

Framework pro backend - vlastnosti, výhody použití

Backend framework

- Sada nástrojů a knihoven pro vývoj serverové části aplikací
- Poskytuje:
 - Routing - mapování URL na akce
 - Zpracování HTTP požadavků a odpovědí
 - Připojení k databázi (ORM)
 - Autentizaci a autorizaci
 - Šablonování a API endpoints
- Umožňuje rychlejší vývoj a standardizaci kódu

Výhody:

- **Rychlejší vývoj**
- **Bezpečnost** - ověřené řešení běžných problémů
- **Komunita** - dokumentace, tutoriály, podpora
- **Standardy** - konzistentní struktura projektu
- **Testovatelnost** - integrované testovací nástroje

Nevýhody:

- **Křivka učení** - nutnost naučit se framework
- **Overhead** - může být zbytečný pro malé projekty
- **Vendor lock-in** - závislost na frameworku
- **Omezení** - framework určuje architekturu

- Přehled jazyků pro backend vývoj v roce 2026:

- **PHP** - Laravel, Symfony
- **Python** - Django, FastAPI
- **JavaScript/TS** - Express, NestJS
- **GO** - Gin, Fiber
- **C#** - ASP.NET Core
- **Java** - Spring Boot

- Jak se kód vykonává?:

- **Interpretované** (PHP, Python, Ruby):
 - Kód se překládá za běhu
 - Pomalejší, ale flexibilní
- **Kompilované** - Go:
 - Kód se překládá před spuštěním do binárky
 - Rychlejší, ale delší build time
- **JIT kompilace** (Java, C#, Node.js)
 - Kombinace - kompilace za běhu
 - Optimalizace hot paths

- Kritéria pro výběr frameworku:

- **Techincká:**
 - Výkon
 - Škálovatelnost
 - Bezpečnost
 - Ekosystém knihoven
- **Praktická:**
 - Křivka učení
 - Dokumentace
 - Komunita a podpora
 - Trh práce (hiring)

- PHP - Zend Engine:

- **Runtime:**

- **Zend Engine** - interpret PHP
- **OPcache** - bytecode cache
- **PHP-FPM** - process manager
- Request-based model (každý request = nova proces)

- **Hlavní frameworky:**

- **Laravel** - nejpopulárnější
- **Symfony** - enterprise-ready
- Slim, CodeIgniter, Nette

Výhody:

- Snadná syntaxe (Laravel)
- Enterprise-ready (Symfony)
- Obrovská komunita
- Eloquent ORM / Doctrine
- Levný hosting

Nevýhody:

- Nižší výkon než kompilované jazyky
- Request-based model limituje real-time
- Snížená reputace starými verzemi

- Python - CPython a ASGI:

- **Runtime:**
 - **CPython** - reference interpreter
 - **GIL** (Global Interpreter Lock)
 - **WSGI** - synchronní interface
 - **ASGI** - async interface (moderní)
- **Hlavní frameworky:**
 - **Django** - batteries included
 - **FastAPI** - async, moderní
 - **Flask** - minimalistický

Vlastnosti	Django	FastAPI	Flask
Filosofie	Batteries included	Async-first	Minimalistický
ORM	Django ORM	SQLAlchemy	Lze SQLAlchemy
Admin panel	Ano	Ne	Ne
Async	Částečně	Plně	Ne
API docs	Ruční	Automatické	Ruční
Use case	Full-stack web	API, ML	Microservices

- JavaScript/TypeScript - V8 a Event Loop:

- **Runtime:**

- **V8 Engine** (Chrome/Node.js)
- **Event Loop** - non-blocking I/O
- **libuv** - async I/O library
- Single-threaded s worker threads

- **Hlavní frameworky:**

- **Express.js** - minimalistický
 - Minimalistický, flexibilní
 - Žádná předepsaná struktura
 - Obrovský npm ekosystém
 - Snadné začít
 - Vhodný pro MVP, API
- **NestJS** - enterprise TypeScript
 - Enterprise-ready Typescript
 - Dependency Injection
 - Modular architecture
 - GraphQL, WebSocket podpora
 - Vhodný pro velké projekty
- **Fastify** - vysoký výkon
- Koa, Hapi

- GO:

- **Runtime:**

- **Kompilace do binárky**
- **Goroutines** - lightweight threads
- **Channels** - komunikace
- Garbage collector
- Žádná VM, žádný interpreter

- **Hlavní frameworky:**

- **Gin** - nejpobulárnější
- **Fiber** - inspirovaný Express
- **Echo** - minimalistický
- Chi, Gorilla Mux

Výhody:

- **Extrémní výkon**
- **Nativní konkurence (goroutines)**
- **Malé binárky, rychlý start**
- **Skvělé pro microservices**
- **Jednoduchá syntaxe**

Nevýhody:

- **Menší ekosystém než Node/Python**
- **Verbose error handling**
- **Žádná generika (do Go 1.18)**
- **Menší komunita**
- **Strmější křivka učení**

- C# - .NET CLR:

- **Runtime:**

- **.NET CLR** (Common Language Runtime)
- **JIT kompilace**
- **Kestrel** - vysokovýkonný server
- Cross-platform (.NET 8+)
- AOT kompilace (optional)

- **Framework:**

- **ASP.NET Core**
- Minimal APIs (lightweight)
- MVC pattern
- Blazor (WebAssembly)
- Entity Framework (ORM)

Výhody:

- **Excelentní výkon (top3)**
- **Nejlepší bezpečnost**
- **Enterprise-ready**
- **Azure integrace**
- **Silný typing - .NET**
- **Skvělé IDE (Visual Studio, Rider)**

Nevýhody:

- **Strmá křivka učení**
- **Microsoft vendor lock-in**
- **Historicky Windows-only**
- **Menší open-source komunita**

- Java - JVM:

- **Runtime:**

- **JVM** (Java Virtual Machine)
- **JIT kompilace** (HotSpot)
- **GraalVM** - polyglot, native image
- Garbage collector (G1, ZGC)
- Vyzrálý ecosystem

- **Hlavní frameworky:**

- **Spring Boot** - de facto standard
- **Quarkus** - cloud-native
- **Micronaut** - microservices
- Jakarta EE (formerly Java EE)

- Výkonnostní srovnání:

- **TechEmpower Benchmarks:**

- Co to je ? = Open-source srovnání výkonu webových frameworků

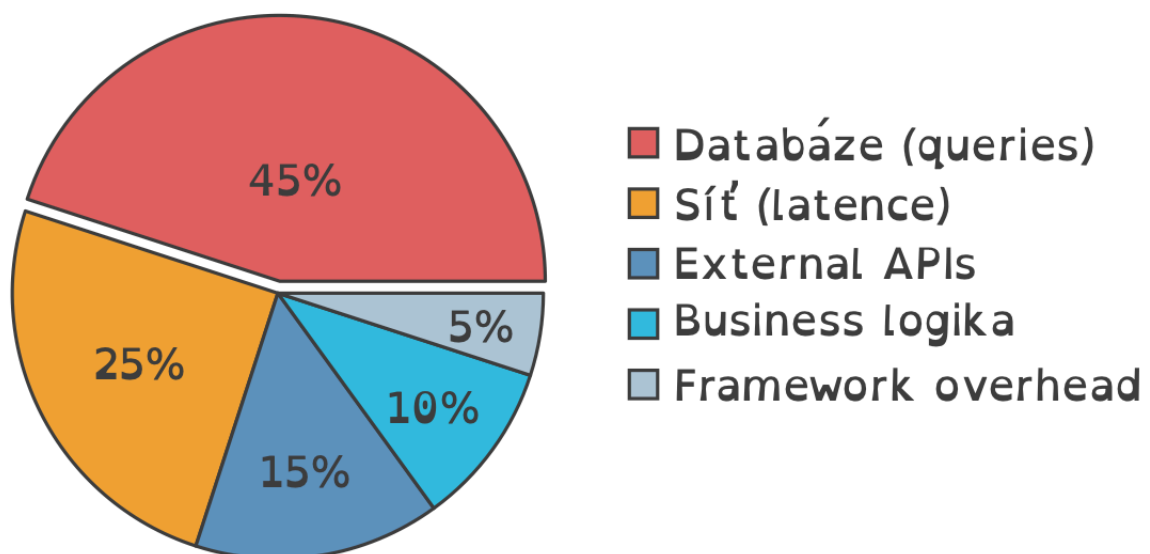
- Testuje reálné scénáře:

- JSON serialization
 - Single/Multiple database queries
 - Fortunes (template rendering)
 - Data updates
 - Plaintext responses

- **Kdy na výkonu záleží?:**

- High-traffic aplikace (miliony requests/den)
 - Real-time systémy (trading, gaming)
 - Infrastrukturní služby (API gateway, proxy)
 - Cost-sensitive deployment (méně serverů)

Kde se skutečně ztrácí čas:



- **Kdo používá které frameworky:**
 - **Enterprise / Banking:**
 - **Spring Boot** - Goldman Sachs, JP Morgan
 - **ASP.NET Core** - Stack Overflow, Microsoft
 - **Big Tech:**
 - **Go** - Google, Uber, Dropbox
 - **Django** - Instagram, Pinterest
 - **Startupy:**
 - **Express/NestJS** - většina JS startupů
 - **FastAPI** - AI/ML startupy
 - **Web agencies:**
 - **Laravel** - webové projekty
 - **Symfony** - enterprise PHP