Technologieauswahl

In diesem Kapitel wird der Entwurf des Systems beschrieben. Ziel des Kapitels ist es, einen detaillierten Überblick über die Systemarchitektur und die Technologieauswahl zu geben sowie die einzelnen Komponenten des Systems darzustellen. Es wird erläutert, warum bestimmte Technologien und Architekturen gewählt wurden und wie die verschiedenen Komponenten des Systems miteinander interagieren.

Zunächst wird die Auswahl der verwendeten Technologien und Frameworks begründet. Anschließend wird die Systemarchitektur detailliert beschrieben, gefolgt vom Datenbankentwurf und dem API-Design. Ein weiterer Abschnitt widmet sich der Implementierung der Authentifizierung und Benutzerverwaltung, gefolgt von der Datenvalidierung. Die Backend-Logik wird ebenfalls im Detail erläutert.

Bun

Bun ist eine JavaScript-Runtime Umgebung für den Server, die anders als Node.js oder Deno nicht auf der V8-Engine basiert, sondern auf einer eigenen JavaScript-Engine, welche mithilfe von Apples WebKit Engine implementiert wurde. Bun wurde zudem in einer “low-level general” Programmiersprache namens Zig geschrieben, welche von Rust und C inspiriert ist. Die Entscheidung für Bun fiel aufgrund der hohen Performance und Sicherheit, die durch die Verwendung von Zig und der WebKit-Engine gewährleistet wird. Bun ermöglicht es, serverseitige Anwendungen in JavaScript zu entwickeln und auszuführen. Bun wurde von Jarred Summer entwickelt und ist eine Open-Source-Software, die unter der MIT-Lizenz veröffentlicht wird. Die erste offizielle Version von Bun (Bun 1.0) wurde im September 2023 veröffentlicht.

Funktionen von Bun

Bun bietet eine Reihe von Funktionen, die es zu einer Plattform für die Entwicklung von serverseitigen Anwendungen machen. Dazu gehören:

- Kompatibilität mit Node.js
- Hohe Laufleistung und geringer Speicherverbrauch
- Vereinfachte Modulverwaltung
- TypeScript-Unterstützung
- Web-Standard-APIs
- JSX-Unterstützung
- Watch-Modus für automatisches Neuladen von Änderungen
- Cross-Plattform-Unterstützung

Bun vs. Node.js

Anders als Node.js ist bun nicht auf npm angewiesen und benötigt keine externen Abhängigkeiten zur Ausführung. Stattdessen wird eine integrierte Standardbibliothek verwendet, die Funktionen wie HTTP-Server, Dateisystemzugriff und Netzwerkkommunikation bereitstellt. Dies macht die Entwicklung und Bereitstellung von Anwendungen mit Bun einfacher und sicherer. Bun basiert zudem anders als Node.js nicht auf der von Google entwickelten V8-Engine, sondern auf einer Erweiterung von JavaScriptCore, die von Apple entwickelt und bereitgestellt wird. JSC priorisiert schnellere Startzeiten und geringeren Speicherverbrauch, was zu einer etwas langsameren Ausführungsgeschwindigkeit führt. V8 priorisiert hingegen die Ausführungsgeschwindigkeit mit mehr Runtime-Optimierungen, was zu einem höheren Speicherverbrauch führen kann. Das führt dazu, dass Bun bis zu 4xmal so schnell startet als Node.js

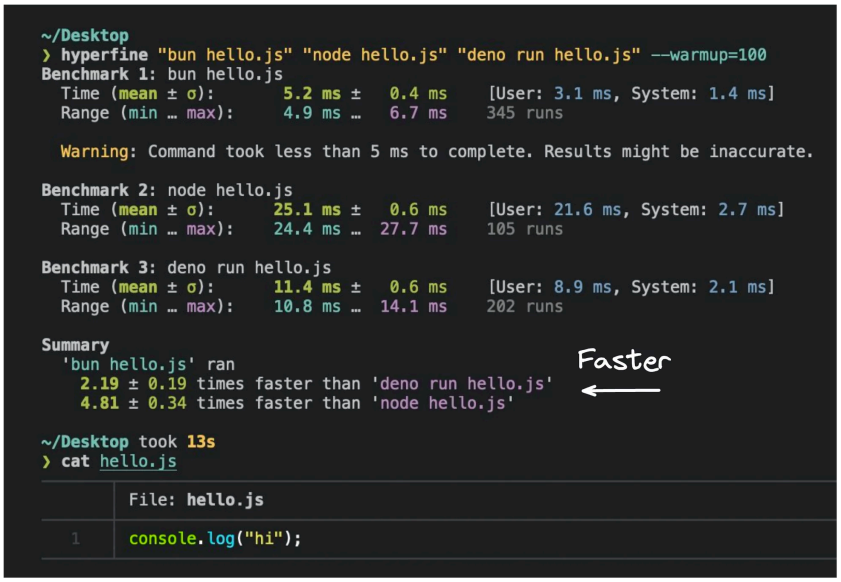

Figure 1: Bun vs. Node.js - Startzeitvergleich, Quelle: Builder.io

Die Benchmark-Ergebnisse, welche in Abbildung 6 gezeigt werden, zeigen eine Verbesserung von mehr als siebzehnmal so schnell wie übliche Paketmanager.

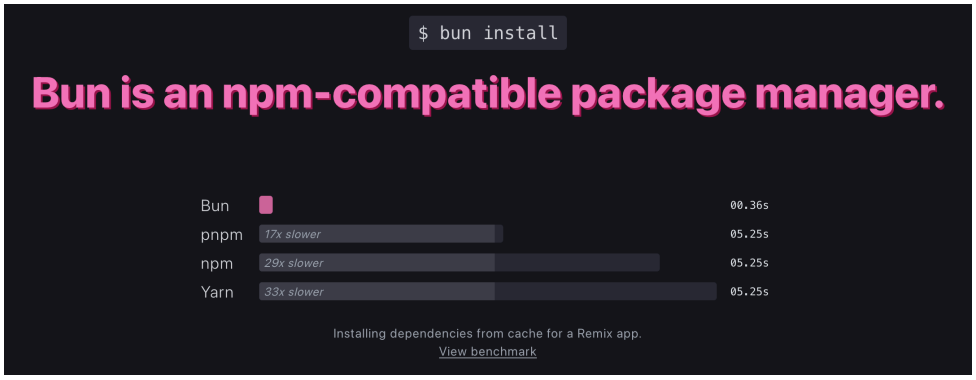

Figure 2: Bun vs. andere Paketmanager - Benchmark-Ergebnisse, Quelle: Bun

Während Node.js eine gute Runtime-Umgebung für JavaScript ist, werden TypeScript Dateien in Node.js nicht direkt unterstützt. TypeScript-Dateien müssen zuerst in JavaScript-Dateien kompiliert werden, bevor sie in Node.js ausgeführt werden können. Bun hingegen unterstützt TypeScript-Dateien direkt, was die Entwicklung von serverseitigen Anwendungen in TypeScript vereinfacht. TypeScript-Dateien können direkt mit dem Befehl bun “Dateiname.ts” ausgeführt werden.

Deshalb wurde Bun als Server-Runtime für die Entwicklung des Webshops gewählt, da es eine hohe Performance, Sicherheit und TypeScript-Unterstützung bietet und die Entwicklung von serverseitigen Anwendungen in JavaScript und TypeScript vereinfacht.

Hono

Hono ist einfaches und ultraschnelles Web-Framework, welches auf jeder JavaScript-Runtime-Umgebung läuft. Entwickelt wurde Hono von Yusuke Wada und ist eine Open-Source-Software, die unter der MIT-Lizenz veröffentlicht wird. Hono wurde speziell für die Entwicklung von Webanwendungen und APIs entwickelt und bietet eine Reihe von Funktionen, die es zu einer leistungsstarken Plattform für die Entwicklung von Webanwendungen machen.

Vorteile von Hono
Die Entscheidung für Hono als Web-Framework wurde aufgrund mehrerer Schlüsselfaktoren getroffen:

- **Ultraschnell und effizient**: Der Router "RegExpRouter" ist besonders schnell und arbeitet nicht mit linearen Schleifen, was eine schnelle und effiziente Routenauflösung ermöglicht. Dies macht Hono ideal für Anwendungen, die eine hohe Geschwindigkeit und geringe Latenz erfordern.
- **Leichtgewichtig und modular**: Hono ist äußerst leichtgewichtig und hat keine externen Abhängigkeiten. Mit dem hono/tiny-Preset beträgt die Größe von Hono weniger als 14 kB, was im Vergleich zu anderen Frameworks sehr kompakt ist. Trotz seiner geringen Größe bietet Hono eine Vielzahl von Middleware- und Hilfsfunktionen, die es einfach machen, leistungsstarke Anwendungen zu entwickeln.
- **Multi-Runtime-Unterstützung**: Hono ist äußerst vielseitig und kann auf einer Vielzahl von Plattformen eingesetzt werden, darunter Cloudflare Workers, Fastly Compute, Deno, Bun, AWS Lambda und Node.js. Dadurch ist es möglich, die gleiche Codebasis auf verschiedenen Plattformen zu verwenden, was die Entwicklung und Wartung von Anwendungen vereinfacht.
- **Middleware inklusive**: Hono bietet eine umfangreiche Sammlung von Middleware, benutzerdefinierten Middleware und Hilfsfunktionen, die es Entwicklern ermöglichen, weniger Code zu schreiben und mehr zu erreichen. Von der Basic Authentication bis zur GraphQL Server-Unterstützung bietet Hono alles, was für die Entwicklung leistungsstarker Webanwendungen erforderlich ist. Zudem auch einen kleinen Validator für die Datenvalidierung, um die Datenintegrität zu gewährleisten.

Hono vs. Express.js
Im Vergleich zu Express.js, einem der beliebtesten Web-Frameworks für Node.js, bietet Hono eine Reihe von Vorteilen:

Vorteile von Hono:

- Mikroservices-Architektur: Hono ist speziell für Mikroservices-Architekturen ausgelegt, was die Skalierbarkeit und Modularität von Anwendungen erleichtert.
- Leistung und Skalierbarkeit: Hono bietet Leistungsbenchmarks und effiziente Anfrageverarbeitung, was besonders für hochskalierbare Anwendungen von Vorteil ist.
- Eingebaute WebSocket-Unterstützung: Hono bietet WebSocket-Unterstützung für die Implementierung von Echtzeitfunktionen an.
- TypeScript-Unterstützung: Hono unterstützt TypeScript nativ, was für Typsicherheit und verbesserte Entwicklerwerkzeuge sorgt.
- Aktive Community-Wartung: Hono wird von einer aktiven Entwicklergemeinschaft gepflegt, was regelmäßige Updates und Verbesserungen gewährleistet.

Deshalb wurde Hono als Web-Framework für die Entwicklung des Webshops gewählt, da es vorallem mit Kombination von Bun ultraschnell, effizient, leichtgewichtig, modular und vielseitig ist.

Framework	Runtime	Durchschnitt	Ping	Query	Body
Hono	bun	184,966.48	234,593.57	185,108.2	135,197.67
Hono	Node	42,699.317	60,797.19	56,645.8	10,654.96
Express	node	16,461.68	17,656.74	16,615.32	15,112.98

Die Ergebnisse sind in req/s gemessen

TypeScript

TypeScript ist eine von Microsoft entwickelte Programmiersprache, die eine strikte Typisierung für JavaScript bietet. Sie erweitert JavaScript um statische Typisierung, Klassen, Interfaces und Module, was die Entwicklung von großen und komplexen Anwendungen erleichtert. TypeScript wird zu JavaScript kompiliert und kann in jedem Browser und auf jedem Betriebssystem ausgeführt werden. Diese Eigenschaften tragen zur Steigerung der Produktivität von Entwicklern und zur Verbesserung der Codequalität bei. TypeScript erweitert JavaScript um zusätzliche Features wie Interfaces, Enums, Generics und Module.

Vorteile von TypeScript

1. Statische Typisierung TypeScript bietet eine statische Typisierung, die es Entwicklern ermöglicht, Typfehler bereits zur Entwicklungszeit zu erkennen und zu beheben. Dies führt zu weniger Fehlern und einer höheren Codequalität.
2. Moderne JavaScript-Features TypeScript unterstützt moderne JavaScript-Features wie Klassen, Interfaces, Generics und Module, die die Entwicklung von großen und komplexen Anwendungen erleichtern. Darüber hinaus unterstützt TypeScript asynchrone Programmierung, was die Handhabung asynchroner Operationen vereinfacht.

Aufgrund dieser Vorteile wurde TypeScript als primäre Programmiersprache für die Entwicklung des Webshops gewählt.

Zod

Zod ist eine TypeScript-First-Schema-Validierungs-Bibliothek, die es Entwicklern ermöglicht, Datenstrukturen zu definieren und zu validieren. Zod bietet eine einfache und deklarative API zum Definieren von Schemas und zur Validierung von Daten. Zod ist speziell für TypeScript entwickelt und bietet eine nahtlose Integration mit der Sprache. Zod unterstützt eine Vielzahl von Datentypen, Validierungsregeln und Transformationen, die es Entwicklern ermöglichen, komplexe Datenstrukturen zu definieren und zu validieren.

Vorteile von Zod

1. Typsicherheit Zod bietet Typsicherheit auf der Ebene der Datenvalidierung, was es Entwicklern ermöglicht, Datenstrukturen zu definieren und zu validieren, ohne zusätzlichen Code schreiben zu müssen. Dies führt zu weniger Fehlern und einer höheren Codequalität.
2. Einfache API Zod bietet eine einfache und deklarative API zum Definieren von Schemas und zur Validierung von Daten. Die API ist intuitiv und leicht verständlich, was die Entwicklung von Datenvalidierungslogik vereinfacht.
3. Integration mit Frameworks Zod bietet eine nahtlose Integration mit verschiedenen Frameworks und Bibliotheken, was es zu einer vielseitigen Lösung für die Datenvalidierung macht. Besonderns in Kombination mit Backend-Frameworks wie Hono ist Zod eine gute Wahl für die Datenvalidierung.
4. Leistungsfähigkeit Zod zeichnet sich durch hohe Leistung und geringen Overhead aus, was die Validierung großer Datenmengen effizient macht, ohne die Anwendungsleistung zu beeinträchtigen.

Anwendung im Webshop

Zod wird im Webshop für die Validierung von Benutzereingaben, API-Anfragen und Datenbankantworten verwendet. Durch die Verwendung von Zod wird sichergestellt, dass die

Datenintegrität gewährleistet ist und sichergestellt wird, dass die vom Benutzer übermittelten Daten den erwarteten Formaten und Typen entsprechen, bevor diese in die Datenbank gespeichert werden. Zudem erleichert Zod die Fehlerbehandlung und die Rückmeldung an den Benutzer im Falle von Validierungsfehlern, anhand von detaillierten Fehlermeldungen.

Kinde Auth

Kinde Auth ist eine Authentifizierung- und Benutzerverwaltungslösung, die speziell für SaaS-Anwendungen entwickelt wurde. Es bietet eine Vielzahl von Funktionen, die es Entwicklern ermöglichen, Benutzerkonten zu verwalten, Authentifizierung zu implementieren und Zugriffsrechte zu steuern. Dabei wird drauf geachtet, dass höchste Sicherheitsstandards eingehalten werden, um die Benutzerdaten zu schützen. Die Integration von Kinde Auth im Webshop ermöglicht es, Benutzerkonten zu erstellen, sich anzumelden und Zugriffsrechte zu verwalten. Dadurch wird eine robuste und flexible Authentifizierunginfrastruktur bereitgestellt, die den Anforderungen des Webshops entspricht, da im Webshop mit besonders sensiblen Daten gearbeitet wird.