Adam Krajčík

Programátor

„Kto neide dopredu, ide dozadu, neexistuje nehybný stav.“
– Vissarion Grigorjevič Bělinskij

Vzdelanie

2013 – dnes **Bakalárske štúdium**, *Masarykova Univerzita - Fakulta Informatiky*, Brno.
špecializácia: Aplikovaná Informatika

2009 – 2013 **Stredoškolské štúdium s maturitou**, *Gymnázium Párovská 1*, Nitra.
špecializácia: Matematika a informatika

Skúsenosti

Profesionálne

2014 – dnes **IT konzultant**, *Temo s.r.o.*, Nitra.
správca siete, nákup techniky, riešenie technických problémov

Vlastné projekty

2015 **Chrome plugin PutlockerSerials**.
Sledovanie postupu sledovania seriálov a kontrola vydania nových časť. Podpuruje import a export dát. Dostupné na `https://github.com/AdamKrajcik/PutlockerSerials`.

Programovacie jazyky

základné **Haskell, C#, PowerShell**.
mierne pokročilé **C, C++, HTML, CSS, JavaScript, XML**.
pokročilé **Java, Bash**.

Technológie

webové **HTML 5, jQuerry, AJAX**.
grafické **GIMP, JavaFX, CSS**.
serverové **Apache, Hibernate**.
databázové **MySQL, SQLite**.
iné **Git, SVN, Maven**.

Mánesova 12a – 602 00 – Brno
(+421) 915 651 818 • *adam.krajcik@gmail.com* • *AdamKrajcik*

Jazyky

Slovenský	**materinský**.
Anglický	**stredne pokročilý(B2)**.
Japonský	**začiatočník**.
Španielsky	**začiatočník**.

Vodičský preukaz

Skupina B	**pasívny vodič**.

Záľuby

Čítanie, politika, technológie, modelovanie, stavba a programovanie jednoduchých elektronických zariadení, Japonsko a jeho kultúra, psychológia a jej využitie.

Mánesova 12a – 602 00 – Brno

(+421) 915 651 818 • *adam.krajcik@gmail.com* • *AdamKrajcik*

Úvod

V súčasnosti je vývoj softvéru neúprosne rýchly. Každý deň sa na svete objavujú nové a nové aplikácie. Či už sú to desktopové, mobilné alebo cloudové aplikácie, každá z nich bola implementovaná podľa určitého vzoru – architektúry [1]. V prípade veľkých projektov je jej správny výber veľmi dôležitý, pretože zlá možnosť môže spôsobiť programovo neriešiteľné alebo finančne náročné problémy. Jedná sa hlavne o serverové enterprise (podnikové) aplikácie, ktoré musia podporovať pripájanie rôznych klientov zostávajúcich z rôznorodých desktopových/mobilných prehliadačov a aplikácií. Systém komunikuje na princípe požiadaviek, ktoré sú spracovávané a následne je odoslaná odpoveď so žiadanými informáciami.

Existuje viacero vzorov, ako tieto aplikácie navrhovať a následne použitím zaužívaných frameworkov implementovať. Najrozšírenejšia jemonolitická architektúra [2]. Aplikácia vytvorená podľa tejto architektúry pracuje ako samostatná jednotka, ktorá spája všetku funkcionalitu (UI , prácu s databázou, pracovnú logiku) do jedného procesu. Druhým vzorom, ktorý k tomuto problému pristupuje práve opačným prístupom sú mikroslužby [3] (ďalej len MS). Aplikácia navrhnutá na princípe MS je modulárna, t. j. sa skladá z množstva nezávislých komponent, ktoré pracujú samostatne a sú nahraditeľné. Sú nimi služby napísané v rôznych programovacích jazykoch, čím sa môže zabrániť vzniku obmedzení spôsobených nepoužiteľnosťou daného jazyka v určitých situáciách. Využíva ju viacero obrovských internetových služieb, ako napríklad Netflix alebo Amazon.

Platforma SilverWare [4] je projektom tímu z firmy Red Hat. Je to implementácia MS postavená na už existujúcich technológiách a zaužívaných frameworkoch. Je prevažne určená pre enterprise aplikácie, ale vhodná aj pre menšie projekty. Jej cieľom je zjednodušiť vstup programátora do sveta MS a umožnil mu naplno využiť všetky výhody, ktoré aplikácia navrhnutá podľa tejto architektúrou môže mať.

Pri tvorbe riešenia bolo prvým krokom naštudovanie si systému a v ňom použitých technológií. Nasledovala analýza fungovania komunikačných metód jednotlivých technológii. Tu platforma podporuje 4 poskytovateľov mikroslužieb, ktorými sú ActiveMQ, Camel, Vert.X a Weld, pričom Každý z nich využíva iný spôsob komunikácie (napr. správy, prúdy). To spôsobuje nekompatibilitu prepojenia medzi jednotlivými komponentami priamo. Prvým návrhom riešenie bolo vytvorenie prostredníka, ktorý skonvertuje požiadavky zaslané jednou službou do formátu, ktorému bude rozumieť služba, ktorá má požiadavky prijať a zaslať odpoveď opačným smerom. Následne vedúci práce doplnil do návrhu použitie medziformátu na zníženie počtu konvertorov a zvýšenie efektivity. Návrh sa ešte raz skontroloval, namodelovali sa možné kritické situácie a ich riešenia, a plán bol schválený. Potom nasledovalo programovanie implementácia komunikačnej vrstvy. Tá je napísaná v jazyku Java. Sú v nej použité rovnaké technológie, aké aj v systéme, doplnené o rozhranie na prácu s medziformátom.

Práca popisuje technológie a proces tvorby komunikačnej vrstvy, ktorá prešla integračnými testami a bola nasadená do systému, kde je plne využívaná. Vrstva tiež dosiahla vo výkonnostných testoch pri väčšom zaťažení dobré výsledky aj malú chybovosť . Ako sa bude MS rozvíjať, platforma SilverWare sa stane dôležitou, voľne šíriteľnou implementáciou tejto architektúry. Systém je voľne dostupný na stránke `http://www.silverware.com`.

Register

Citácie

[1] Wikipedia. *Enterprise software — FromWikipedia, the free encyclopedia*. [Online; prístup 16-november-2016]. 2016. URL: `https://en.wikipedia.org/wiki/Enterprise_software`.

[2] Chris Richardson. *Pattern: Monolithic Architecture*. [Online; prístup 16-november-2016]. 2014. URL: `http://microservices.io/patterns/monolithic.html`.

[3] Martin Fowler. *Microservices*. [Online; prístup 16-november-2016]. 2014. URL: `http://www.martinfowler.com/articles/microservices.html`.

[4] SilverWare. *SilverWare.io*. [Online; prístup 16-november-2016]. 2014. URL: `http://www.silverware.io/`.

Môj obrázok