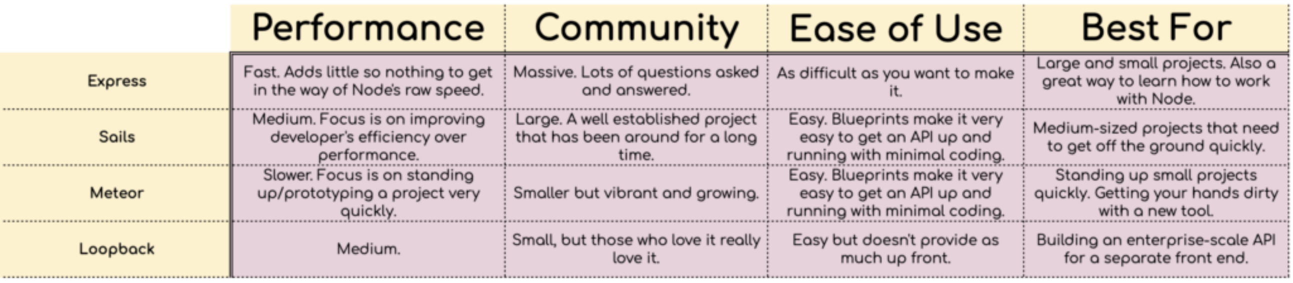


	Performance	Community	Ease of Use	Best For
Express	Fast. Adds little so nothing to get in the way of Node's raw speed.	Massive. Lots of questions asked and answered.	As difficult as you want to make it.	Large and small projects. Also a great way to learn how to work with Node.
Sails	Medium. Focus is on improving developer's efficiency over performance.	Large. A well established project that has been around for a long time.	Easy. Blueprints make it very easy to get an API up and running with minimal coding.	Medium-sized projects that need to get off the ground quickly.
Meteor	Slower. Focus is on standing up/prototyping a project very quickly.	Smaller but vibrant and growing.	Easy. Blueprints make it very easy to get an API up and running with minimal coding.	Standing up small projects quickly. Getting your hands dirty with a new tool.
Loopback	Medium.	Small, but those who love it really love it.	Easy but doesn't provide as much up front.	Building an enterprise-scale API for a separate front end.

The Best NodeJS Frameworks for 2021

Last Updated on April 20, 2021 by Lizzie Shipton Leave a Comment

Choosing a framework for your NodeJS application can be tough. There are so many of them, with their own strengths and weaknesses. At the end of the day, the framework you choose will depend on your programming environment, and what you want your app to do. Some frameworks are lightweight, fast and don't care about how you build your app. Others have more features but may compel you to use their engines and organize your code in a certain way.

The goal of this article is to compare some of the most popular Node frameworks so that you can ultimately decide which one is right for your project.

View the Best Free APIs List

What is a Framework?

Generally speaking, a framework is a support system for your application. Node provides a set of tools to interface with things your app needs, like the network and the filesystem. A framework extends these capabilities and abstracts away some of the more complicated aspects of development. For example, a handler in your code might receive a request object from the network. A framework could intercept this request object before it reaches your handler, adding some additional methods and capabilities to make it easier to work with.

There are several flavors of the Node framework, and each provides a different programming experience.

Browse APIs

MVC Frameworks

MVC (Model View Controller) architecture is a useful design pattern that splits application logic into three parts — **models** to define the shape of the data, **views** to organize the user interface, and **controllers** to communicate between the two. Node frameworks that support MVC architecture are useful if you don't want to spend too much time worrying about how to organize your code.

Express

We may as well get Express out of the way first, as no list of Node frameworks would be complete without it. Express is still the reigning champion of popular frameworks, as its

47.5K Github stars, 1.8K watches, and 7.7K forks will attest. This alone makes it an excellent choice. It is robust, well-tested, and there is a large community maintaining and working with it who can support you and answer questions.

Express touts itself as a "fast, un-opinionated, minimalist framework for Node." The focus is on performance and providing only what you need. The framework provides little functionality of its own, instead of allowing for extensive middleware chaining to process requests.

Getting Started

Create a server.js or app.js file in the root of your project and add the following code

```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

Run the server with

```
node ./server.js
```

and visit localhost:3000 in your browser to see the response.

Sails

Sails emphasize stability and ease of use and provides a lot more out of the box than Express. It leans in the full stack direction, providing support for authentication, its own ORM for database interfacing, WebSockets and templating. Sails is more opinionated than Express — meaning you must use the provided ORM (Waterline) and templating engine (EJS, although an extension called Consolidate provides support for other templating engines that are compatible with Express.) Sails also boasts an impressive API generator called Blueprints, which allows you to generate API endpoints with minimal manual coding. From the Sails documentation:

> *…if you create a User.js model file in your project, then with blueprints enabled you will be able to immediately visit /user/create?name=joe to create a user, and visit /user to see an array of your app's users. All without writing a single line of code!*

Sails has been around for a long time and is among the most popular frameworks according to its Github stars, views and forks. Many large companies rely on Sails, meaning it's robust enough to support enterprise-level applications.

Getting Started

Sails provides a powerful command-line interface to get up and running with a new app very quickly

```
npm install -g sails
sails new <project_name>
```

You'll be prompted to choose either a web app template with built-in authentication and login or an empty app. Once Sails finishes setting up the basics, cd into your project folder and run

```
sails lift
```

You can now navigate to localhost:1337 to see the generated homepage.

Full-Stack Frameworks

Full-stack frameworks provide structure and functionality for your entire application – from the client to the server to the database. A full-stack framework might come with many features, including templating engines, WebSocket libraries, and ORMs. Depending on the size of your team and application, these features could be very useful or they could be more than you need.

Meteor

Meteor is the second-most starred, viewed and forked Node framework on Github. The community is vibrant and the documentation is extensive. Documentation includes best practices, recommended style guides and many tutorials and technical articles. Meteor comes bundled with npm, and its own package manager called Atmosphere, as well as built-in support for a Mongo datastore and easy integration with React, Angular or Blaze. It is more opinionated than Express or Sails.

Meteor uses “data on the wire” as opposed to server-side rendering — so the server sends data, not HTML, and the client renders it. The Meteor build tool provides out of the box support for mobile through Cordova and supports hot reloading.

Getting Started

Meteor’s build tool is what creates, compiles and runs your app. Download it via curl.

```
curl https://install.meteor.com/
meteor create <project_name>
```

This will create the file structure for everything you need in a basic Javascript app, organized into */client* and */server* directories. cd into your project root and run

```
meteor run
```

The app should now be running on localhost:3000

Browse APIs

REST API Frameworks

If you have the client-side of your application covered, you might just need a framework for the server part of your stack. In this case, you might go with a simple REST API framework just to handle CRUD requests to your server. You could pretty much do this with Express, but there are also frameworks specifically geared toward handling this particular case.

Loopback

Loopback is the second most popular REST API framework, according to Github. Developed by IBM, it is a "highly extensible, open-source Node framework based on Express that allows you to quickly create APIs and microservices." It comes with a command-line tool for generating projects and creating controllers and models with ease and provides add-on support for easy authentication and authorization. It doesn't provide any support for views or templating or a dedicated ORM, as it is intended to be used only as an API.

Loopback allows you to create a dynamic API in minutes with minimal coding. The development cycle is very quick, and the file structure is clean and useful. Setup includes options to configure eslint, prettier, mocha and docker right out of the box.

Getting Started

```
npm install -g @loopback/cli

lb4 <project_name>
```

You'll be prompted to answer several questions about how you want your app set up, and the file structure will be scaffolded. cd into the project root and run

```
npm start
```

To create a new route, run

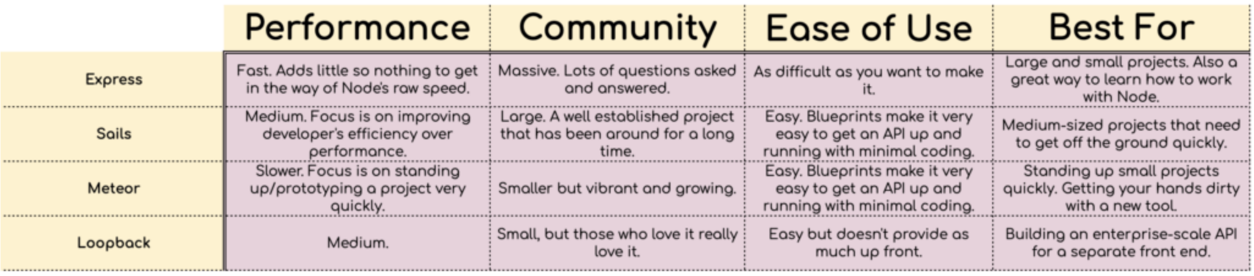
```
lb4 controller
```

and answer the prompts. Once the controller file is created, you can import the Loopback get function and use it to setup the controller

```
import {get} from '@loopback/rest';

export class HelloController {
  @get('/hello')
  hello(): string {
    return 'Hello world!';
  }
}
```

Summary: Best Node.js Frameworks



	Performance	Community	Ease of Use	Best For
Express	Fast. Adds little so nothing to get in the way of Node's raw speed.	Massive. Lots of questions asked and answered.	As difficult as you want to make it.	Large and small projects. Also a great way to learn how to work with Node.
Sails	Medium. Focus is on improving developer's efficiency over performance.	Large. A well established project that has been around for a long time.	Easy. Blueprints make it very easy to get an API up and running with minimal coding.	Medium-sized projects that need to get off the ground quickly.
Meteor	Slower. Focus is on standing up/prototyping a project very quickly.	Smaller but vibrant and growing.	Easy. Blueprints make it very easy to get an API up and running with minimal coding.	Standing up small projects quickly. Getting your hands dirty with a new tool.
Loopback	Medium.	Small, but those who love it really love it.	Easy but doesn't provide as much up front.	Building an enterprise-scale API for a separate front end.

Framework	Performance	Community	Ease of Use	Best For
Express	Fast. Adds little so nothing to get int he way of Node's raw speed.	Massive. Lots of questions asked and answered.	As difficult as you want to make it.	Large and small projects. Also a great way to learn how to work with Node.
Sails	Medium. Focus is on improving developer's efficiency over performance.	Large. A well established project that has been around for a long time.	Easy. Blueprints make it very easy to get an API up and running with minimal coding.	Medium-sized projects that need to get off the ground quickly.
Meteor	Slower. Focus is on standing up/prototyping a project very quickly.	Smaller but vibrant and growing.	Easy. Blueprints make it very easy to get an API up and running with minimal coding.	Standing up small projects quickly. Getting your hands dirty with a new tool.
Loopback	Medium.	Small, but those who love it really love it.	Easy but doesn't provide as much up front.	Building an enterprise-scale API for a separate front end.

Browse APIs

Conclusions

It’s difficult to choose a definitive “best” Node framework, as the framework you choose will change based on your needs. If you want the freedom to choose your own libraries and third-party integrations, Express or Loopback might be your best bet. But if your team wants to stand up a brand new app front to back in a matter of a few days, Meteor or Sails might better suit your needs.

There are many other frameworks beyond those discussed in this article, and the best way to know what will work for your app or team is to read through the docs and try a few out.

View the Best Free APIs List

Related Links

- How to use an API
    - with JavaScript
- How to create an API
    - with Node.js and Express.js
- How to Build a Weather App with Node.js
- How to Set Up a GraphQL Server with Node and Express

2.7/5 - (3 votes)

« How to Create a Weather App with Dark Sky API and JavaScript (Node.js)

Introducing Our Student Program: RapidAPI School [Who is Eligible, How to Apply] »

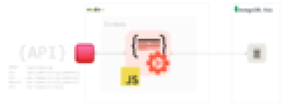
Related Blog Posts

How to Use the REST Countries API with Python, PHP, Ruby & Javascript Examples

Top Golang Web Frameworks to Choose In 2021

REST API Documentation Templates, Tools, and Examples

REST API Best Practices for REST API Developers



How to build a REST API with Node.js (MongoDB & Express) – Part 2

What is API Design? API Design Best Practices and Principals for APIs

Filed Under: Case Study Tagged With: express.js, framework, full stack, loopback, meteor, mvc, node.js, rest api, sails

Lizzie Shipton

Lizzie is a Full Stack Engineer at Udacity and freelance technical content writer. She has experience working with Node, GraphQL, Postgres, React and Sass. She writes about things that matter: code, sailing and dogs.

LinkedIn | Blog

Search this website

Search

Leave a Reply

Comment

Name *

Email *

Post Comment

Build amazing apps, faster.

Discover, evaluate, and integrate with any API. RapidAPI is the world’s largest API Hub with over 2,000,000 developers and 20,000 APIs.

Browse APIs

APIs mentioned in this article

Browse Most Popular APIs

Learn

How to use an API

API Glossary

API Testing

API Management

For Developers

For API Providers

Most Popular APIs

Free APIs List

About

Team

Jobs

Contact Us

Write for Us

API Directory

Press Room

Privacy Policy

Terms of Use