

[WRITE A POST](#)**Naeem**[FOLLOW](#)

Full Stack Javascript Developer

# Node.js With Express And EJS

Published Feb 26, 2015

## Why EJS

The beauty of EJS is that, you can create partial views using EJS. For example you can have a common header, footer, navigation for all pages and just change the internal content using EJS.

Also you are able to pass data to views. For instance consider the username, which is different for each user, using EJS you can do something like this:

```
app.get('/', function(req, res){  
  res.render('index', {user: "John Smith"})  
});
```

And set the dynamic username each time.

As the name implies, you can use javascript for generating HTML from EJS.

Please accept our cookies! Read [Cookie Policy](#)

ACCEPT COOKIES

- Create file -> index.ejs

```
<html>
  <head><title><%= title %></title></head>
  <body>
    welcome <%= user%>;
  </body>
</html>
```

Done! you have created your first template with EJS, You found `<%= title %>` in the above code. It is just a binding and we will discuss about that later. Now what you want to do is just render the index.ejs file from your node server using EJS view engine.

## Rendering EJS

- Create a server.js file, and install the ejs module.
- Write code for your express app

```
var express = require('express');
var app = express();
```

Now, Inside you server.js set the view Engine to ejs as follows

```
app.set('view engine', 'ejs');
```

Please accept our cookies! Read [Cookie Policy](#) 

ACCEPT COOKIES

```
app.get('/', function(req, res){
    res.render('index', {user: "Great User", title: "home"});
});
```

Remember our index.ejs file, we had put tags for binding title and user in the page, here in the render function we are passing the parameters for their values

Done! Run your server.js file

## Creating Partial Views

So far we have created a single view using some dynamic data and rendered the output. Now we will try to create some Nested Views. i.e. Partial Views.

- Replace contents of your index.ejs file with this:

```
<html>
  <head>
    <% include('header.ejs') %>
  </head>
  <body>
    <% include('body.ejs') %>
  </body>
</html>
```

• Now create two more files in the views folder

Please accept our cookies! Read [Cookie Policy](#) 

ACCEPT COOKIES

```
<title>
Hello World
</title>
```

body.ejs

```
<div>
Welcome, Great User
</div>
```

- Now start your server. You have embedded partial views into your main view for index.ejs.

## Using JavaScript

As I said, you can use javascript for templating in ejs. Now we will see how it's done.

- Write this in your index.ejs file,

```
<ul>
  <% users.forEach(function(user) { %>
    <%= user.name %>
  <% }) %>
</ul>
```

Please accept our cookies! Read [Cookie Policy](#) 

ACCEPT COOKIES

```
app.get('/', function(req, res){
  res.render('index',{users : [
    { name: 'John' },
    { name: 'Mike' },
    { name: 'Samantha' }
  ]});
});
```

That's it, here we have iterated into the array, passed to the render object, and listed the names of each user using EJS.

Node.js

Ejs

Express

Enjoy this post? Give **Naeem** a like if it's helpful.

♥ 4

🗒 2

🔗 SHARE

**Naeem**

Full Stack Javascript Developer

Full Stack Javascript Developer.

Please accept our cookies! Read [Cookie Policy](#) 🍪

ACCEPT COOKIES

## 2 Replies

Leave a reply

**Pat** 2 years ago



Respectfully, either this code has errors or lacks clarity. Please fix this so that it helps others instead of confusing them.

#1 For newcomers, its very helpful to display end results

[Show more](#)

Reply

**Younis Shah\_\_** 3 years ago



```
<ul>
<% users.forEach(function(user){ %>
<%= user.name %>
<% })%>
```

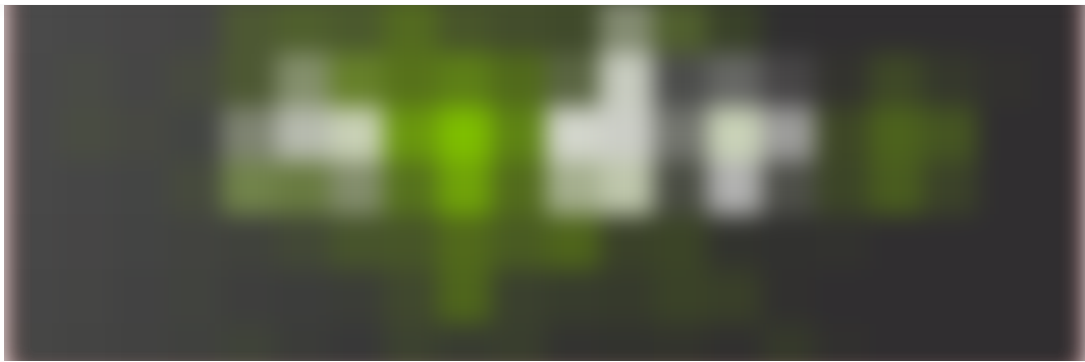
[Show more](#)

Reply

**Ioan Manuel Vasquez**

Please accept our cookies! Read [Cookie Policy](#)

ACCEPT COOKIES



CRUD is an acronym for "CREATE, READ, UPDATE, DELETE" in SQL. Basically what we are going to demonstrate is how to use MySQL in Node JS. There are lots of tutorials out there on how to make a CRUD using Mongo DB, but what we really want to show in here is how to use SQL in Node Js.

#### WHAT DO YOU NEED TO FOLLOW THIS TUTORIAL?

ES6/ES7

MYSQL

SOME BASIC KNOWLEDGE OF NODE JS

#### WHAT ARE WE GOING TO BUILD?

We are going to create a TODO LIST with Node JS using ES6 and Common JS. All we need in this project is "promise-mysql".

Let's install: `npm i -s promise-mysql`

[READ MORE](#)

Please accept our cookies! Read [Cookie Policy](#) 

ACCEPT COOKIES

# Walkthrough: Create a Simple Cucumber.js Test



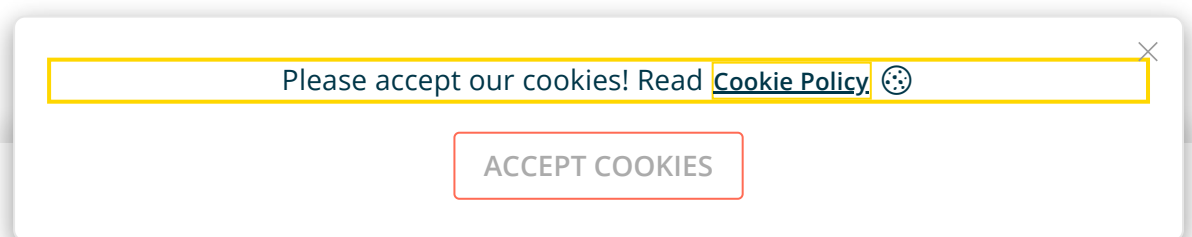
This walkthrough will create a basic behavior-driven test scripts using CukeTest:

- Create a gherkin(or feature) file and implement the code
- Run the script and get html report
- Convert the test to data driven
- Use tags to filter the run

Below are the steps.

## 1. New Project

1. Click the [New Project] button on the toolbar to bring up the "New Project" dialog box.





New Project

Project Name:

BasicTest

Project Template:

Basic

Basic Cucumber.js template

Project Path:

C:\temp\BasicTest

Browse...

Cancel

Create

[READ MORE](#)



CukeTest

## Walkthrough: Windows Notepad Test Automation with Node.js



Please accept our cookies! Read [Cookie Policy](#)

ACCEPT COOKIES

for windows TO Notepad, using Node.js and Cucumber.js. You will also learn how to create object model for your Windows application, and how to use Windows automation APIs in CukeTest to write the script.

In this walkthrough, we will:

- Create BDD scenarios that describe the automation steps
- Spy Notepad controls and add test objects to object model
- Implement the automation code
- Debug and run the script to get report

[READ MORE](#)

Please accept our cookies! Read [Cookie Policy](#) 

ACCEPT COOKIES