



12 Популярных JavaScript-фреймворков для построения API и микросервисов

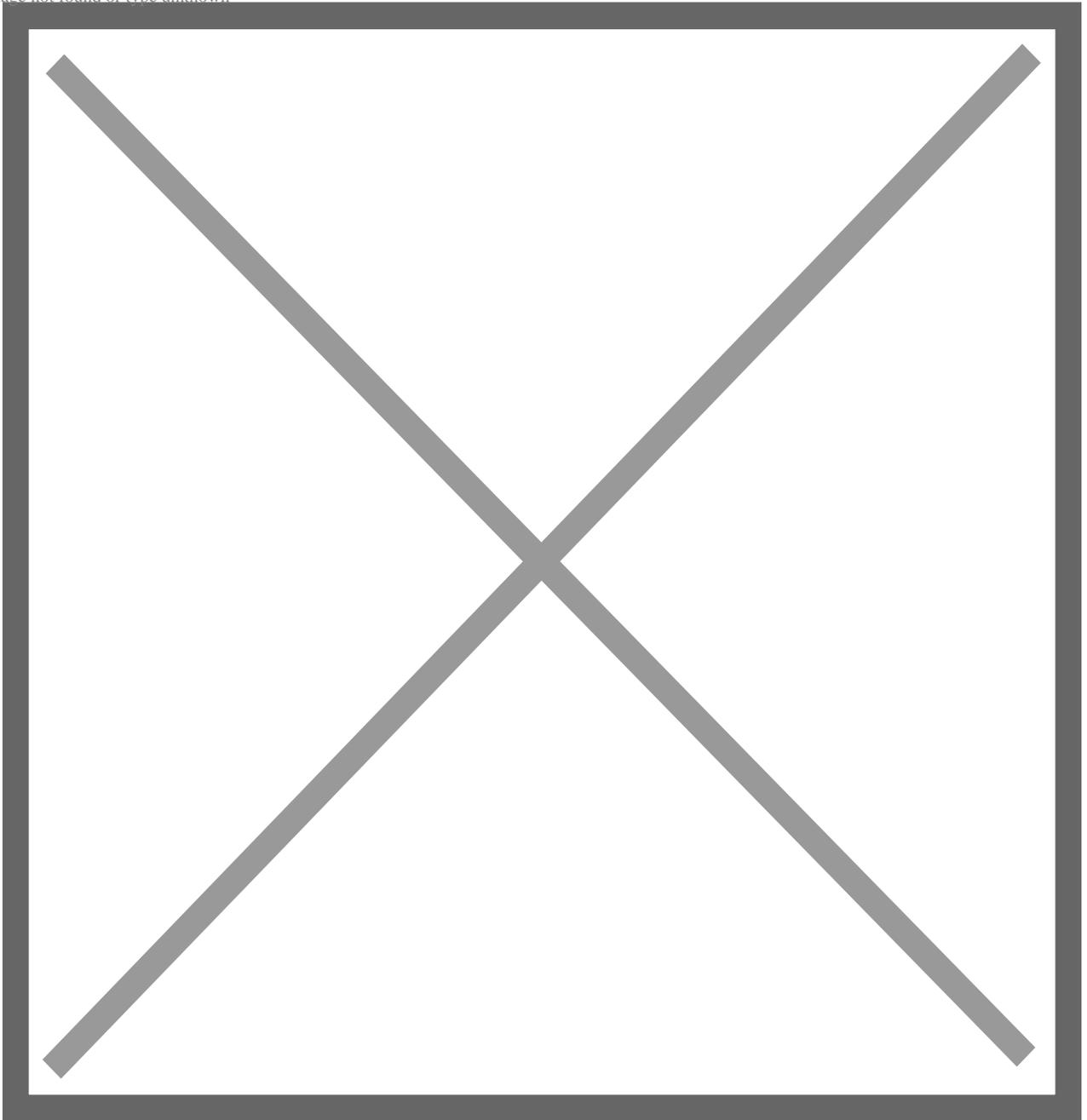
Описание

Создавайте современные приложения, управляемые API! API похожи на пользовательские интерфейсы, с той лишь разницей, что пользователь – это машина или программное обеспечение. Давайте разберемся, что такое API-first design approach и почему имеет смысл ему следовать. Мы также уделим немного внимания JavaScript и его популярности, а затем, наконец, обсудим лучшие трендовые JavaScript-фреймворки для построения API и микросервисов.

API-первый подход

Все мы слышали термин API first design или просто API. На самом деле, термин API first design стал популярным после того, как в индустрии начали упоминать термин cloud-native application. С течением времени от разработки монолитного приложения программисты начали разбивать эти приложения на микросервисы.

Image not found or type unknown



Все взаимодействия с этими микросервисами из Web, Android или iOS устройств должны происходить через API. Поэтому проектирование этих API играет важную роль в предоставлении данных для этих интерфейсов. Это означает, что у микросервисов есть только один способ взаимодействия – API. Таким образом, на первый план выходит принцип проектирования API. Мы должны сначала продумать дизайн API, прежде чем писать реализацию для наших микросервисов/бэкенд-сервисов. Помните: API являются драйверами бизнеса, поэтому разрабатывайте их в самом начале разработки системы. При подходе API-first design API рассматривается

как гражданин первого класса. Суммируя приведенные выше утверждения, мы получим три ключевых момента, которые можно принять за принцип подхода API first.

- API – это первый пользовательский интерфейс нашего приложения.
- API идет первым, а затем приложение.
- API описывается (и, возможно, даже самоописывается).

Существует множество компаний, использующих подход API first, таких как Twilio, NETFLIX, Stripe, Algolia, SendGrid, Etsy, Adobe, Amazon и многие другие. Amazon была одной из первых компаний, которая проинструктировала своих сотрудников об использовании этого подхода. Важность этого подхода можно понять на примере случая, когда Джефф Безос (генеральный директор Amazon) издал мандат под названием Bezos Mandate (примерно в 2002 году) с предупреждением, что любой, кто не будет следовать этому процессу, будет уволен.

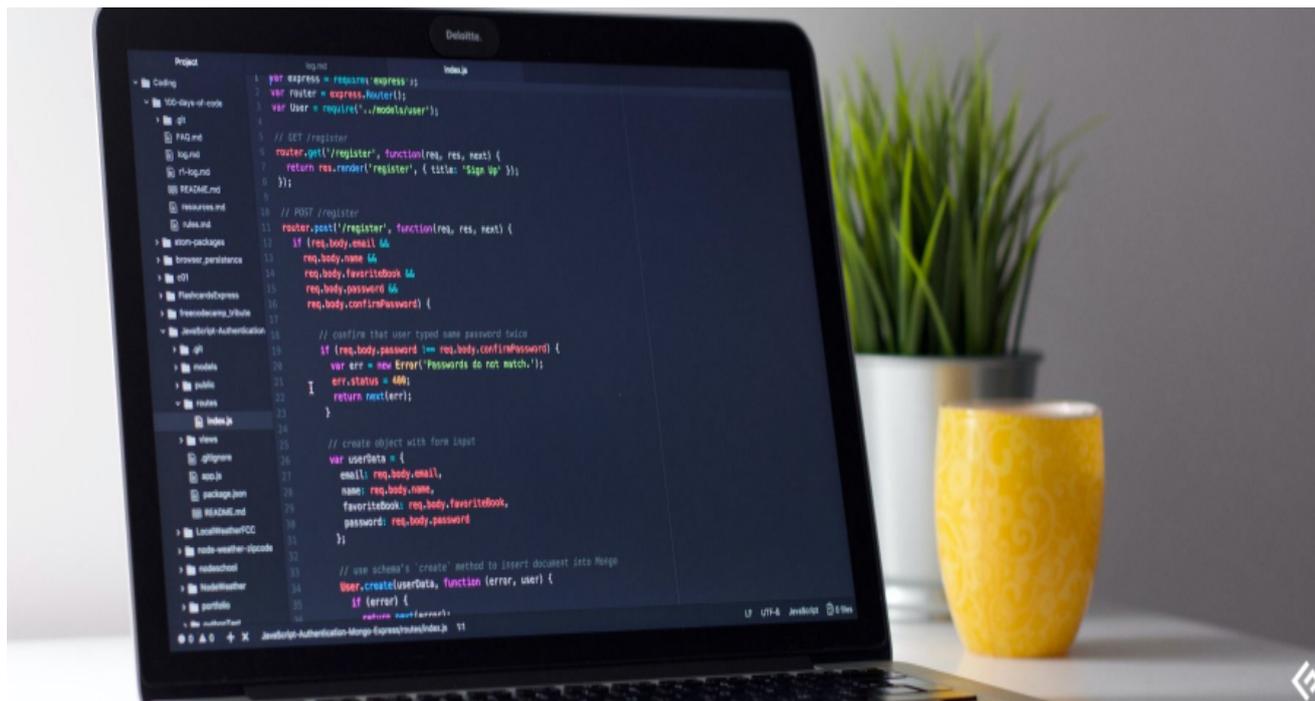
Преимущества подхода API-first

Множество! Ниже перечислены некоторые из них:

- Многоразовое/составное предприятие
- Позволяет вести параллельную разработку
- Снижает общие затраты на протяжении жизни проекта.
- Снижает риск неудач
- Более быстрый рост
- Свобода от ограничений

Теперь давайте рассмотрим один из мощных языков Интернета, который поможет нам реализовать вышеописанный подход.

JavaScript: Язык веба



JavaScript – один из самых важных языков сегодняшнего и завтрашнего дня.

JavaScript – это высокоуровневый интерпретируемый язык сценариев на стороне клиента. Он используется для того, чтобы сделать веб-страницы более интерактивными. Он позволяет нам реализовывать сложные и красивые дизайны. JavaScript – это объектно-ориентированный язык сценариев, который поддерживает инкапсуляцию, полиморфизм и, в некоторой степени, наследование. Кроме того, он не требует компиляции, в отличие от C, C++ и Java. JavaScript может выполняться в браузере, на сервере и на любом устройстве с движком javascript.

Хорошо бы упомянуть, что JavaScript не имеет ничего общего с Java. Тогда возникает вопрос, почему он называется JavaScript? На самом деле, когда JavaScript был первоначально выпущен, он назывался Mocha; позже он был переименован в live script, а затем в JavaScript, когда Netscape и SUN заключили лицензионное соглашение. JavaScript используется при разработке практически всех типов веб-продуктов и приложений, таких как смарт-часы, игры, веб-сайты, мобильные и веб-приложения. Некоторые из самых популярных веб-сайтов, такие как Google, Facebook, Netflix, Amazon, используют JavaScript для создания своих продуктов.

Ограничивается ли он клиентской стороной?

Изначально JavaScript был разработан для работы в браузере как язык

программирования на стороне клиента. Но сейчас он сильно эволюционировал и стал чем-то большим, чем был раньше. Более того, JavaScript – это не только язык программирования на стороне клиента, но и фреймворки на основе JavaScript используются также для программирования на стороне сервера. Примером такого фреймворка является Nodejs.

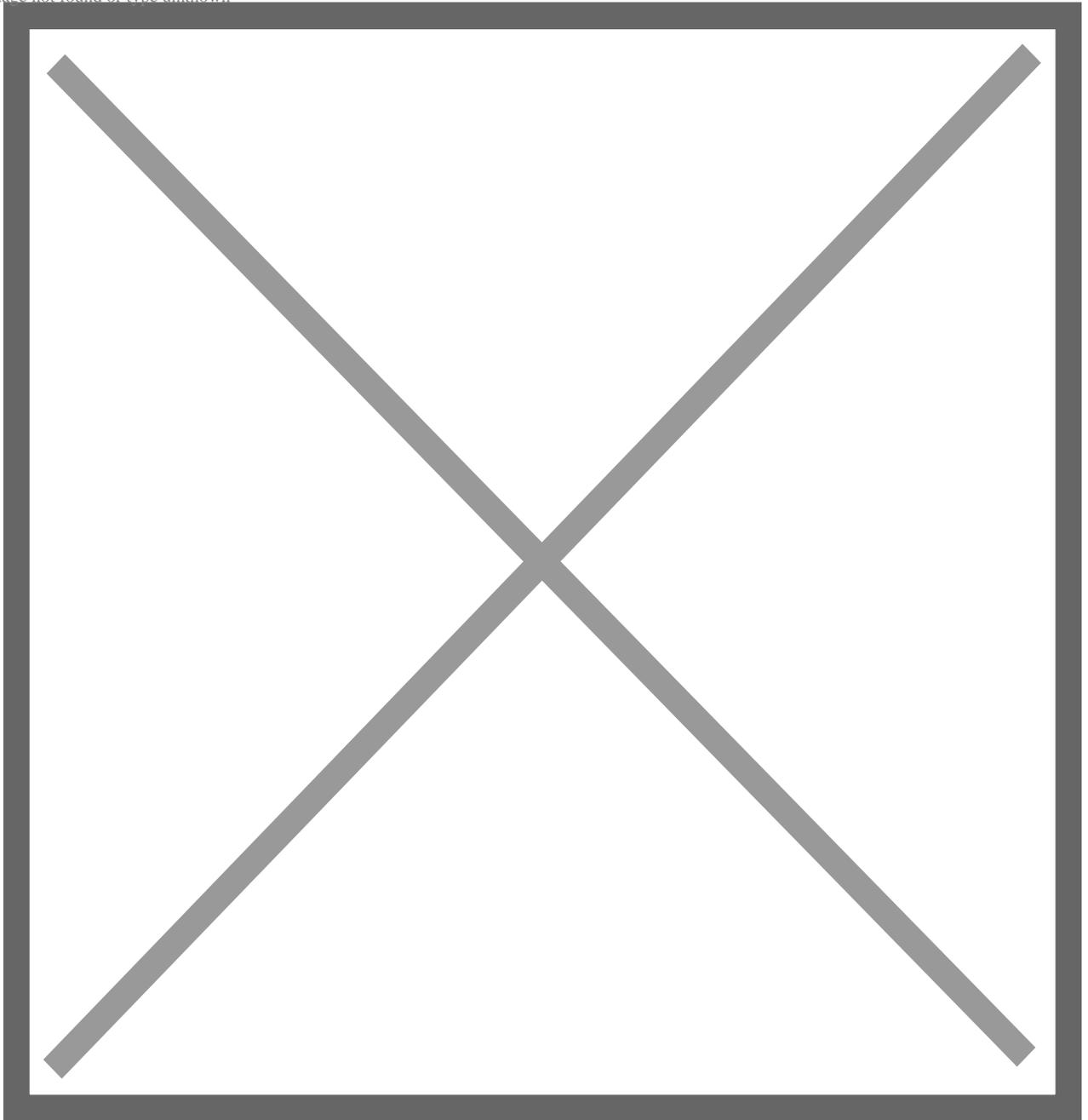


На этом возможности JavaScript не заканчиваются. Помимо использования в качестве клиентской и серверной стороны, он также может быть использован для разработки приложения с единой кодовой базой и запуска на мобильных (Android, iOS) и веб. Примером такого фреймворка является Reactjs.

Популярность JavaScript

JavaScript, пожалуй, один из самых важных языков на сегодняшний день. Развитие Интернета привело JavaScript туда, куда он никогда не задумывался. В настоящее время существует более 100 языков программирования, и каждый год появляются новые языки. Среди них есть несколько мощных языков, которые быстро меняют рынок, и JavaScript, безусловно, один из них. Он всегда был в списке популярных языков программирования. По данным StackOverflow, JavaScript был самым популярным и часто используемым языком программирования.

Image not found or type unknown

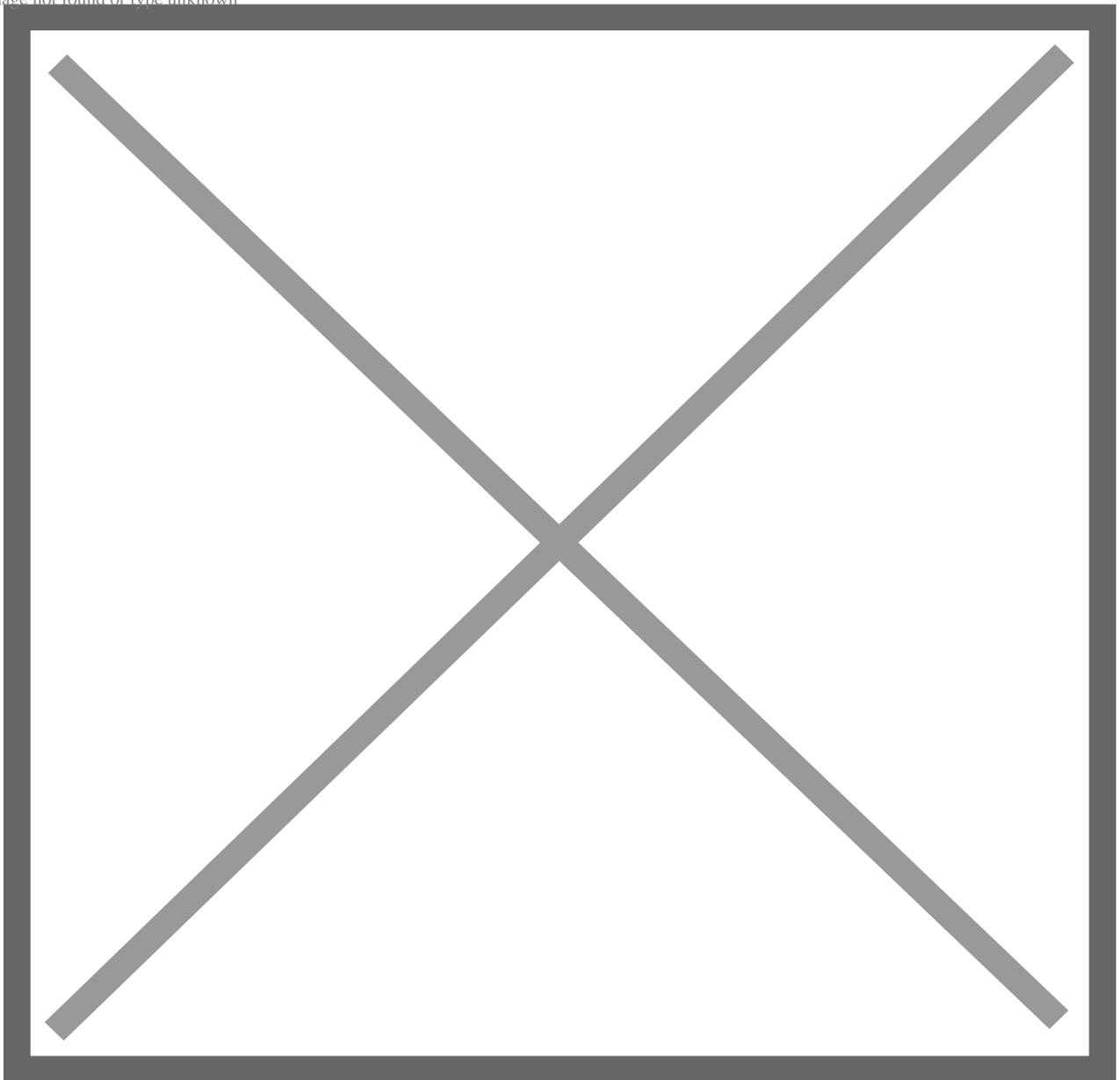


Знаете ли вы? Почти 70% профессиональных разработчиков, ответивших на опрос Stack Overflow в 2020 году, кодируют на JavaScript. Теперь давайте рассмотрим некоторые из самых популярных JavaScript-фреймворков для построения API и микросервисов.

12 Популярных JavaScript-фреймворков для построения API и микросервисов

Нарі

Image not found or type unknown

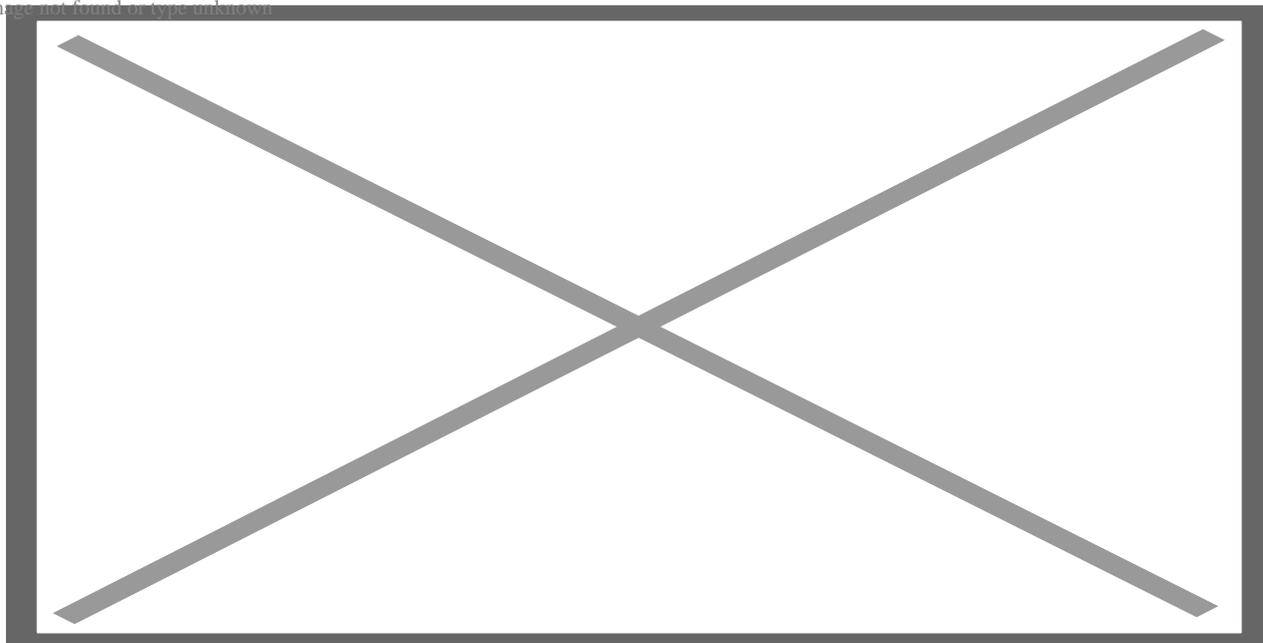


Нарі – это простой, безопасный фреймворк javascript с открытым исходным кодом. Он используется для создания мощных, масштабируемых приложений с минимальными накладными расходами. Изначально он был разработан для работы в масштабах “черной пятницы” Walmart, но сейчас Нарі продолжает оставаться

проверенным выбором для бэкенда корпоративного уровня. Такие компании, как Beats Music, Vrbo, Conde Nast, Auth0, Brave и многие другие используют Narі и довольны им. Бесплатная поддержка всегда доступна на GitHub. Однако для более быстрого реагирования присоединяйтесь к каналу Slack – именно там многие члены сообщества общаются и помогают друг другу.

Restify

Image not found or type unknown



Restify оптимизирован для создания семантически корректных RESTful веб-сервисов, готовых к использованию в производственных масштабах. Когда говорится о семантически корректных, это означает, что они остаются верными спецификации, что было одной из главных целей команды проекта Restify. Он оптимизирован для интроспекции и производительности и используется в большинстве крупнейших развертываний Node.js. Одной из самых интересных особенностей Restify является то, что он автоматически создает для вас зонды DTrace всякий раз, когда вы добавляете новый маршрут/хандлер. На самом деле в Restify поставляется три отдельных клиента:

- `JsonClient`: отправляет и ожидает `application/json`
- `StringClient`: отправляет URL-кодированный запрос и ожидает `text/plain`
- `HttpClient`: тонкая обертка над HTTP/HTTPS библиотеками `node`.

Restify используется некоторыми из самых уважаемых компаний в отрасли, такими как Netflix, Joyent, Pinterest, для обеспечения работы некоторых крупнейших

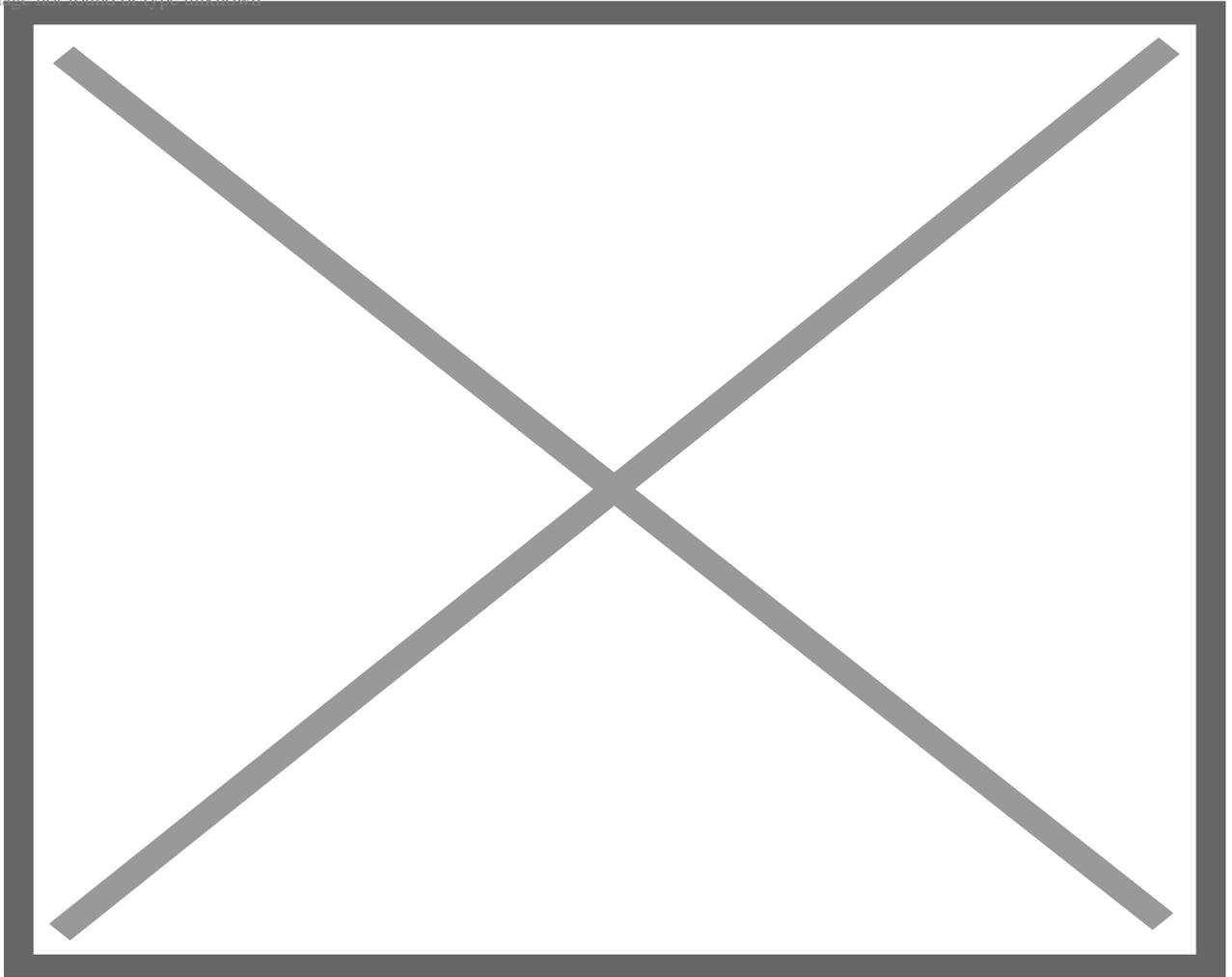
развертываний.

ExpressJS

ExpressJs – это проект OpenJS Foundation. Благодаря его надежности, многие популярные фреймворки (такие как Feathers, Kraken, Sails, Kites, NestJs и т.д.) основаны на Express. Это минимальный и гибкий фреймворк веб-приложений на базе node.js, который предоставляет надежный набор функций для веб- и мобильных приложений. Поскольку он предоставляет слой поверх node.js, мы можем эффективно обращаться к функциям JavaScript. Он подходит для написания приложений с серверным рендерингом, поскольку поддерживает множество шаблонизаторов (таких как pug и handlebars). Для разработки API он содержит промежуточные обработчики и множество методов HTTP-сервисов/утилит.

SailsJS

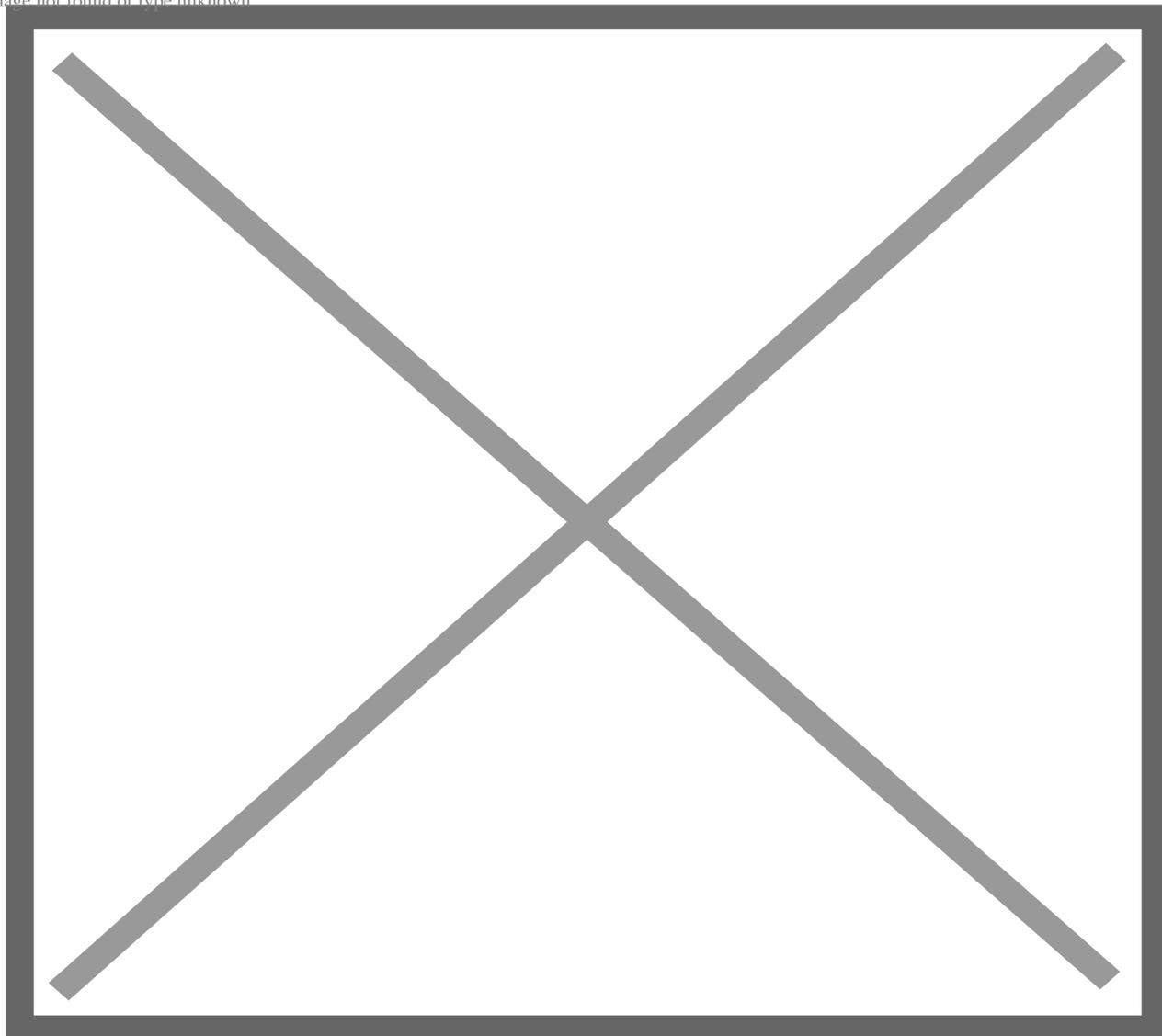
Image not found or type unknown



SailsJS – это MVC-фреймворк, созданный компанией Sails со штаб-квартирой в Остине, штат Техас. Он был разработан Майком МакНилом для помощи его команде в создании масштабируемых проектов на Node.js для стартапов и корпоративных клиентов. С момента своего выпуска в 2012 году Sails стал одним из наиболее широко используемых фреймворков для веб-приложений во всем мире. Это легкая платформа, которая располагается поверх Express. Он совместим с любым фронтендом: Angular, React, iOS, Android. Sails претендует на интеграцию с любой базой данных благодаря мощному ORM под названием Waterline. Он использует Express для обработки HTTP-запросов и обертывает socket.io для управления WebSockets.

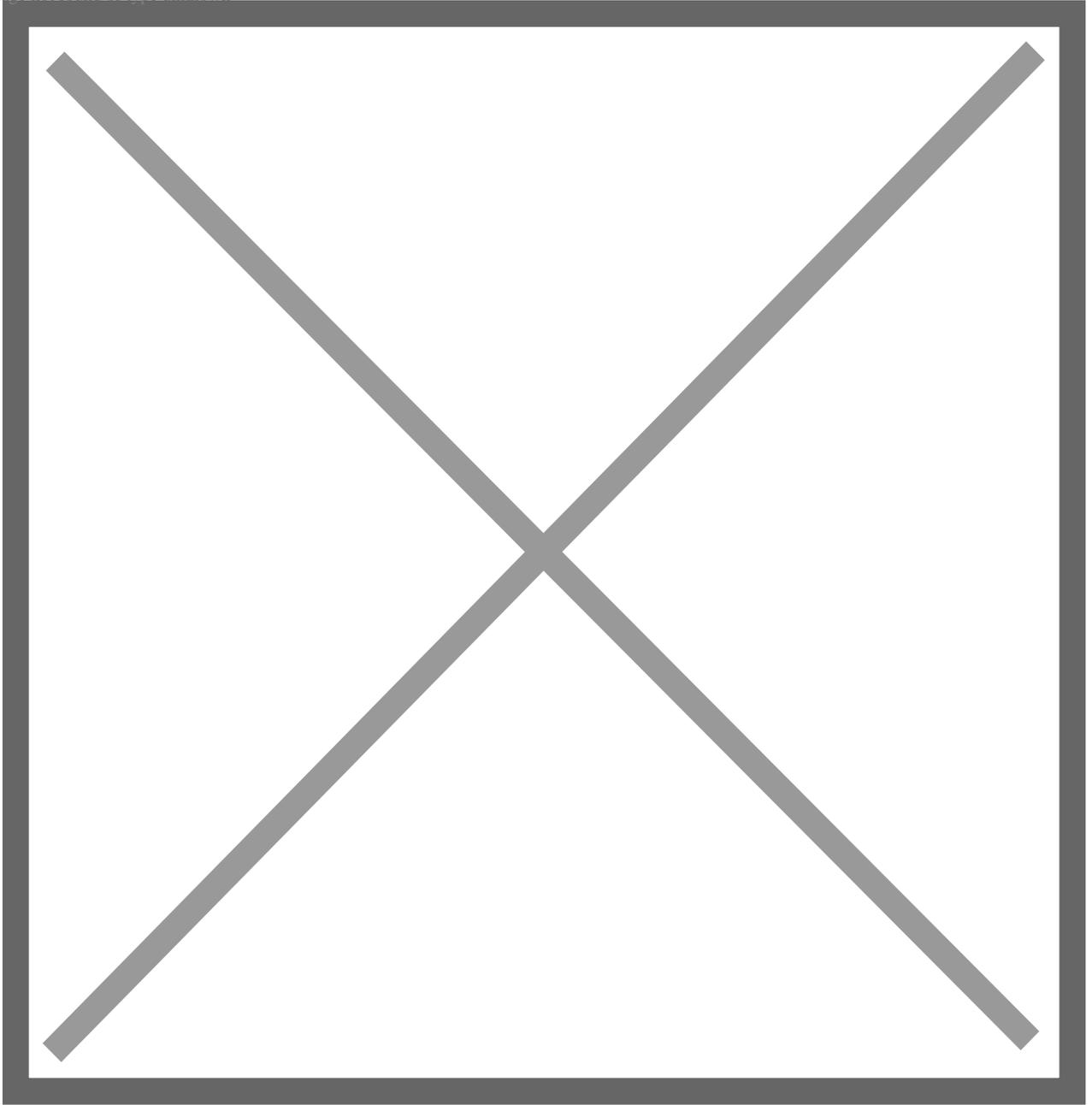
Action Hero

Image not found or type unknown



Actionhero – это мультитранспортный API-сервер с интегрированными возможностями кластера и отложенными задачами. Цель Actionhero – создать простой в использовании набор инструментов для создания многократно используемых и масштабируемых API для HTTP, WebSockets и многого другого. Он был создан с нуля и включает в себя все функции, которые мы ожидаем от современного API-фреймворка. Тип рабочей нагрузки, с которой Actionhero справляется лучше всего, включает создание и потребление API, хранение и извлечение данных из баз данных, изменение файлов и другие подобные задачи. Actionhero имеет 5 ключевых концепций, из которых состоит каждое приложение: Действия, Задачи, Инициализаторы, Чат и Серверы.

Image not found or type unknown



LoopBack

Image not found or type unknown



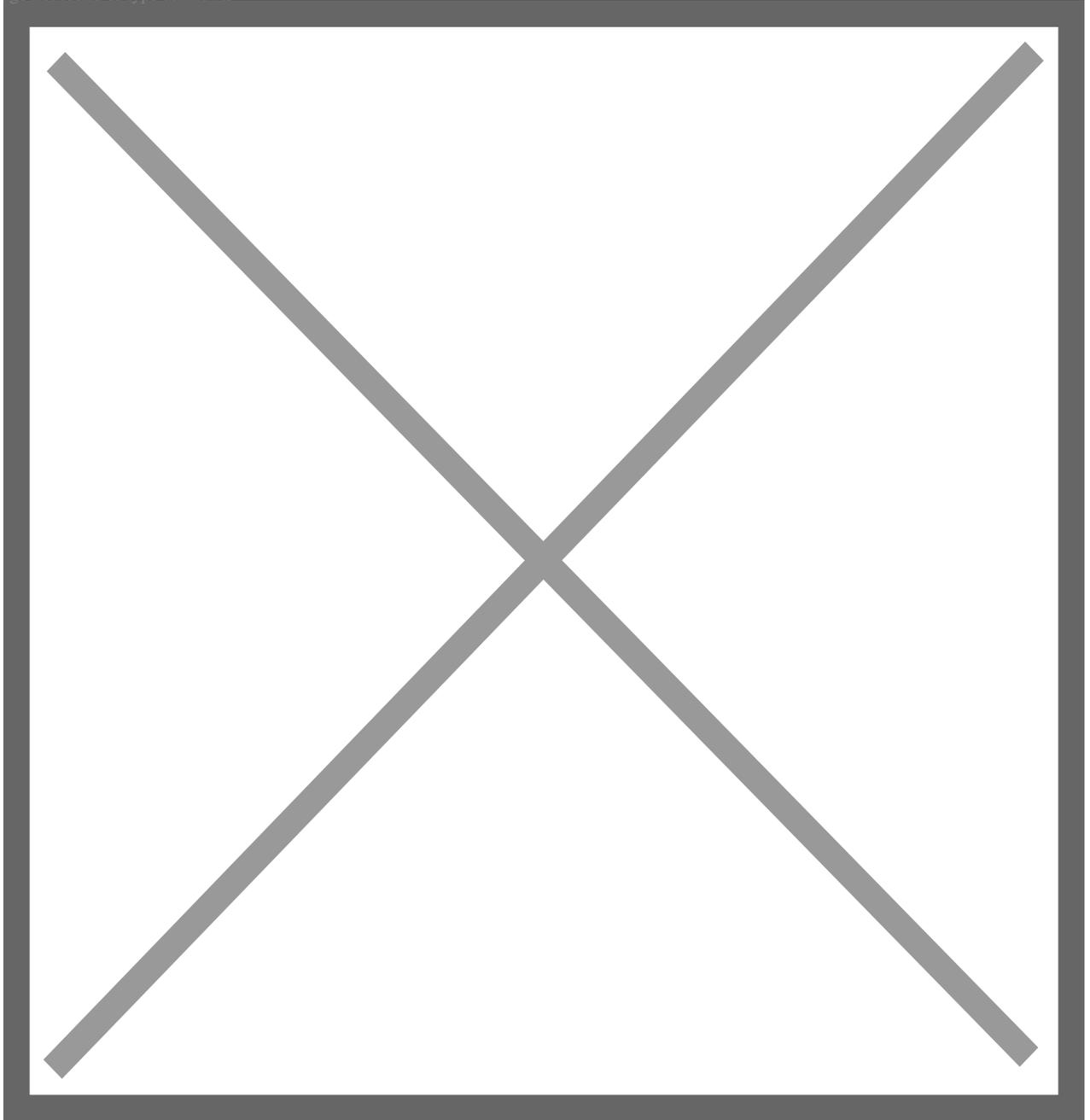
LoopBack – это легко расширяемый фреймворк с открытым исходным кодом для Node.js и TypeScript, основанный на Express. Он позволяет быстро создавать API и микросервисы, состоящие из бэкенд-систем, таких как базы данных и сервисы SOAP или REST. Это новая, улучшенная модель программирования с Dependency Injection и новыми концепциями, такими как Components, Mixins, Repositories и т.д., которые делают его наиболее расширяемым. LoopBack позволяет нам создать интерфейс GraphQL для любого REST API с помощью OpenAPI-to-GraphQL. Мы можем начать работу с Loopback в кратчайшие сроки, просто следуя шагам в руководстве по началу работы. Их документация классифицируется по четырем квадрантам, а именно:

- **Ориентированная на обучение:** Учебники предоставляют практические уроки, где пользователи могут изучить фреймворк на практике.
- **Ориентированная на решение проблем:** How-to Guides предоставляют рецепты для решения конкретных задач, с которыми вы можете столкнуться при создании проекта LoopBack.
- **Ориентированные на понимание:** Страницы концепций объясняют концепции архитектуры, более широкий взгляд и более глубокие знания о фреймворке.
- **Информационно-ориентированные:** Справочные руководства предоставляют техническое описание механизма и способы его использования.

Такие компании, как GoDaddy, Symantec, Sapient, Fundrails, ShoppinPal и WhiteBoard широко используют LoopBack.

FeathersJs

Image not found or type unknown



Feathers – это набор инструментов и архитектурный паттерн, позволяющие легко создавать масштабируемые REST API и приложения реального времени. Мы можем создавать прототипы за несколько минут, а готовые к производству приложения –

за несколько дней. Feathers может взаимодействовать с любой технологией back-end, поддерживать более десятка баз данных и работать с любой технологией front-end, такой как React, VueJS, Angular, React Native, Android или iOS. Чтобы начать работу над первым приложением Feathers, следуйте краткому руководству по запуску. Если вы застряли или у вас возник вопрос, вы можете получить помощь на их различных каналах, включая Slack, GitHub, StackOverflow, Telegram и т.д.

KoaJS

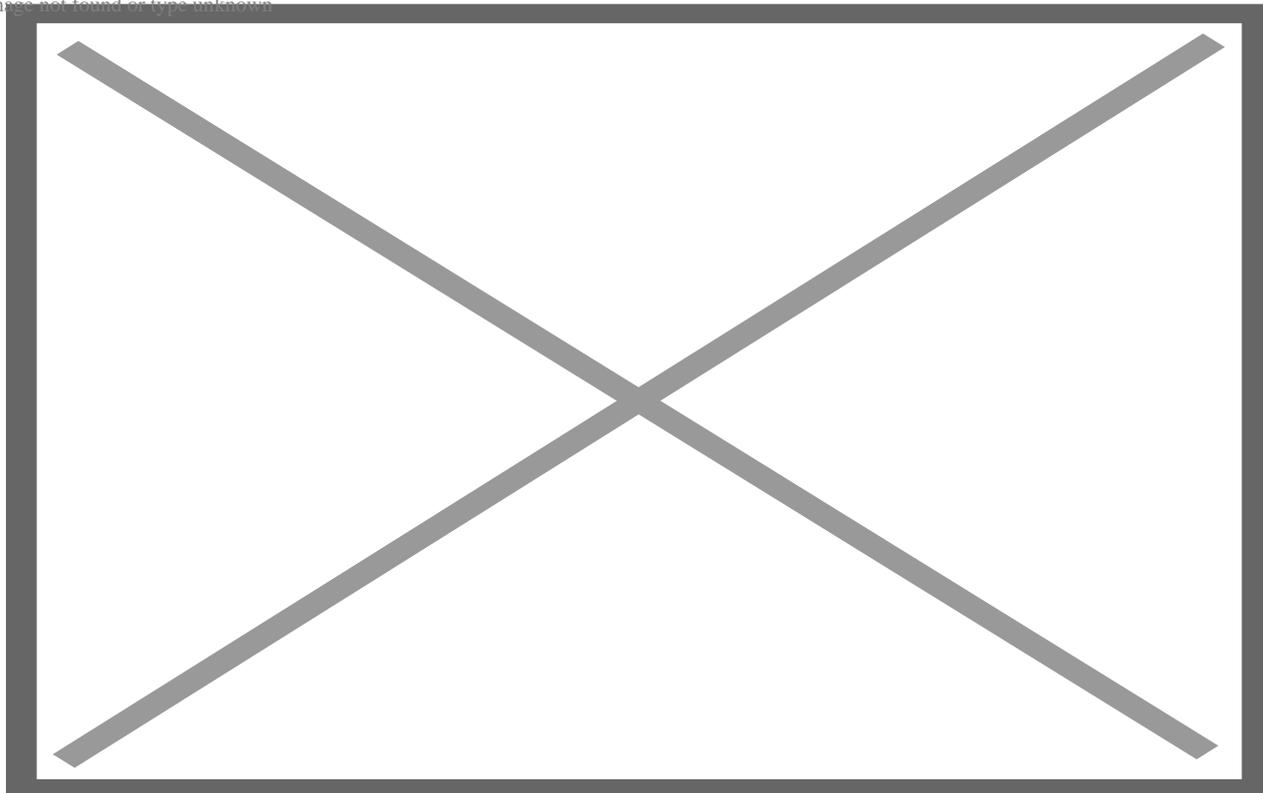
Команда, стоящая за Express, разрабатывает koa. Его цель – стать меньшей, более выразительной и надежной основой для веб-приложений и API. Koa не включает в свое ядро никакого промежуточного программного обеспечения и предоставляет элегантный набор методов, которые делают написание серверов быстрым и приятным. Используя асинхронные функции, Koa позволяет отказаться от обратных вызовов и значительно повысить эффективность обработки ошибок. Запустите его с помощью трех команд, как показано ниже.

```
$ nvm install 7
$ npm i koa

$ node my-koa-app.js
```

AdonisJS

Image not found or type unknown

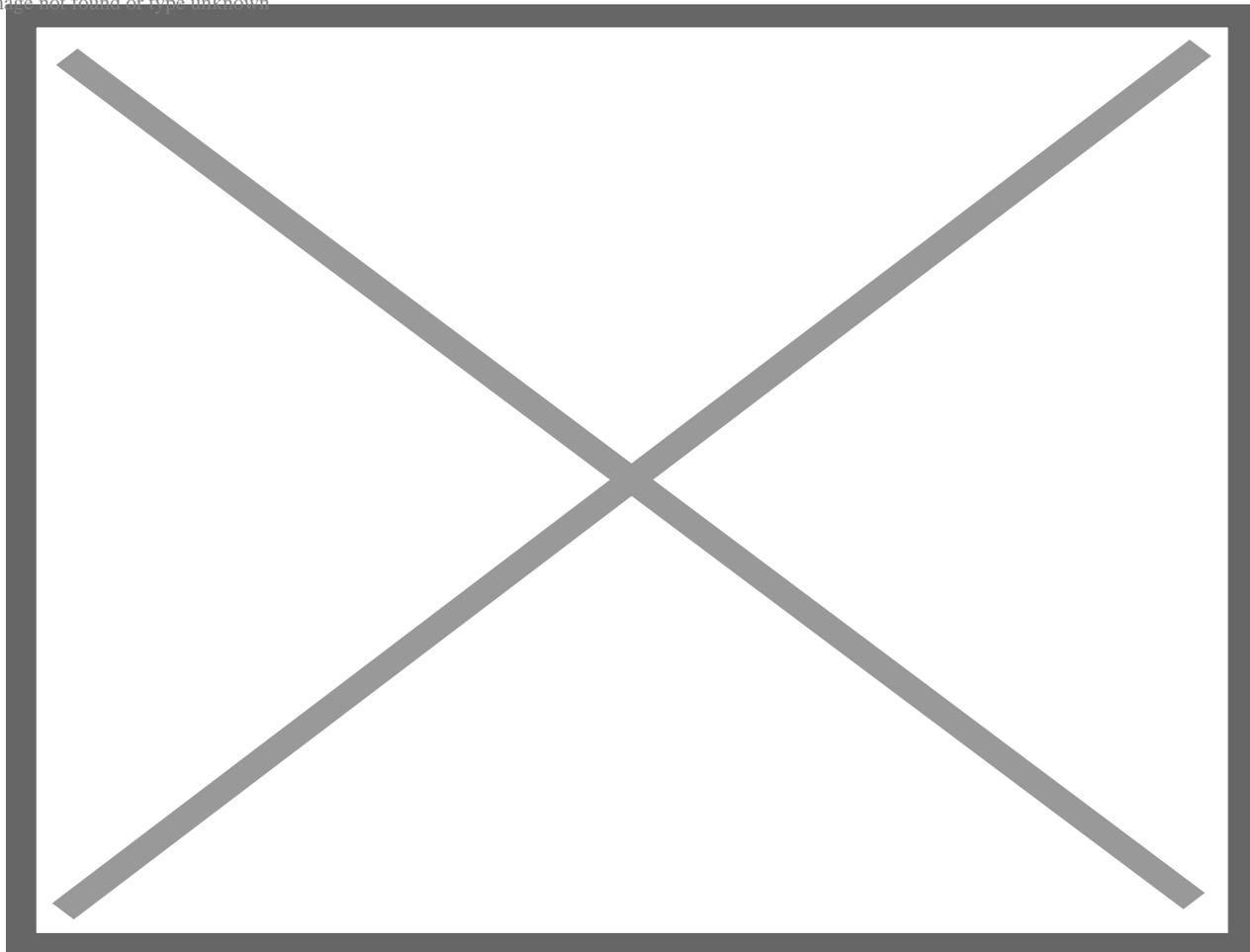


AdonisJs – это фреймворк Node.js MVC, который поддерживается и поддерживается активным сообществом. Он предлагает стабильную экосистему для написания приложений на стороне сервера, чтобы вы могли сосредоточиться на потребностях бизнеса, а не на том, какой пакет выбрать. Он похож на любой другой модуль npm с некоторым кодом сверху для бесперебойной работы с приложениями AdonisJs. Для использования AdonisJs нет жестких предварительных требований, но обычное понимание JavaScript, Async программирования и Node.js будет очень полезным. Обязательно прочитайте руководство по установке. В случае возникновения каких-либо вопросов или ошибок, вы можете связаться с командой AdonisJs по указанным ниже каналам.

- Github: Делитесь ошибками или создавайте запросы на функции.
- Discord: Присоединяйтесь к серверу Discord, чтобы мгновенно общаться с другими членами сообщества.
- Twitter: Оставайтесь в курсе ежедневного прогресса и получайте информацию о потрясающих проектах сообщества.

TotalJS

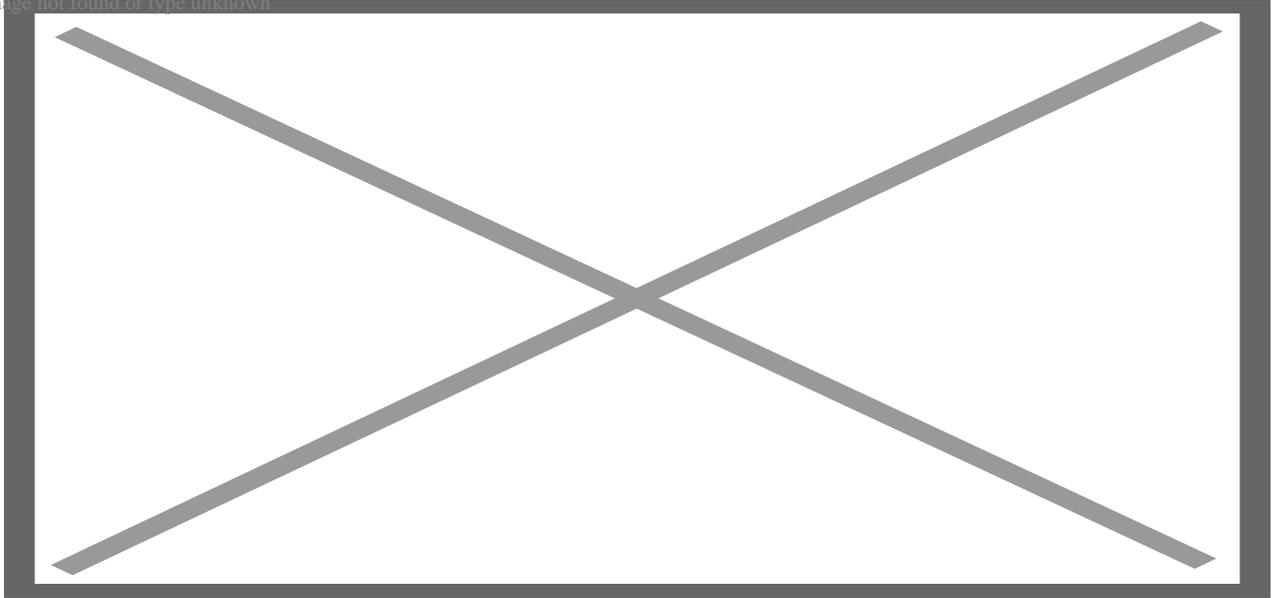
Image not found or type unknown



Платформа Total.js представляет собой набор библиотек, пакетов, компонентов пользовательского интерфейса, практик и готовых продуктов, написанных на JavaScript. Основой платформы является серверный фреймворк под названием Total.js framework и клиентская библиотека под названием jComponent. Обе части были написаны с нуля. В большинстве случаев нам не нужно использовать какие-либо зависимости с платформой Total.js, потому что она содержит все необходимое для разработки. Вся платформа Total.js – продукты, библиотеки, модули, пакеты и компоненты пользовательского интерфейса – с открытым исходным кодом под лицензией MIT.

Moleculer

Image not found or type unknown



Moleculer – это быстрый, современный и мощный фреймворк микросервисов для Node.js. Он предоставляет множество функций для построения и управления микросервисами, таких как:

- Решение на основе обещаний (совместимое с `async/await`)
- Концепция запрос-ответ
- Поддержка событийно-ориентированной архитектуры с балансировкой
- Встроенный реестр сервисов и динамическое обнаружение сервисов
- Балансировка нагрузки на запросы и события (по кругу, случайным образом, по потреблению процессора, задержке, чередованию)
- Функции отказоустойчивости (Circuit Breaker, Bulkhead, Retry, Timeout, Fallback)
- Встроенное решение для кэширования (Memory, MemoryLRU, Redis)
- Подключаемые регистраторы (Console, File, Pino, Bunyan, Winston, Debug, Datadog, Log4js)
- Транспортёры (TCP, NATS, MQTT, Redis, NATS Streaming, Kafka, AMQP 0.9, AMQP 1.0)
- Сериализаторы (JSON, Avro, MsgPack, Protocol Buffer, Thrift)
- Архитектура без хозяина, все узлы равны
- Встроенная функция метрики с репортерами (Console, CSV, Datadog, Event, Prometheus, StatsD)
- Функция трассировки с экспортерами (Console, Datadog, Event, Jaeger, Zipkin).

Fastify

Начните свое путешествие с Fastify. Fastify сфокусирован на предоставлении лучшего опыта для разработчиков с наименьшими накладными расходами и мощной архитектурой плагинов. Napi и Express вдохновляют его, и это один из самых быстрых веб-фреймворков в отрасли. Начните свое путешествие с Fastify.

```
npm install fastify
```

Он обеспечивает высокую производительность и в зависимости от сложности кода может обслуживать до 30 тысяч запросов в секунду. Он полностью расширяем с помощью крючков, плагинов и декораторов. Даже если это не является обязательным, Fastify рекомендует использовать JSON Schema для проверки маршрутов и сериализации результатов, внутри Fastify компилирует схему в высокопроизводительную функцию.

Заключение

Создание API может быть сложной задачей, но с правильным фреймворком жизнь становится намного проще. Изучите перечисленные выше фреймворки для вашего следующего API и микросервисов и выберите наиболее подходящий. Как только ваш API будет готов, не забудьте про его мониторинг.

Дата Создания

30.06.2023