

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)[Javascript](#)[Node](#)

Express com Template Engine PUG

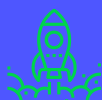
Neste artigo vamos estruturar um projeto utilizando Express com o Template Engine PUG e entender as suas principais características.

Wesley Gado · há 2 anos 5 meses

Quer receber
conteúdos
exclusivos sobre
programação?

Você sabia que a TreinaWeb é a mais completa escola para desenvolvedores do mercado?

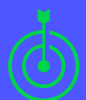
O que você encontrará aqui na TreinaWeb?



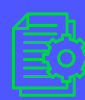
Mentoria
de carreira



**Suporte
direto** com
os professores



**Plano de
estudos** simples
e direto

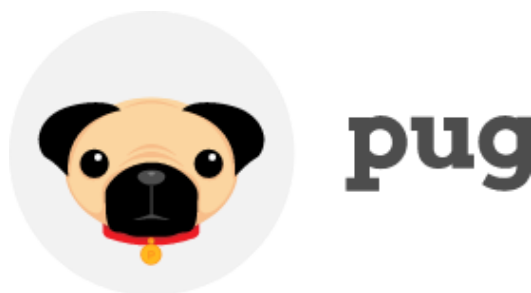


Projetos
voltados
ao **mercado de
trabalho**

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)

com elementos estáticos e elementos dinâmicos (sendo processado no servidor).

Um dos Templates Engines que é muito utilizado é o PUG, isso mesmo, o do cachorrinho :D. Veremos como instalar, um pouco sobre a sintaxe e como utilizar.



Curso

Node.js - Fundamentos[Conhecer o curso](#)

Preparando o ambiente Node.js

Antes de tudo, você deve estar com o ambiente Node.js instalado e configurado, para isto basta seguir os passos do nosso artigo [Instalação do nodejs no windows, mac e linux](#).

Logo em seguida você precisa instalar o Express, em nosso artigo [criando um servidor HTTP com express](#) explicamos o passo a passo da instalação e como configurar um servidor.

Após tudo configurado e funcionando, vamos efetivamente a instalação do pug.

Instalando o template engine Pug

Primeiramente, vamos criar um projeto novo, vou nomeá-lo de PUG. Para isto, vamos criar o diretório PUG e iniciar o nosso projeto com o comando `npm init.P`. Posteriormente, instalar o Express e configurar nosso servidor HTTP, a estrutura do projeto ficará da seguinte forma:

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)[Copiar](#)

```
import express from 'express';

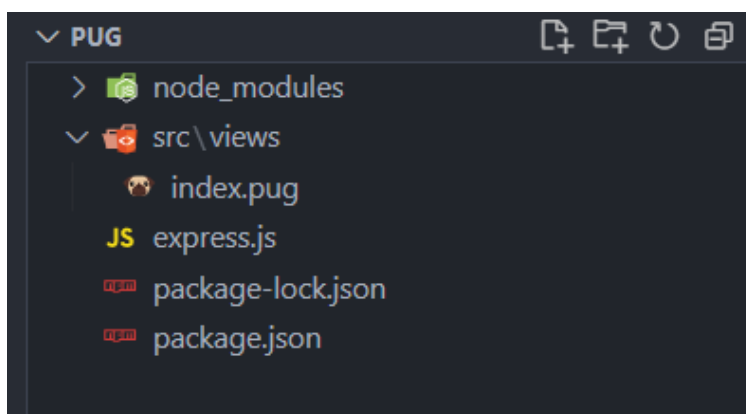
const app = express();

app.get('/', (req, res) => {
  res.send('<h1 style="color: blue">Utilizando Template Engine Pug com Express</h1>');
});

app.listen(3000, () => {
  console.log('Servidor iniciado na porta 3000');
});
```

Logo, vamos instalar o Template Engine Pug digitando o comando `npm install pug` no terminal.

Após a instalação vamos estruturar melhor o nosso projeto. Crie uma pasta *src* na raiz e dentro dela a pasta *views*, será nesta pasta que ficará os arquivos que vamos desenvolver utilizando o pug. Em seguida crie o arquivo *index.pug*.



Express

[Curso](#)**Express - Desenvolvendo aplicações web**[Conhecer o curso](#)

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)[Copiar](#)

```
import express from 'express';

const app = express();

app.set('view engine', 'pug')
app.set('views', './src/views')

app.get('/', (req, res) =>
  res.render('index')
);

app.listen(3000, () =>
  console.log('Servidor iniciado na porta 3000')
);
```

Utilizando o Template Engine PUG

Desta forma nosso arquivo index.pug será carregado quando a rota principal '/' for chamada. Porém, agora precisamos trabalhar no arquivo index.pug e colocar algum conteúdo para ser retornado, vamos iniciar com um código simples:

[Copiar](#)

```
head
  title Utilizando Template Engine PUG
body
  h1#id Este é um exemplo de como utilizar o PUG
  p.classe Exemplo de parágrafo com classe
  p(style={color: 'red'}) Exemplo de parágrafo com PUG e estilo inline
```

Perceba que o PUG possui várias características próprias, vamos listar algumas delas aqui:

- Não há tags de fechamento;
- Indentação é fundamental (justamente pela falta de tags de fechamento);
- Para utilizar um ID em um elemento, usar o '#';
- Para utilizar uma Classe em um elemento, usar o '.';

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)

Vamos criar o array no nosso arquivo express.js:

[Copiar](#)

```
import express from 'express';

const app = express();

app.set('view engine', 'pug')
app.set('views', './src/views')

app.get('/', (req, res) =>
  res.render('index', {
    carros: [                //array de carros
      "fiesta",
      "gol",
      "uno"
    ]
  })
);

app.listen(3000, () =>
  console.log('Servidor iniciado na porta 3000')
);
```

E agora exibir a lista em nosso arquivo index.pug utilizando for:

[Copiar](#)

```
head
  title Utilizando Template Engine PUG
body
  h1#id Este é um exemplo de como utilizar o PUG
  p.classe Exemplo de parágrafo com classe
  p(style={color: 'red'}) Exemplo de parágrafo com PUG e estilo inline
  ul
    for carro in carros      //usamos for para listar o array
      li= carro
```

Feito essas alterações, o resultado exibido será:

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)

Conclusão

Neste artigo aprendemos a utilizar o template engine PUG com express, ele é muito útil para criar um HTML dinâmico, pois ele é processado pelo servidor. Neste caso o template engine nos permite imprimir variáveis, utilizar condicionais, estruturas de repetição, entre outras possibilidades no HTML.

Curso

Node.js - Templates com PUG

[Conhecer o curso](#)

Por fim, caso queira aprender mais sobre Node.js saiba que aqui na [TreinaWeb](#) temos o curso [Node.js - Templates com PUG](#) que possui **01h07 de vídeos** e um total de **15 exercícios**. Conheça também nossos outros cursos de [Node.js](#).

Veja quais são os tópicos abordados durante o curso de Node.js - Templates com PUG:

- Integração com o Express;
- Modos de carregamento e envio de informações para as views;
- Modos de impressão do PUG;
- Condicionais;
- Laços de repetição;
- Uso de variáveis;
- Criação e herança de Layouts;

[#Template Engine](#)[#express](#)[#Node.js](#)

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)

desenvolvedor Front-End. Nas horas vagas grava Podcast e arrisca uns três acordes no violão.

[Todos os artigos](#)

Artigos relacionados

[Ver todos →](#)

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)

Vamos fazer um
compilado de
tópicos úteis sobre
o framework,
desde cr...

utilizar a Template
Engine EJS com
Node.js e também
alguns recursos
in...

com o Expressjs,
com exemplos
utilizando as
requisições GET,
POST...

Desenvolvimento
Back-end

NestJS com template engine handlebars

Neste artigo
vamos aprender a
configurar o
NestJS com o
handlebars, uma
template engine
que permite...

Javascript

Usando Sequelize ORM com Node e Express

Vamos aprender a
configurar uma
aplicação com
banco de dados
MySQL usando
Sequelize ORM
com Node.js...

Javascript

Criando um servidor HTTP com Express

O Express é um
framework
utilizado em
conjunto com o
Node.js. Ele possui
características que
facilit...

[Cursos](#)[Todos os cursos](#)[Formações](#)[Projetos práticos](#)[Direto ao ponto](#)[Quanto custa?](#)[Vantagens](#)[Artigos](#)[Login](#)[Matricule-se](#)

vamos aprender a utilizar o recurso de upload de arquivos locais com o NestJS utilizand...

vamos aprender como enviar emails utilizando o NestJS e o Nodemailer, um dos módulos ma...

vamos aprender os primeiros passos para trabalhar com o Prisma ORM e o NestJS.

Node

Como realizar upload no S3 com NestJS

Neste artigo vamos aprender o passo a passo de como realizar o upload de arquivos para a AWS S3 util...

Desenvolvimento
Back-end

Criando o primeiro CRUD com NestJS

Neste artigo veremos como criar um CRUD com NestJS, o passo a passo utilizando ferramentas como o Ty...

Desenvolvimento
Back-end

Validação e Serialização no NestJS: class-validator e class-transformer

Vamos abordar a utilização de duas bibliotecas de serialização e validação fundamentais no NestJS: cl...



Escola online para desenvolvedores



Inscreva-se e receba nossos lançamentos, promoções e novidades

[Inscreva-se](#)[Cursos](#)[A empresa](#)[Contato](#)[Artigos](#)[Baixe nosso aplicativo](#)



- Cursos 
- Todos os cursos
- Formações
- Projetos práticos
- Direto ao ponto
- Quanto custa?
- Vantagens
- Artigos

Login

Matricule-se

 Av. Paulista, 1765, Conj 71 e 72 - Bela Vista - São Paulo - SP - 01311-200

© 2004 - 2023 TreinaWeb Tecnologia LTDA - CNPJ: 06.156.637/0001-58