

Askalot meets Harvard Courses at edX

[Askalot2edX]

Moduly systému (Zmena architektúry)

Tím: číslo 6, AskEd
Vedúci tímu: Ing. Ivan Srba
Členovia tímu: Černák Martin, Gallay Ladislav, Hnilicová Eva, Huňa Adrián, Jandura Filip,
Žuffa Tibor
Akademický rok: 2015/2016
Autor: Černák Martin, Gallay Ladislav, Hnilicová Eva, Huňa Adrián, Jandura Filip,
Žuffa Tibor
Verzia číslo: 1
Dátum poslednej zmeny: 14.12.2015

Úvod

Tento dokument obsahuje všetky navrhnuté a implementované časti modulu *Zmena architektúry*. Ku každému z nich sú uvedené tri zložky: analýza, návrh, implementácia.

1. Modulárna architektúra

Analýza

Vzhľadom na rozšírenie pôsobnosti systému Askalot, bolo potrebné prehodnotiť organizáciu v projekte. Bolo potrebné oddeliť rôzne časti systému používané v rôznych prostrediach. Zároveň je potrebné aby niektoré časti kódu boli spoločné pre všetky inštancie projektu. Jednou možnosťou bolo vytvorenie osobitného repozitára a nezávislý vývoj MOOC a University Askaloto. Ak by trebalo rovnakú funkcionálnosť do oboch projektov, musela by sa ručne kopírovať medzi projektami. Ďalšou možnosťou by bolo mať dve nezávislé vetvy v rámci jedného projektu a zmeny by bolo možné presúvať medzi vetvami. Stále to však znamená udržiavať časť toho istého kódu na dvoch miestach. Rozhodli sme sa preto pre jeden projekt a funkcionálnosť rozdeliť do tzv. *rails engines*.

Návrh

Vytvorili sa tri *rails engines* s názvami `shared`, `mooc` a `university`. `Shared` bude obsahovať funkcionálnosť, ktorá je spoločná pre všetky projekty. `Mooc` a `university` budú obsahovať špeciálnu funkcionálnosť pre jeden alebo druhý projekt. Hlavná aplikácia potom bude slúžiť iba ako kontajner. Všetky modely, kontrolery, pohľady, služby, knižnice, testy, databázové migrácie a konfigurácie sa budú nachádzať výhradne v moduloch. Hlavná aplikácia bude iba spúšťačom jedného alebo druhého projektu, ktorý obsahuje vlastný *engine* a *shared engine*.

Implementácia

V zložke `components/` boli vytvorené tri podzložky `shared`, `mooc` a `university`. Každá predstavuje jeden *rails engine*, ktorý sa potom načítava v hlavnom Gemfile podľa toho, o ktoré prostredie ide. Keďže prakticky nebolo možné oddeliť testy do modulov, ostali v zložke `spec/` kde boli tiež vytvorené podzložky `shared`, `mooc` a `university`.

Každý *rails engine* sa podobá na samostatnú Rails aplikáciu s náležitou štruktúrou zložiek a súborov. Treba myslieť na to, že každú triedu alebo modul treba zabaliť do `module Shared`, `module Mooc` alebo `module University` podľa toho, o ktorý *engine* sa jedná. Zároveň aj pri používaní modelov a iných tried treba uvádzať tento modul, napr. `Shared::Category.all`.

V hlavnom súbore `config/routes.rb` sa potom *shared engine* napája na hlavnú cestu projektu. Taktiež aj projektové *engines* sa podmienene napájajú na hlavnú cestu. Okrem toho je dôležitý súbor `components/university/lib/university/engine.rb`, ktorý obsahuje konfiguráciu modulu a spustí sa iba keď je modul načítaný v Gemfile.

V prípade, že sa vyvíja funkcionálnosť, ktorá je potrebná pre oba projekty, patrí do modulu `shared`. Ostatná funkcionálnosť sa umiestňuje do konkrétnych projektových modulov. Pokiaľ treba nejakú rovnakú službu alebo funkcionálnosť implementovať doma spôsobmi, treba ju rozdeliť do modulov, prípadne vytvoriť spoločný kód v `shared` moduli a v projektových moduloch dediť od tejto spoločnej triedy.

Aplikáciu treba spúšťať s konkrétnym prostredím, napr. `RAILS_ENV=fiit_production rails server`. Toto prostredie musí obsahovať názov projektového modulu, ktorý sa použije pri inicializácii aplikácie. Viac informácií o prostrediach sa nachádza v kapitole 2.

2. Prostredia

Askalot je nasadený na univerzitnom serveri na FIIT a jeho produkčná verzia je dostupná na adrese <http://askalot.fiit.stuba.sk>. Nasadzovanie je realizované prostredníctvom doplnku (angl. gem) *capistrano*. V súvislosti so zmenou architektúry je definovaných niekoľko plánov pre nasadenie. Každý plán obsahuje kompletnú konfiguráciu, ktorá zabezpečí nakopírovanie zmenených súborov na server, úprava databázového modelu, predpríprava statických súborov a reštartovanie nasadzovanej inštancie. Každý plán sa v konfigurácii odkazuje na rovnomenné prostredie. Tabuľka 1 zobrazuje plány a ich konfigurácie.

Tabuľka 1: Nasadzovacie plány a ich konfigurácia.

Názov plánu	Modul	Prostredie	Vetva	Port	Zložka na serveri
fiit_production	university	production	gama	13021	askalot-fiit-production
fiit_staging	university	staging	gama	13011	askalot-fiit-staging
fiit_demo	university	demo	gama	13001	askalot-fiit-demo
fiit_experimental	university	experimental	gama	13031	askalot-fiit-experimental
edx_production	mooc	production	gama	13020	askalot-edx-production
edx_staging	mooc	staging	gama	13010	askalot-edx-staging
edx_demo	mooc	demo	gama	13000	askalot-edx-demo
edx_experimental	mooc	experimental	gama	13030	askalot-edx-experimental

Modul opisuje názov použitého modulu, *engine*, ktorý sa použije v kombinácii s modulom *shared*. Hoci každý plán používa rovnomenné samostatné prostredie, vychádza toto prostredie zo štandardných nastavení pre pôvodné prostredia opísané v tabuľke. Momentálne je vo všetkých prostrediach nasadená vývojová vetva *gama* ktorá ako jediná obsahuje novú modulárnu architektúru. V budúcnosti sa tieto vetvy oddelia pre jednotlivé prostredia.

Na serveri sú zdrojové kódy umiestnené v adresári `/home/deploy/projects/` kde je každá inštancia umiestnená vo vlastnej podzložke. Každá inštancia má potom nakonfigurovaný *nginx* server, ktorý prístupuje k *unicorn* HTTP serveru. Jednotlivé inštancie bežia každá na vlastnom porte.

Každá inštancia používa vlastnú Postgres databázu a vlastný Elasticsearch index. Názov databázy a indexu ja podobný názvu zložky, iba namiesto pomlčky obsahuje podržník, napr. `askalot_fiit_production`.