Téma: Využití jazyka Kotlin pro vývoj serverových aplikací

Cíl: Zhodnocení současných možností jazyka Kotlin pro tvorbu serverových aplikací

Otázky:

Proč použít Kotlin na serverový vývoj?

Jaké jsou jeho přínosy?

Lze Kotlinem a jeho proprietárními nástroji plně nahradit současná řešení při vývoji na serverech?

Osnova:

Porovnání jazyka Java a Kotlin

Výběr a hodnocení frameworků pro serverový vývoj

* Web-Microframeworky
* Dependency Injection
* Testování
* ORM/DB

**Metoda hodnocení web frameworků**

* Vybrány mikro-frameworky, které jsou nejvíce využívány + mají nativní podporu Kotlinu:
  + Vert.X
  + Ktor
  + Spark
  + Javalin
  + Micronaut

**Škálovatelnost** – zhodnocení možností frameworku při velkém rozvoji aplikace

**Ekosystém** – míra využití Kotlin features (corutiny), použitá architektura

**Výkonost** – měření výkonosti frameworku (využity ext. Zdroje), velikost projekt/warko, využití paměti

**Modifikovatelnost** – míra závislosti frameworku na konkrétní implementace/míra volnosti pro přizpůsobení

Moduly – mandatorní (vždy přítomné) vs. Volitelné

Komponenty – Možnost rozšíření (míra zapouzdření, rozšiřitelnosti), případně jejich záměna

**Funkčnost** – funkce, které framework poskytuje

Zhodnocení funkčnosti (chybí některá, je nadbytečná, jsou logicky uspořádané (rodič – potomek) a správně seskupeny)

Podpora jazykových mutací

Podpora AJAX

Podpora ORM

Moduly Security (+ ošetření současných hrozeb), Templating, Caching, Form validation, Page Navigation (+ POST, Redirect…)

**Komunita** – oblíbenost v komunitě, míra podpory

**Podpora** – frekvence verzí, dokumentace

**Praxe** – učící křivka, tech. Komplexnost, složitost orientace v projektu, složitost založení projektu, nástroje

**Testovatelnost** – jak je navrženo testování/lze kompletně pokrýt testy



# Historie vývoje serverových aplikací

Server je počítačový program, nebo zařízení jež poskytuje funkcionalitu (služby) jiným programům nebo zařízením, kteří se označují jako klienti. První zmínka o serverech se datuje k roku 1969 v RFC 5 (<https://tools.ietf.org/html/rfc5>), což je jeden z dokumentů, který popisuje ARPANET, globální síť, jež je předchůdcem internetu tak jak ho známe dnes. V prvopočátcích byl vývoj serverových aplikací prováděn velmi ad-hoc, tvořila se nestandardizovaná API a využívali se programovací jazyky, které byly zrovna k dispozici, jednalo se nejčastěji o jazyk C. Programovací jazyk C byl vyvinut v roce 1972 určený pro všeobecné použití, jedná se o imperativní, procedurální jazyk. Každý web server měl vlastní API např. NSAPI, Microsoft ISAPI. Toto v roce 1993 vyřešil CGI (Common Gateway Interface) standart, který umožnil webovým serverům poskytovat jednotné rozhraní pro zpracování požadavků tzv. http requestů. Velkou výhodou tohoto přístupu byla technologická volnost pro vývojáře, kteří si mohli vybrat jazyk, který jim vyhovoval pro vývoj.

Kromě jazyka C se v té době pro serverový vývoj využíval jazyk Perl. Perl je skriptovací jazyk, který byl vyvinut v roce 1987, jedná se o dynamicky slabě typový jazyk. Jméno jazyka Perl je zkratka pro Practical Extraction and Reporting Language, jazyk, jak je patrné z názvu byl primárně určený pro vývoj skriptů pro systémy UNIX. První polovinu 90. let ve vývoji pro servery tedy více či méně ovládl jazyk Perl. Avšak pořád neexistoval jazyk, který by plně uspokojil požadavky vývojářů a plně podporoval serverový vývoj. Samotný jazyk Perl oplýval nedostatky jazyk nebyl určen pro serverový vývoj, je velmi volný sémanticky a syntakticky a primárně určen pro vývoj skriptů, tím pádem byl méně použitelný pro vývoj enterprise aplikací, které se stávali poměrně obtížně čitelné.

V této době se začali tvořit jazyky, které se používají i v dnešních dnech. V roce 1991 se objevil jazyk Python, interpretovaný a objektový jazyk, který byl určen jak pro malé ale i enterprise řešení, které se vyznačovali čistotou kódu. Python je silně dynamicky typovaný. Dá se říct, že se jednalo o volného následníka jazyka Perl, který se dostal velmi výrazně do popředí v žebříčcích v průběhu tohoto desetiletí, kdy je přiřazován k nejvíce progresivnějším jazykům.

O pár let později v roce 1995 se objevil velmi populární skriptovací jazyk na serverech jednalo se o jazyk PHP. Jazyk PHP je interpretovaný jazyk, který je multi-paradigmatický, avšak nejčastěji se programuje v PHP buďto procedurálně, ale v současné době spíše objektově, s využitím frameworků, které jsou v současné době v podstatě nutností, při použití PHP.

Ve stejný rok se poprvé objevil jazyk Java, který způsobil ve světě vývoje poměrně velký boom. Java přišla s myšlenkou WORA – write once, run everywhere. Java je objektově orientovaný, interpretovaný a silně staticky typovaný jazyk, který se kompiluje do bitecodu, který je následně spuštěn v JVM – Java virtuál machine, který zajišťuje jeho platformní nezávislost. Další vlastností díky, které se jazyk stal populárním je jeho syntaktická podobnost s jazyky z rodiny C, které se v té době hojně využívali, avšak oproti nim nebyla Java tak nízko úrovňový jazyk, navíc disponovala velmi kvalitní správou paměti tzv. Garbage collector, který se stará o dealokaci paměti automaticky. Jedním z výrazných principů, je zpětná kompatibilita, což je i současně poměrně velkou brzdou v rozvoji jazyka.

Po miléniu na rozmach Javy zareagoval Microsoft a vydal jazyk C#. Jazyk C# je objektově orientovaný, kompilovaný a silně staticky typovaný jazyk. Není orientován primárně jen na servery a jeho využití je mnohostranné a oproti Javě se v C# vyvíjejí i hry, avšak není tolik multiplatformní jako Java, byť podporuje technologii IL code (Intermediate Langugae).

Až v současném desetiletí se začali objevovat poměrně inovativní přístupy, které poměrně výrazně změnili pohled na serverový vývoj. V roce 2010 společnost Twitter představila jazyk JavaScript, který byl do té doby používán výhradně pro implementaci klientské strany, na straně serveru. První řešení vykazovalo výkonnostní problémy, avšak po mnoha optimalizacích architektury se podařilo získat velmi uspokojivé výsledky. V roce 2013 byl implementován jeden z prvních větších komerčních projektů, realizovaných pomocí JavaScriptu, konkrétně pomocí knihovny Node.js, který umožňuje jednoduché použití JavaScriptu pro serverové aplikace. Jazyk JavaScript byl vydán již v roce 1995 a sloužil jako jazyk na tvorbu skriptů na webové stránky. Jedná se dynamicky typovaný jazyk, který je multi-paradigmatický. Nejčastěji se využívá v kombinaci s nějakým frameworkem na klientu i na serveru.

Po půli prvního desetiletí milénia se objevila tendence hledat alternativy pro Javu, která byla podpořena poměrně dlouho pauzou mezi vydávanými verzemi, která trvala od roku 2006 do roku 2011. Java díky zpětné kompatibilitě a malé aktivitě tehdejšího vlastníka jazyka společnosti Sun, nepřidávala zásadní vylepšení prostředí a jazyka jako takového. Vývojáři hledali alternativy, avšak nikdo nechtěl opouštět svět, který se točil kolem poměrně kvalitního prostředí JVM. Začali se objevovat jazyky jako Clojure, Scala, Groovy, které byly postavené nad JVM, využívali její možnosti mnohem více než Java a projevovalo se to lepším výkonem než, který měla samotná Java. Výše zmíněné jazyky byly spíše pokusy, které využívali netradiční přístupy. Díky tomu faktu a s tím související poměrně náročné adaptaci a také ne zcela příznivé kompatibilitě s Javou se tyto jazyky stály více méně na pozadí a aktivně je v projektech využívalo poměrně malé množství vývojářů v porovnání s původním jazykem postaveným nad JVM a to Javou.

Jazyk, který zaznamenal úspěch a přitáhl pozornost mnoha vývojářů byl jazyk Kotlin, který sázel na plnou kompatibilitu s Javou a velmi rychlou adaptaci Java vývojáři. Kotlin je silně staticky typovaný jazyk, který umožňuje vývoj podle nejpoužívanější paradigmat (procedurální, objektový, funkcionální). Kotlin je plně objektový jazyk, oproti Javě v něm nenajdeme primitivní datové typy. První zmínky o Kotlin se datují do července roku 2011.

# Současné technologie

Zmapovaní současných nejpoužívanějších technologií, které se používají pro vývoj serverových aplikací bylo provedeno na základě více metrik. Každé hodnocení totiž pokrývá jen část domény a jen ve společném kontextu mohou dát směrodatný výsledek.

Pro hodnocení zájmu o jazyk jsem použil metriku, která hledá trendy ve vyhledávání klíčových slov, která se vztahují k četnosti vyhledávání tutoriálů programovacích jazyků, což se dá pokládat za množství zájmu, které se technologii dostává. Toto hodnocení se jmenuje PYPL (Popularity of Programming Language). K červnu 2019 mluví statistiky jasně o vítězi, kterým se stává jazyk Python s 28 % a přírůstkem 4,7 %, druhá v pořadí je Java s 20 % a s úbytkem 1,8 %, na třetím místě je JavaScript s již pouze 8 % a drobným úbytkem v řádu desetin procent, na čtvrtém místě se se 7 % umisťuje jazyk C#, která má úbytek zhruba půl procenta, páté místo obsadilo PHP s necelými 7 % a poměrně vysokým úbytkem v podobně 1 %. Jazyk Kotlin obsadil třinácté místo s 1,5 % a přírůstkem 0,5 %, což je v procentuálním vyčíslení největší posun v celém žebříčku. Je zřejmé že vývojáři mají tendenci učit se nové a netradiční jazyky.



Další metrikou je TIOBE Index, který měří popularitu jazyků napříč vyhledávači, kde sleduje zájem o jazyky podle vyhledávání a neomezuje se pouze na tutoriály například jako výše zmíněný PYPL. Z grafu, je patrný v posledních letech prudký růst dotazů na jazyk Python, u ostatních jazyků je trend spíše klesající. Zajímavý je výrazný propad, který Java zažila po roce 2016, který utnul její poměrně strmý růst.



Ben Fredirickson vytvořil hodnocení jazyků podle struktury repositářů na veřejném Git repository GitHub, který má celkem 37 miliónů uživatelů, 75 miliónů repositářů, 1,25 miliard událostí. Každá událost je přiřazena k určitému, repositáři a uživateli, Fredirickson agregoval data za měsíc a každou interakci přiřadil ke konkrétnímu jazyku, jako jeho použití. Z toho vznikly hodnoty Monthly Active User, které dosahují v roce přes 2 miliony aktivních uživatelů za měsíc. Díky tomu mají statistky poměrně dobrou vypovídající hodnotu.



První metrika byla podle počtu aktivních uživatelů za měsíc. U této metriky vyšel jako vítěz JavaScript, který měl 23 % a klesající trend a pomalu se vrací na úroveň z roku 2011 kdy začal expandovat, druhý Python s 15 % s mírným růstem, Java třetí se 14 % a lehce klesajícím trendem, PHP se umístilo na šestém místě s 6 % a velmi klesajícím trendem, na sedmém C# s 5 % stabilně rostoucí. Kotlin se objevil na patnáctém místě s 1,3 % a velmi prudkým růstem.



Druhou statistikou jsou jazyky, které mají procentuálně největší růst aktivních uživatelů. Je to tedy statistika pro jazyky, které jsou nejvíce progresivní a programátoři o ně jeví poměrně velký zájem. Nejvíce progresivním jazykem je Go, jedná se kompilovaný, silně staticky typovaný, multi-paradigmatický jazyk odvozený z jazyka C vyvinutý společností Google. Záměrem bylo poskytnout jazyk podobný jazyku C, avšak který je obohacen o vlastnosti moderních programovacích jazyků, a přitom zůstal stejně rychlý a efektivní jako jazyk C. Hlavními benefity je bezpečnost paměti, garbage collector a gorutiny, které umožňují paralelizaci na úrovni jazyka, avšak zdrojově o dost méně náročnější než klasická vlákna. Jazyk však nedisponuje všemi vlastnostmi, na které jsme zvyklí z jiných jazyků např. dědičnost, generika a další jazykové funkce jsou do jazyka Go přidávány poměrně konzervativně, což mu někteří kritici vyčítají. Do jazyka Go jsou přepisovány síťové a webové aplikace na kterých je běžně zaznamenáno zrychlení v některých případech až sto násobné. Druhý jazyk je TypeScript, což je nadstavba pro JavaScript, která poskytuje typovou kontrolu a další vlastnosti, které jsou používány v objektových jazycích. Třetí je jazyk Kotlin, který zažil největší růst od roku 2017, což se dá spojit s oficiálním představením podpory jazyka Kotlin pro Android.



Třetí statistikou jsou naopak největší procentuální odliv aktivních uživatelů od jazyků. Zde můžeme pozorovat, které jazyky stagnují v očích vývojářů, kteří je méně často využívají, respektive nepřispívají do repositářů s danou technologií. Ze serverových jazyků velmi upadá Ruby, nad kterým se používá oblíbený webový framework Rails tzv. Ruby on Rails. Dalším velmi populárním jazykem, který upadá zájmu je jazyk PHP, pokles je nejspíše způsoben růstem inovativních jazyků, avšak PHP zůstává ve středu pozornosti, co se týče využívání. To potvrzuje průzkum W3Techs, který zkoumá technologie webových stránek, do statistiky je zahrnuto horních 10 miliónů stránek. Za zmínku stojí i pokles jazyka Perl, který byl zmíněn v kapitole o historii vývoje.



Zajímavá statistika je také přehled oblíbenosti, který je omezen na funkcionální jazyky, které byly v popředí zájmů i pro vývoj serverových aplikací. Avšak trend pro všechny funkcionální jazyky je velmi klesající a nových

# Trendy ve vývoji serverových aplikací

PHP, JavaScript, Java, Kotlin

Současné používané technologie

Porovnání současných technologií a trendů

Směr, kterým se ubírá současný trend vývoje bacené řešení

GraalVM