 **answers** Trusted answers to developer questions

## How to use pug with Express.js

### Get Started With Machine Learning

Learn the fundamentals of Machine Learning with this free course. Future-proof your career by adding ML skills to your toolkit — or build foundational ML skills to land a job in AI or Data Science.

[Get Free Course](#)

**Pug** is an easy and simple template engine that could use with both NodeJs and web browsers. It allows us to write [HTML](#) using a simplified syntax that can improve code readability and boost productivity. With Pug, it's easy to create reusable HTML templates and to render data retrieved from databases or APIs dynamically.

Overall, this Pug code is similar to the HTML code but uses a more concise and streamlined syntax. The code uses indentation to indicate the nesting of elements, and it omits some of the closing tags required in HTML. The dynamic data for the title and message variables is again passed in from the server-side code using the Pug templating engine, which replaces the variables with their actual values before the HTML is sent to the client's web browser.

**Note:** Detailed explanation about pug syntax can be found [here](#).

## Implementing Pug with Express.js

First, we create a file `app.js` and folder `views`, which contain the initially empty template file `index.pug`. We also have to make a `package.json` file for our app using the following command:

```
npm init -y
```

Command to create JSON file for the project

After creating files and running a command, our project structure is as follows:

```
app.js
package.json
>views
  index.pug
```

Project's directory structure

Then, we install Express and Pug with the following command:

```
npm install express pug
```

Did you find this helpful?   



## Code explanation



As previously mentioned, we have created an empty pug file, and we'll currently insert the following code into the `index.pug` file:

```
1  html
2    head
3      title = title
4    body
5      h2 = message
```

index.pug

- **Line 3:** We set the title of the HTML document to the value of the title variable, which is passed in from the Node.js/Express application. The `=` sign is used to indicate that the content of the title tag should be dynamically generated by the server-side code.
- **Line 5:** We create an `h2` heading element with the content of the `message` variable, which is also passed in from the Node.js/Express application.

Furthermore, we are now inserting the following code into the `app.js` file, which was previously empty.

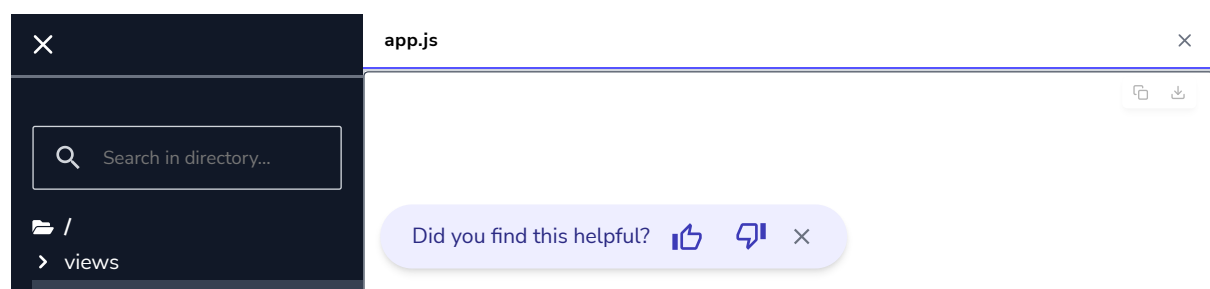
```
1  const express = require("express");
2  const app = express();
3
4  const pug = require('pug');
5
6  app.set('view engine', 'pug');
7
8  app.get('/', (req, res)=>{
9    res.render(
10      'index',
11      { title: 'PUG with Express', message: 'Hello from educative!' }
12    )
13  });
14
15  app.listen(3000);
```

app.js

- **Line 4:** We import the `pug` package, which is a templating engine for Node.js, and assigns it to a constant variable called `pug`.
- **Line 6:** We set the default `view engine` for the app to `pug` so that when the app renders views, it will use the `pug` templating engine.
- **Lines 9–11:** We render the `index` template using `pug` and passes in an object with two properties: `title` and `message`. These properties will be available to the `index` template and can be accessed using `pug` syntax.

## Executable application

Let's click the "Run" button to observe the functionality of our application.



[Courses](#)[Log In](#)[Join for free](#) →

🎉 Don't miss your Year-End Offer: Up to 20% OFF!

[Get Offer](#)

We have generated two files named `index.pug` and `app.js`. The `index.pug` file has the code that specifies the layout of the webpage, like the placement of various components on the webpage. On the other hand, the `app.js` file contains the functionalities and provides data to `index.pug` for rendering content on the webpage.

#### RELATED TAGS

[node.js](#)[pug](#)

#### CONTRIBUTOR

[Zunair Ahmad](#)

Copyright ©2023 Educative, Inc. All rights reserved



#### Related Courses

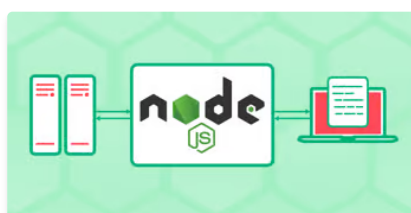
[Harsh Jain](#)

**Building Scalable  
Implementation with Node.js**


Did you find this helpful?




from Scratch

[Kappa Innovation LLC](#)



  
Intermediate

Preview →

  
Intermediate


Preview →

Keep Exploring

HTML to PUG

What is deflateRaw() in Node.js?

FUSE

  
Learn in-demand tech skills in half the time

PRODUCTS

Courses

CloudLabs New

Skill Paths

Projects

Assessments

CONTRIBUTE

Become an Author

Become an Affiliate

Earn Referral Credits

LEGAL

Privacy Policy

Cookie Policy

TRENDING TOPICS

Learn to Code

Interview Preparation

Data Science & ML

GitHub Students Scholarship

Early Access Courses

Explore Catalog

RESOURCES

Blog

Webinars

Answers

PRICING

For Individuals

Try for Free

ABOUT US




Our Team


Careers Hiring

Frequently Asked Questions

Contact Us

Press

Did you find this helpful?   



Terms of Service

Business Terms of Service

>

Data Processing Agreement

f

in

t

▶

↓

AICPA  
SOC  
2011-12

Copyright ©2023 Educative, Inc. All rights reserved.

Did you find this helpful? 

👍

👎

×

