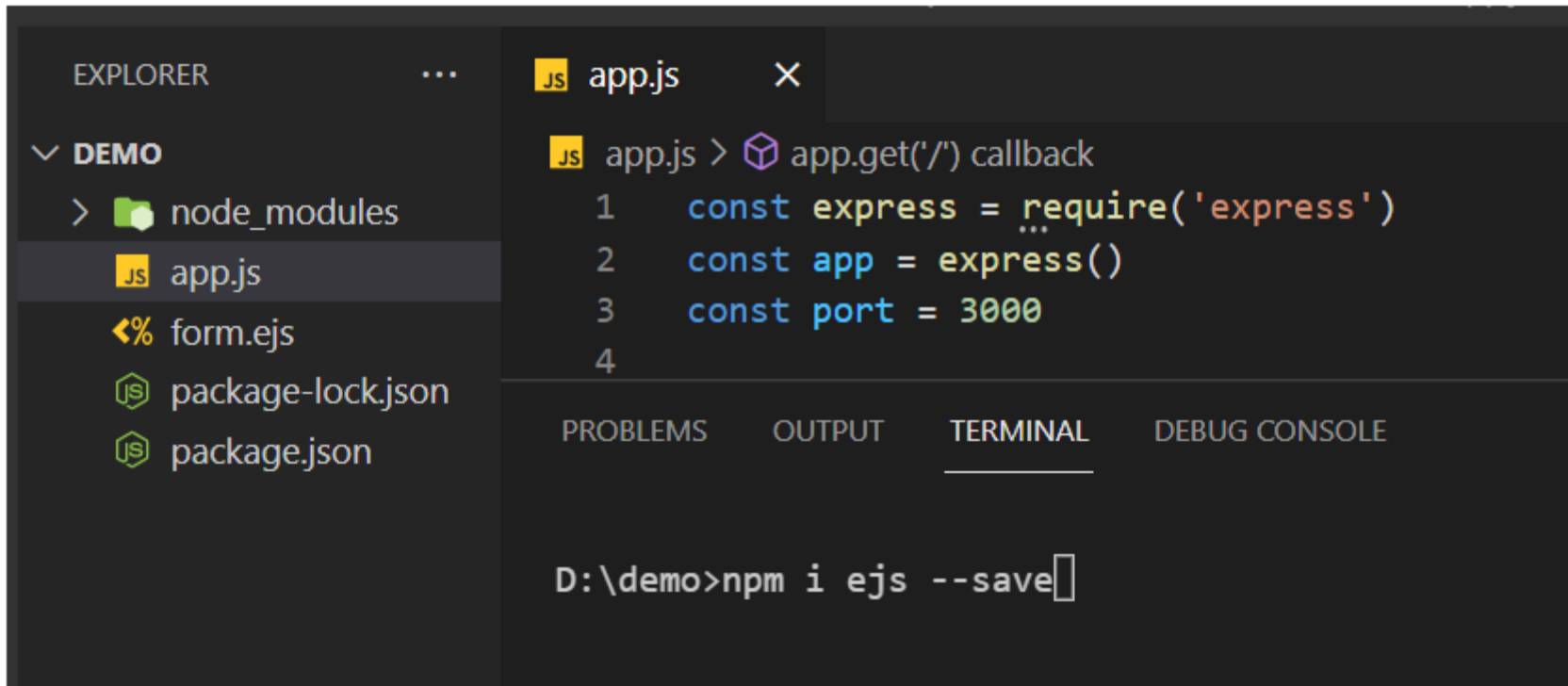




```

<% form.ejs > ...
1  <html>
2      <body>
3          <form method="post" action="form-process">
4              Name:<input type="text" name="txt1" placeholder="Enter Name"/>
5              Contact Number:<input type="text" name="txt2" placeholder="Enter Contact No"/>
6              <input type="submit" />
7          </form>
8      </body>
9  </html>
10

```





EXPLORER

▼ DEMO


- > node_modules
- app.js
- form.ejs
- package-lock.json
- package.json

app.js

```
app.js > app.get('/') callback
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

D:\demo>npm i ejs --save



EXPLORER

✓ DEMO

- > node_modules
- JS app.js
- form.ejs
- package-lock.json
- package.json

JS app.js > app.get('/') callback

```
1 const express = require('express')
2 const app = express()
3 const port = 3000
4
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE


```
D:\demo>npm i ejs --save

up to date, audited 73 packages in 1s

2 packages are looking for funding
  run `npm fund` for details

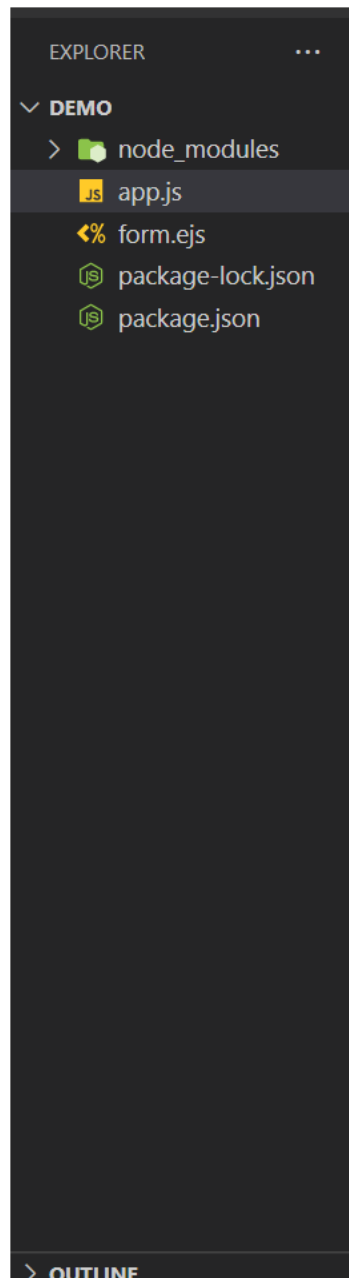
found 0 vulnerabilities

D:\demo>npm i body-parser --save
```



```
EXPLORER    ...    JS app.js    X
  DEMO
  > node_modules
    JS app.js
    % form.ejs
    package-lock.json
    package.json

JS app.js > ...
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  const bodyparser = require('body-parser')
6  const path = require('path')
7
8  // View Engine Setup
9  app.set("views", path.join(__dirname))
10 app.set("view engine", "ejs")
11 console.log(path.join(__dirname));
12
13 // Body-parser middleware
14 app.use(bodyparser.urlencoded({extended:false}))
15 app.use(bodyparser.json())
16
17 app.get('/', (req, res) => {
18   res.send('Hello World!')
19 })
20
21 app.listen(port, () => {
22   console.log(`Example app listening at http://localhost:\${port}`)
23 })
24
```



```
app.js > app.get('/') callback
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  const bodyparser = require('body-parser')
6  const path = require('path')
7
8  // View Engine Setup
9  app.set("views", path.join(__dirname))
10 app.set("view engine", "ejs")
11 console.log(path.join(__dirname));
12
13 // Body-parser middleware
14 app.use(bodyparser.json())
15 app.use(bodyparser.urlencoded({extended:false}))
16
17 app.get('/', (req, res) => {
18   res.send('Hello World!')
19 })
20
21 app.get('/form', (req, res) => {
22   res.render('form')
23 })
24
25 app.post('/form-process', (req, res) => {
26   console.log('Form Data ' + req.body.txt1 + req.body.txt2);
27 })
28
29 app.listen(port, () => {
30   console.log(`Example app listening at http://localhost:${port}`)
31 })
32
```



127.0.0.1:3000/form x +

← → ↻ 🏠 ⓘ 127.0.0.1:3000/form

Name: Contact Number:

127.0.0.1:3000/form x +

← → ↻ 🏠 ⓘ 127.0.0.1:3000/form

Name: Contact Number:

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node app.js
D:\demo
Example app listening at http://localhost:3000
Form Data Akash123456
█
```



Moment



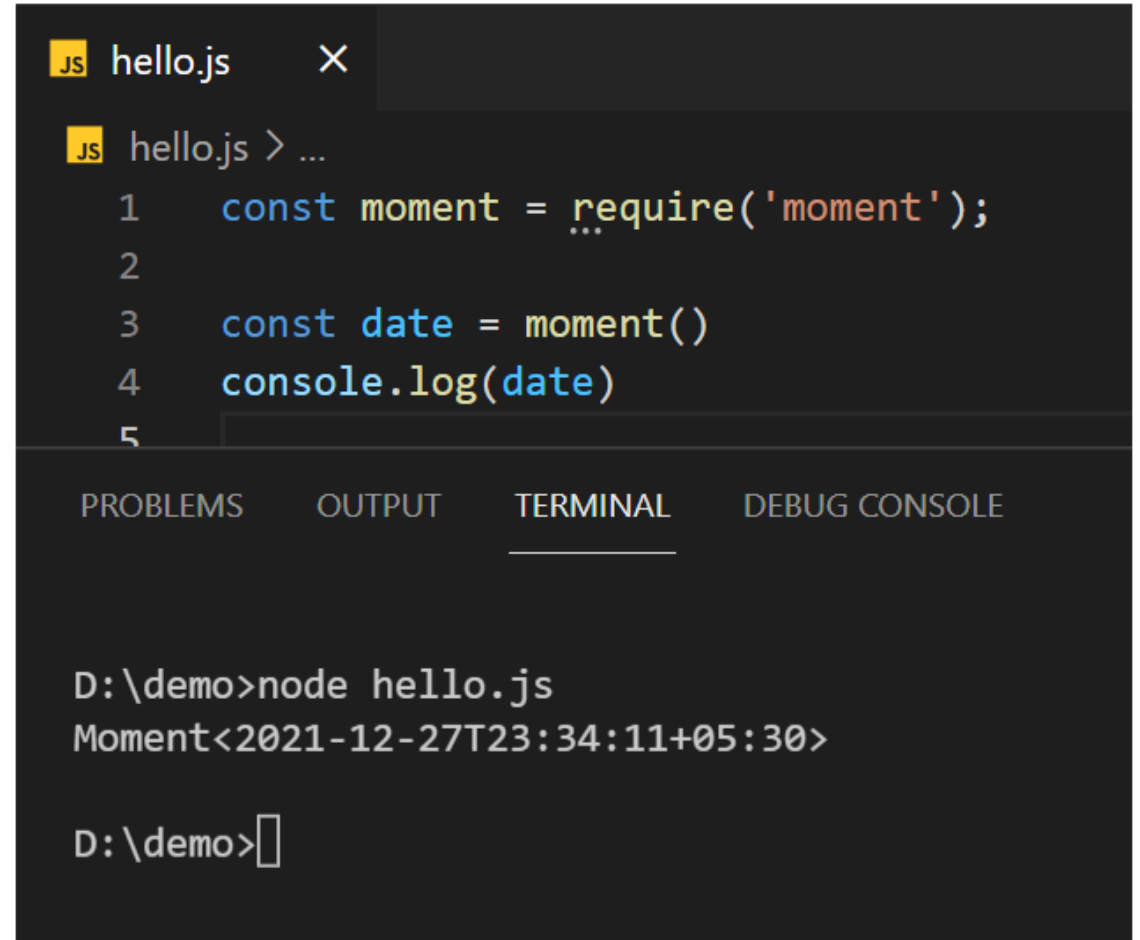
Moement

- [Moment.js](#) is an awesome JavaScript library that helps you manage dates, in the browser and in Node.js as well.



Get the current date and time

- `const moment = require('moment');`
- `const date = moment();`
- `console.log(date)`



```
JS hello.js  X
JS hello.js > ...
1  const moment = require('moment');
2
3  const date = moment()
4  console.log(date)
5

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>node hello.js
Moment<2021-12-27T23:34:11+05:30>

D:\demo>
```



Creating Moment.js objects

- We can use several ways to create date and time Moment.js objects. These objects have to be formatted later to human-readable format.

```
JS hello.js > [?] d3
1  const moment = require('moment');
2  //object from a string
3      let d1 = moment("2018-06-03");
4      console.log(d1.format('llll'));
5  //object is created from an array
6      let d2 = moment([2017, 11, 23]);
7      console.log(d2.format('llll'));
8  // JSON objects to create moment objects
9      let d3 = moment({ year :2010, month :3, day :5,
10         hour :15, minute :10, second :3, millisecond :123});
11      console.log(d3.format('llll'));
12  //unix time stamp (in milliseconds) to define a moment object
13      let d4 = moment(1530471537000);
14      console.log(d4.format('llll'));
15  //JavaScript built-in Date object to define a moment object
16      let d5 = moment(new Date(2011, 11, 22));
17      console.log(d5.format('llll'));
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
Sun, Jun 3, 2018 12:00 AM
Sat, Dec 23, 2017 12:00 AM
Mon, Apr 5, 2010 3:10 PM
Mon, Jul 2, 2018 12:28 AM
Thu, Dec 22, 2011 12:00 AM
```



Example

```
const moment = require('moment');  
//object from a string  
let d1 = moment("2018-06-03");  
console.log(d1.format('llll'));  
//object is created from an array  
let d2 = moment([2017, 11, 23]);  
console.log(d2.format('llll'));  
// JSON objects to create moment objects  
let d3 = moment({ year :2010, month :3, day :5,  
hour :15, minute :10, second :3, millisecond :123});  
console.log(d3.format('llll'));  
//unix time stamp (in milliseconds) to define a moment object  
let d4 = moment(1530471537000);  
console.log(d4.format('llll'));  
//JavaScript built-in Date object to define a moment object  
let d5 = moment(new Date(2011, 11, 22));  
console.log(d5.format('llll'));
```



- `const moment = require('moment');`
- `var currentDate1 = moment();`
- `console.log('Current Date 1: ' + currentDate1);`
- `var currentDate2 = moment().format();`
- `console.log('Current Date 2: ' + currentDate2);`
- `var currentDate3 = moment().format('DD/MM/YYYY');`
- `console.log('Current Date 3: ' + currentDate3);`
- `var currentDate4 = moment().unix();`
- `console.log('Current Date 4: ' + currentDate4);`
- `var currentDate5 = moment().toISOString();`
- `console.log('Current Date 5: ' + currentDate5);`

```
JS hello.js X
JS hello.js > ...
1  const moment = require('moment');
2  var currentDate1 = moment();
3  console.log('Current Date 1: ' + currentDate1);
4
5  var currentDate2 = moment().format();
6  console.log('Current Date 2: ' + currentDate2);
7
8  var currentDate3 = moment().format('DD/MM/YYYY');
9  console.log('Current Date 3: ' + currentDate3);
10
11 var currentDate4 = moment().unix();
12 console.log('Current Date 4: ' + currentDate4);
13
14 var currentDate5 = moment().toISOString();
15 console.log('Current Date 5: ' + currentDate5);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
Current Date 1: 1640628922371
Current Date 2: 2021-12-27T23:45:22+05:30
Current Date 3: 27/12/2021
Current Date 4: 1640628922
Current Date 5: 2021-12-27T18:15:22.380Z
```



Moment.js datetime parts

- `const moment = require('moment');`
- `let now = moment();`
- `let year = now.get('year');`
- `let month = now.get('month');` // 0 to 11
- `let date = now.get('date');`
- `let hour = now.get('hour');`
- `let minute = now.get('minute');`
- `let second = now.get('second');`
- `let millisecond = now.get('millisecond');`
- `console.log("Year: " + year);`
- `console.log("Month: " + month);`
- `console.log("Date: " + date);`
- `console.log("Hour: " + hour);`
- `console.log("Minute: " + minute);`
- `console.log("Second: " + second);`
- `console.log("Millisecond: " + millisecond);`

```
JS hello.js > ...
1  const moment = require('moment');
2  let now = moment();
3
4  let year = now.get('year');
5  let month = now.get('month'); // 0 to 11
6  let date = now.get('date');
7  let hour = now.get('hour');
8  let minute = now.get('minute');
9  let second = now.get('second');
10 let millisecond = now.get('millisecond');
11
12 console.log("Year: " + year);
13 console.log("Month: " + month);
14 console.log("Date: " + date);
15 console.log("Hour: " + hour);
16 console.log("Minute: " + minute);
17 console.log("Second: " + second);
18 console.log("Millisecond: " + millisecond);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
Year: 2021
Month: 11
Date: 28
Hour: 8
Minute: 46
Second: 15
Millisecond: 730
```



Moment.js day of week, month, year

- `const moment = require('moment');`
- `let now = moment();`
- `console.log("Day of week: " + now.weekday());`
- `console.log("Day of month: " + now.date());`
- `console.log("Day of year: " + now.dayOfYear());`
- `console.log("Quarter of year: " + now.quarter());`
- `console.log("Weeks in year: " + now.weeksInYear());`

```
JS hello.js > ...
1  const moment = require('moment');
2  let now = moment();
3  console.log("Day of week: " + now.weekday());
4  console.log("Day of month: " + now.date());
5  console.log("Day of year: " + now.dayOfYear());
6
7  console.log("Quarter of year: " + now.quarter());
8  console.log("Weeks in year: " + now.weeksInYear());
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
Day of week: 2
Day of month: 28
Day of year: 362
Quarter of year: 4
Weeks in year: 52
```



Moment.js localized date and time

```
const moment = require('moment');
```

```
moment.locale('en');
```

```
let now = moment();
```

```
console.log(now.format('LLLL'));
```

```
moment.locale('gu');
```

```
now = moment();
```

```
console.log(now.format('LLLL'));
```

```
moment.locale('hi');
```

```
now = moment();
```

```
console.log(now.format('LLLL'));
```

```
JS hello.js > ...
1  const moment = require('moment');
2
3  moment.locale('en');
4  let now = moment();
5  console.log(now.format('LLLL'));
6
7  moment.locale('gu');
8  now = moment();
9  console.log(now.format('LLLL'));
10
11 moment.locale('hi');
12 now = moment();
13 console.log(now.format('LLLL'));
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
Tuesday, December 28, 2021 8:51 AM
मंगलवार, २८ दिसम्बर २०२१, सां. ८:५१ बजे
मंगलवार, २८ दिसम्बर २०२१, सुबह ८:५१ बजे
```



Parse a date

- A moment object can be initialized with a date by passing it a string:
- `const date = moment(string)`



Date

d	0..6	Weekday
dd	Su	
ddd	Sun	
dddd	Sunday	
YY	13	Year
YYYY	2013	
M	1..12 (Jan is 1)	Month
Mo	1st..12th	
MM	01..12 (Jan is 1)	
MMM	Jan	
MMMM	January	
Q	1..4	Quarter
Qo	1st..4th	
D	1..31	Day
Do	1st..31st	
DD	01..31	
DDD	1..365	Day of year
DDDo	1st..365th	
DDDD	001..365	
w	1..53	Week of year
wo	1st..53rd	
ww	01..53	



Time

H	0..23	24h hour
HH	00..23	
h	1..12	12h hour
hh	01..12	
m	0..59	Minutes
mm	00..59	
s	0..59	Seconds
ss	00..59	
a	am	AM/PM
A	AM	
Z	+07:00	Timezone offset
ZZ	+0730	
S	0..9	Deciseconds
SS	00..99	Centiseconds
SSS	000..999	Milliseconds
X		Unix timestamp
x		Millisecond Unix timestamp



Presets

LT	8:30 PM
LTS	8:30:25 PM
LL	August 2 1985
II	Aug 2 1985
LLL	August 2 1985 08:30 PM
III	Aug 2 1985 08:30 PM
LLLL	Thursday, August 2 1985 08:30 PM
IIII	Thu, Aug 2 1985 08:30 PM



Cheat Sheet

- <https://devhints.io/moment>



Date Format

```
const moment = require('moment');
```

```
let m = moment().format('D-M-YYYY hh:mm A ');
```

```
console.log(m);
```

```
let m1 = moment().format('dddd, MMMM Do YYYY');
```

```
console.log(m1);
```

```
JS hello.js  X
JS hello.js > ...
1  const moment = require('moment');
2
3  let m = moment().format('D-M-YYYY hh:mm A ');
4  console.log(m);
5
6  let m1 = moment().format('dddd, MMMM Do YYYY');
7  console.log(m1);

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>node hello.js
27-12-2021 11:12 PM
Monday, December 27th 2021

D:\demo>node hello.js
27-12-2021 11:12 PM
Monday, December 27th 2021

D:\demo>
```



Validating a date

- Any date can be checked for validity using the isValid() method:

```
const moment = require('moment');
```

```
let m = moment('2021-13-23').isValid()  
console.log(m); //false
```

```
let m1 = moment('2021-11-23').isValid()  
console.log(m1); //true
```

```
JS hello.js X  
JS hello.js > [?] m1  
1  const moment = require('moment');  
2  
3  let m = moment('2021-13-23').isValid()  
4  console.log(m); //false  
5  
6  let m1 = moment('2021-11-23').isValid()  
7  console.log(m1); //true  
  
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE  
  
D:\demo>node hello.js  
false  
true
```



Time ago, time until date

- Use `fromNow()`. Strings are localized:
- `const moment = require('moment');`
- `let m = moment('2015-05-23').fromNow();`
`console.log(m); // 7 Years Ago`
- `let m1 = moment('2021-11-27').fromNow();`
`console.log(m1); // A Month Ago`

```
JS hello.js X
JS hello.js > ...
1  const moment = require('moment');
2
3  let m = moment('2015-05-23').fromNow();
4  console.log(m); // 7 Years Ago
5
6  let m1 = moment('2021-11-27').fromNow();
7  console.log(m1); // A Month Ago
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
7 years ago
a month ago
```



Manipulate a date

- You can add or subtract any amount of time to a date:
- You can use those values:
 - years
 - quarters
 - months
 - weeks
 - days
 - hours
 - minutes
 - seconds
 - milliseconds



Manipulate a date

- `const moment = require('moment');`
- `let m = moment('2021-11-23').add(1, 'week');`
- `console.log(m);`
- `let m1 = moment('2021-11-25').subtract(1, 'month');`
- `console.log(m1);`

```
JS hello.js X
JS hello.js > [?] m
1  const moment = require('moment');
2
3  let m = moment('2021-11-23').add(1, 'week')
4  console.log(m);
5
6  let m1 = moment('2021-11-25').subtract(1, 'month')
7  console.log(m1);

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>node hello.js
Moment<2021-11-30T00:00:00+05:30>
Moment<2021-10-25T00:00:00+05:30>
```



Difference Between Two Dates In Years, Months, Days

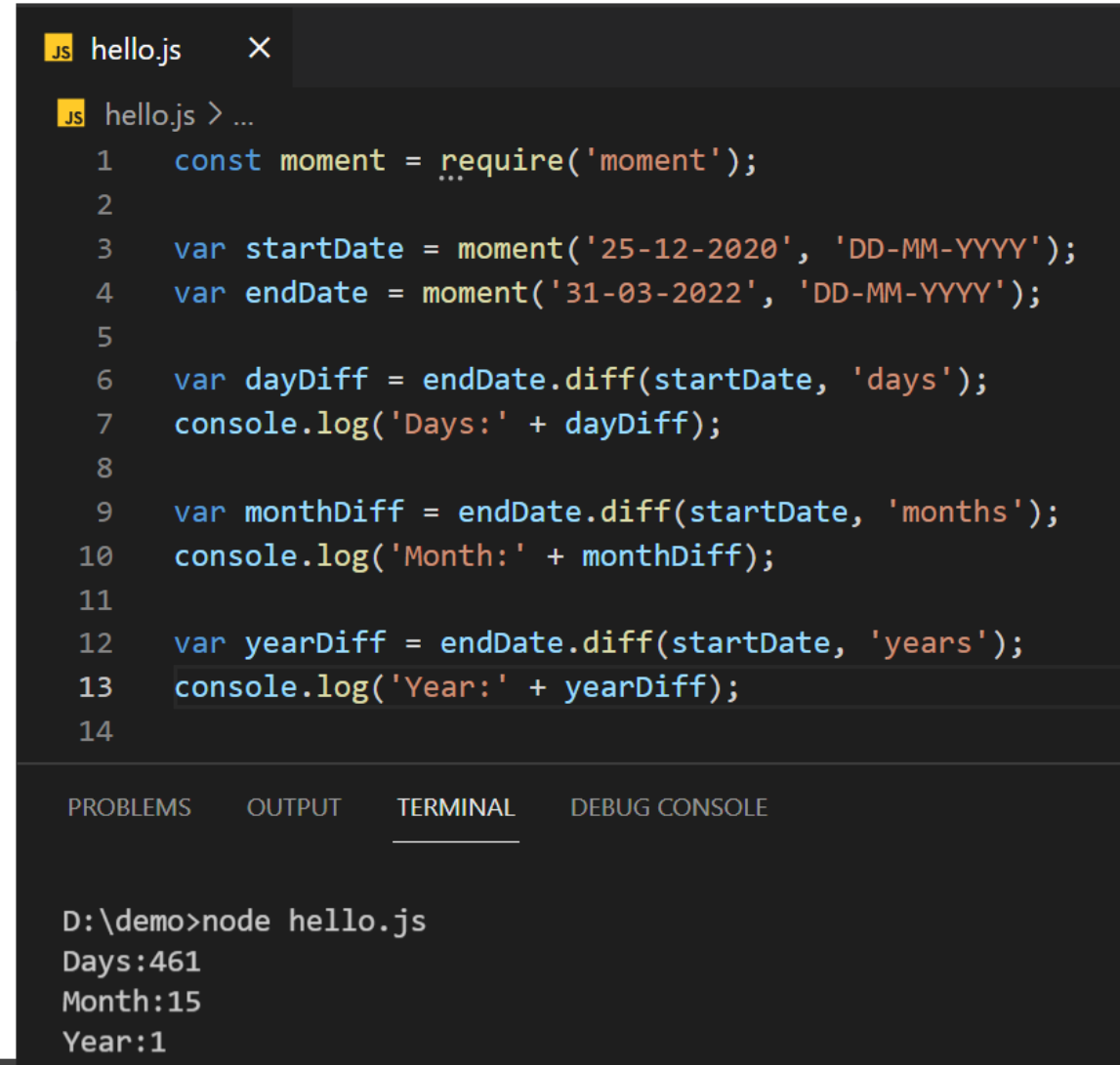
```
const moment = require('moment');

var startDate = moment('25-12-2020', 'DD-MM-YYYY');
var endDate = moment('31-03-2022', 'DD-MM-YYYY');

var dayDiff = endDate.diff(startDate, 'days');
console.log('Days:' + dayDiff);

var monthDiff = endDate.diff(startDate, 'months');
console.log('Month:' + monthDiff);

var yearDiff = endDate.diff(startDate, 'years');
console.log('Year:' + yearDiff);
```



The screenshot shows a code editor with a file named 'hello.js'. The code in the editor is as follows:

```
1  const moment = require('moment');
2
3  var startDate = moment('25-12-2020', 'DD-MM-YYYY');
4  var endDate = moment('31-03-2022', 'DD-MM-YYYY');
5
6  var dayDiff = endDate.diff(startDate, 'days');
7  console.log('Days:' + dayDiff);
8
9  var monthDiff = endDate.diff(startDate, 'months');
10 console.log('Month:' + monthDiff);
11
12 var yearDiff = endDate.diff(startDate, 'years');
13 console.log('Year:' + yearDiff);
14
```

Below the code editor, there is a terminal window with the following output:

```
D:\demo>node hello.js
Days:461
Month:15
Year:1
```

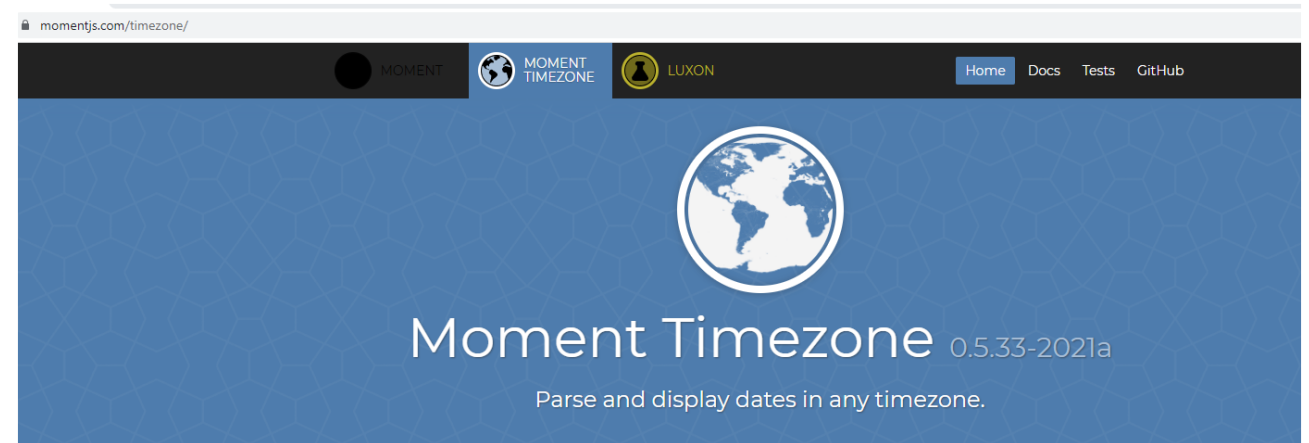


Dealing with timezones using moment.js



Timezone

- To work with timezones in moment we need moment.js core, moment-timezone.js and timezone data.

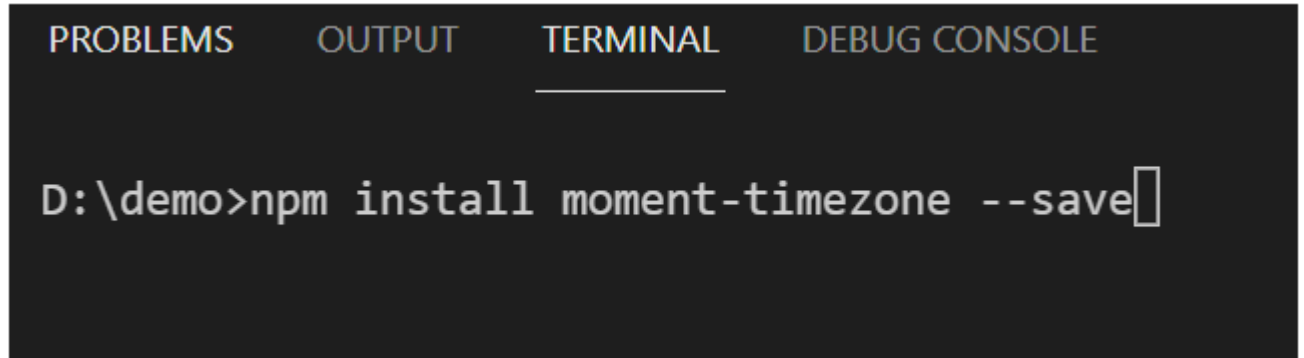


Download	Install
moment-timezone-with-data-10-year-range.js moment-timezone-with-data-10-year-range.min.js 11.5k gz	<code>bower install moment-timezone --save # bower</code> <code>npm install moment-timezone --save # npm</code> <code>yarn add moment-timezone # Yarn</code>
moment-timezone-with-data-1970-2030.js moment-timezone-with-data-1970-2030.min.js 19.8k gz	
moment-timezone-with-data.js moment-timezone-with-data.min.js 29k gz	
moment-timezone.js (no data) moment-timezone.min.js 2.9k gz	



Install Package

- `npm install moment-timezone --save`



A screenshot of a VS Code terminal window. The terminal has tabs at the top labeled 'PROBLEMS', 'OUTPUT', 'TERMINAL' (which is selected and underlined), and 'DEBUG CONSOLE'. The terminal content shows the command `D:\demo>npm install moment-timezone --save` being entered at the prompt, with a cursor at the end of the line.



Print Date Time

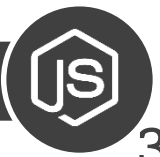
- `var moment = require('moment-timezone');`
- `var a = moment().tz("America/Los_Angeles").format();`
- `console.log(a);`

- `var a = moment().tz("Europe/London").format();`
- `console.log(a);`

```
JS hello.js X
JS hello.js > [a]
1  var moment = require('moment-timezone');
2  var a = moment().tz("America/Los_Angeles").format();
3  console.log(a);
4
5  var a = moment().tz("Europe/London").format();
6  console.log(a);

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>node hello.js
2021-12-27T19:37:51-08:00
2021-12-28T03:37:51Z
```



Get Timezone

- `var moment = require('moment-timezone');`
- `var timeZone = moment.tz.guess();`
- `console.log(timeZone);`

```
JS hello.js > ...  
1   var moment = require('moment-timezone');  
2   var timeZone = moment.tz.guess();  
3   console.log(timeZone);  
4  
5
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

```
D:\demo>node hello.js  
Asia/Calcutta
```



set custom timezone

- `var moment = require('moment-timezone');`
- `moment.tz.setDefault('America/Los_Angeles');`
- `var a = moment().format();`
- `console.log(a);`

```
JS hello.js > ...
1   var moment = require('moment-timezone');
2
3   moment.tz.setDefault('America/Los_Angeles');
4   var a = moment().format();
5   console.log(a);
6
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
2021-12-27T19:45:37-08:00
```



Different Timezone

```
var moment = require('moment-timezone');

var newYork = moment.tz("2021-06-28 12:00", "America/New_York");
var losAngeles = newYork.clone().tz("America/Los_Angeles");
var london = newYork.clone().tz("Europe/London");
var india = newYork.clone().tz("Asia/Calcutta");

console.log(newYork.format());
console.log(losAngeles.format());
console.log(london.format());
console.log(india.format());
```

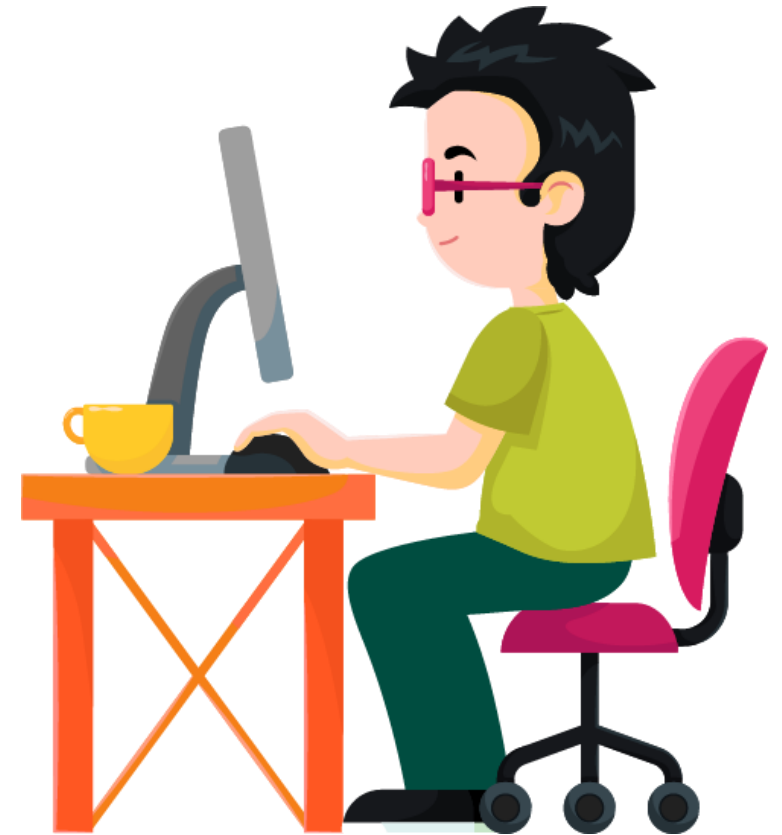
```
JS hello.js > ...
1  var moment = require('moment-timezone');
2
3  var newYork = moment.tz("2021-06-28 12:00", "America/New_York");
4  var losAngeles = newYork.clone().tz("America/Los_Angeles");
5  var london = newYork.clone().tz("Europe/London");
6  var india = newYork.clone().tz("Asia/Calcutta");
7
8  console.log(newYork.format());
9  console.log(losAngeles.format());
10 console.log(london.format());
11 console.log(india.format());
12
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node hello.js
2021-06-28T12:00:00-04:00
2021-06-28T09:00:00-07:00
2021-06-28T17:00:00+01:00
2021-06-28T21:30:00+05:30
```



Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



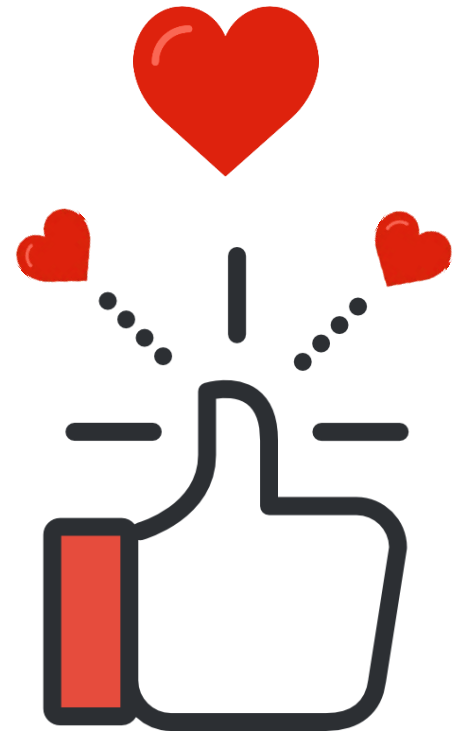
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/O79PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechlabs.com
www.akashpadhiyar.com
www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar
#apptutorials