Изучайте Deno и выходите за рамки Node.js

08.05.2023

Хотя Node.js остается наиболее используемой серверной средой выполнения JavaScript с огромным отрывом, альтернативные среды выполнения, такие как Deno и Bun, привлекли к себе внимание, поскольку они пытаются улучшить концепцию Node.js. Deno, более популярная из двух новых сред исполнения, решает некоторые проблемы безопасности, присущие Node.js, и обеспечивает более полную поддержку таких технологий, как TypeScript и WebAssembly. В этой статье вы изучите основы Deno, узнаете, как он сопоставим с Node.js и Bun, а также проследите за практической демонстрацией, в которой Deno используется для создания простого HTTP веб-сервера.

Что такое Deno?

Когда разработчики познакомились с JavaScript, они увидели его потенциал для программирования на локальных машинах. Поэтому они создали серверные среды исполнения — среды, позволяющие выполнять код JavaScript на машинах без использования браузера. Райан Дал разработал для этой цели Node.js, а позже создал Deno, чтобы решить некоторые проблемы, с которыми он столкнулся в оригинальном дизайне Node.js. Некоторые заметные недостатки включают зависимость от централизованного менеджера пакетов, такого как npm, отсутствие стандартной библиотеки и слабые настройки безопасности по умолчанию.

К основным преимуществам Deno относятся следующие:

- **Безопасность по умолчанию** — пользователь должен явно дать разрешение коду на доступ к сети, файловой системе или окружению.
- **Встроенная поддержка TypeScript и WebAssembly** —

Выполнять программы на TypeScript и WebAssembly в Deno так же просто, как и программы на JavaScript. Среда выполнения компилирует эти языки так же, как и JavaScript.

- **Децентрализованный менеджер пакетов** – Вместо того чтобы полагаться на репозиторий пакетов, например, npm или менеджер пакетов Bun, Deno может импортировать код непосредственно из URL. Эта возможность означает, что вы можете загружать зависимости из любого места, где они размещены, включая ваш репозиторий GitHub, сервер или CDN. Deno также предлагает услугу хостинга скриптов для еще более удобного доступа.
- **Соответствие веб-стандартам** – Deno стремится следовать тем же API, что и браузеры, что означает, что код, написанный для браузеров, легко переносится в среду исполнения.

Крупные компании и основные игроки отрасли, такие как Slack, Netlify и Supabase, приняли Deno, но его принятие среди веб-разработчиков было менее распространено. По данным опроса Stack Overflow 2022 года, всего 1,47% профессиональных разработчиков, принявших участие в опросе, используют Deno, в то время как 46,31% сообщили об использовании Node.js.

Что делает Deno?

Как и любая другая среда выполнения JavaScript, Deno позволяет разработчикам выполнять JavaScript на стороне сервера. В результате вы можете использовать Deno для выполнения широкого спектра задач программирования. Больше всего Deno подходит для таких задач, как разработка серверных приложений, которые отвечают на запросы пользователей в Интернете. Например, если вы создаете книжный интернет-магазин, вы можете использовать Deno для создания приложения, которое получает информацию из базы данных PostgreSQL, создает страницу, которую хочет просмотреть пользователь, и отправляет ее в браузер для

визуализации. Вы также можете использовать Deno для программирования более низкого уровня, например, для создания инструмента командной строки для управления делами через терминал. Другими словами, вы можете использовать Deno для достижения тех же целей, что и с помощью таких языков, как Python или Ruby.

Deno против Node

Deno стремится стать улучшением Node.js, и он выполняет это обещание в нескольких ключевых областях. Deno улучшает безопасность, позволяя более тонкие настройки доступа для различных модулей кода. Он также фокусируется на соответствии API веб-стандартам, что позволяет разработчикам использовать один и тот же код как в браузере, так и на сервере. Для команд, работающих над проектами JavaScript на стороне сервера, Deno стал жизнеспособной альтернативой Node. И хотя их схожая функциональность убедила некоторых разработчиков в том, что Deno может заменить Node.js, такая возможность маловероятна по нескольким ключевым причинам.

Node.js — это самая популярная среда выполнения JavaScript, и она собрала обширную экосистему предварительно написанных пакетов и большое, активное сообщество пользователей. Эти бесценные ресурсы помогают Node.js оставаться чрезвычайно привлекательной средой выполнения. В отличие от него, Deno является новым: версия 1.0 была выпущена в мае 2020 года, поэтому сравнительно немногие разработчики успели поиграть с ним. Освоение нового инструмента удлиняет сроки разработки. Кроме того, неясно, принесет ли Deno значительные преимущества многим простым проектам. Но если вы создаете приложение в области, где безопасность очень важна, например, в финансовой сфере, то возможности Deno по обеспечению безопасности могут сделать переход на новую систему оправданным.

Deno против Bun

Бывший инженер Stripe Джарред Самнер впервые выпустил Bun в июле 2022 года для бета-тестирования. Bun — это более экспериментальная среда выполнения, чем Deno, и, в отличие от Deno, она разработана с учетом широкой обратной совместимости с Node.js.

Bun также может похвастаться молниеносной производительностью, превосходящей Node.js и Deno. Ключевые особенности позволяют реализовать эти возможности:

- **Лучший движок** — Вместо движка V8 JavaScript и Web Assembly от Google, Bun использует более быстрый и эффективный JavaScriptCore в качестве базового движка JavaScript.
- **Больше контроля над кодом** — Bun написан на Zig, низкоуровневом языке, который обеспечивает больший контроль над выполнением кода, чем JavaScript.
- **Тонкая настройка эффективности** — Команда, работающая над Bun, уделяла приоритетное внимание профилированию, бенчмаркингу и оптимизации во время разработки для обеспечения эффективности кода.

Bun настолько новый, что поддержка сообщества относительно невелика, чтобы помочь в устранении неполадок. Тем не менее, Bun может быть интересен для экспериментов. Команды, которым специально требуется повышение производительности, могут найти Bun полезным для своих проектов, но в веб-разработке часто приоритетны другие факторы, а не производительность.

Начало работы с Deno

Теперь, когда вы немного узнали о Deno и его сравнении с другими популярными JavaScript runtimes, пришло время посмотреть, как он работает. В этом разделе вы узнаете, как

создать простой сервер в Deno, который отвечает на HTTP-запросы словами “Привет от сервера!”.

Установка Deno

Вы можете установить Deno на машину как двоичный исполняемый файл, используя эти инструкции по установке из официальной документации. На macOS, например, вы можете установить Deno с помощью команды **brew install deno**.

Другой способ начать работу с Deno – установить его как пакет npm, например, следующим образом:

Создайте папку для вашего проекта (возможно, deno_example) и выполните в ней команду **npm init**. (Вы можете принять все опции по умолчанию, предложенные init, поскольку он создает базовый файл package.json).

После инициализации приложения выполните команду **npm install deno-bin** для установки бинарного пакета Deno. Теперь вы можете обновить файл package.json, чтобы включить запуск приложения с помощью **npm start**. Добавьте жирную строку ниже к свойству объекта “scripts” в стандартном файле package.json:

```
"scripts": {
    "start": "deno run --allow-net app.ts",
    "test": "echo "Error: no test specified" && exit 1"
},
```

Это дополнение к сценарию позволяет Deno запускать модуль app.ts с сетевыми привилегиями (**–allow-net**). Помните, что при работе с Deno вам необходимо явно разрешить доступ к сети или файловой системе. Теперь вы готовы к созданию модуля app.ts, отвечающего за прослушивание порта и обслуживание запросов пользователей.

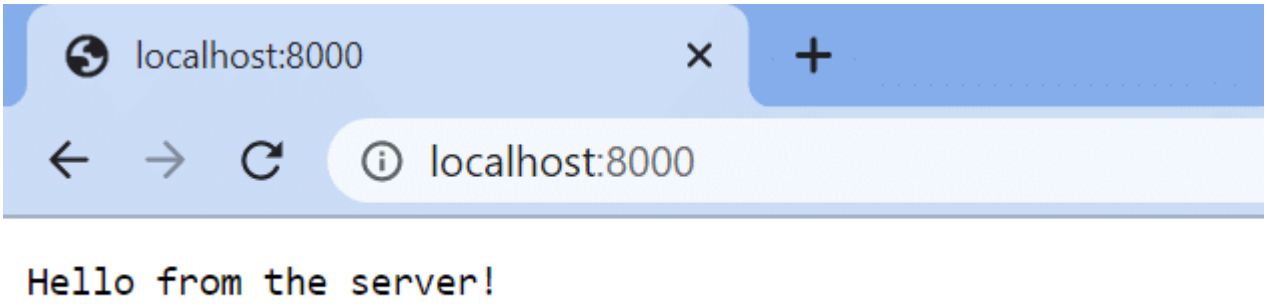
Создание модуля App.ts

Создать базовый сервер в Deno очень просто. Сначала создайте файл app.ts и вставьте в него следующий код:

```
import { serve } from "https://deno.land/std@0.177.0/http/server.ts";

serve((_req) => new Response("Hello from the server!"), { port: 8000 });
```

Код использует функцию serve из библиотеки server.ts Deno, хранящейся на официальном сайте Deno.land. Этот код также предоставляет функцию-обработчик serve для входящих запросов. Функция-обработчик отвечает на каждый запрос словами “Привет от сервера!”. Функция serve также принимает необязательные параметры, такие как номер порта, на котором вы хотите обслуживать запрос. В примере кода эти параметры используются для обслуживания на порту 8000. Далее запустите сервер, выполнив команду **npm start**. Это запустит сервер, который прослушивает localhost:8000 и отвечает на запросы приветствием.



Если вы хотите расширить сервер до полноценного API, вам, вероятно, потребуется добавить возможность подключения к базам данных. Это легко сделать, поскольку сообщество Deno создало драйверы, поддерживающие такие популярные базы данных, как MariaDB/MySQL, PostgreSQL, MongoDB и другие.

Заключение

Времена выполнения варьируются от обычных и надежных до очень экспериментальных. Выбор подходящего для вашего проекта зависит от вашего проекта и от того, как вы хотите, чтобы среда выполнения помогла вам достичь ваших целей. Node.js хорошо подходит для большинства проектов. У него большая

экосистема и многочисленное сообщество, которое может помочь в решении широкого спектра проблем. Преимущество Deno заключается в дополнительной безопасности и лучшем опыте разработчика. В то же время он лучше всего подходит для опытных команд, чтобы его преимущества перевесили временные и трудовые затраты на изучение незнакомой среды выполнения. Наконец, хотя Bun слишком экспериментален для большинства профессиональных проектов, это уникальная и интересная среда исполнения, которую можно взять для личного проекта или для расширения. В целом, Deno предлагает баланс между преимуществами Node.js и экспериментальными возможностями Bun. В то время как Node.js является подходящим выбором для большинства проектов, Deno может оказаться на переднем крае того, как веб-разработка будет развиваться в будущем.