Ročníkový projekt

Webová aplikácia s využitím funkcionálneho programovania

Richard Tóth

Vedúci projektu: **RNDr. Richard Ostertág, PhD**.

Cieľ projektu:

Vytvoriť webovú stránku pomocou funkcionálneho jazyka (Haskell). Na vytvorenie stránky použijem Yesod Web Framework (Backend) a "nevybrané momentálne" (Frontend).

Popis vývoja projektu v zimnom semestri:

Začal som študovaním odbornej literatúry pre programovací jazyk Haskell. Hlavným zdrojom informácií bola pre mňa kniha v elektronickej podobe na stránke http://learnyouahaskell.com.

V jednotlivých kapitolách som lepšie porozumel funkcionálnemu programovaniu, syntaxi haskellu a dozvedel som sa, ako fungujú higher order functions, modules, functors, applicative functors, monoidy, monády, atď. Do svojej teoretickej prípravy som zahrnul aj precvičovanie rôznych úloh zo stránky https://exercism.io/tracks/haskell.

Ďalej som pokračoval štúdiom backend frameworku, Yesod Web Framework, na stránke https://www.yesodweb.com/book, kde som sa naučil, ako vytvoriť prvú webovú stránku, dozvedel som sa, aké funkcie ponúka Yesod na tvorbu aplikácií a aké rôzne pluginy Yesodu môžem použiť.

Po teoretickej príprave som začal vytvárať webovú stránku, použil som Scaffolded site. Scaffolded site je kostra webovej stránky, ktorá umožňuje rozloženie stránky na menšie logické jednotky. Vytvoril som pribižný návrh a dizajn stránky. Prihlasovanie som vytvoril s použitím pluginu pre Yesod HashDb, ktorý umožňuje prihlasovanie pomocou prihlasovacieho mena a hesla, ktoré je hashované. Nakoniec som naprogramoval registráciu na stránku, teda pridávanie nových užívateľov do databázy.

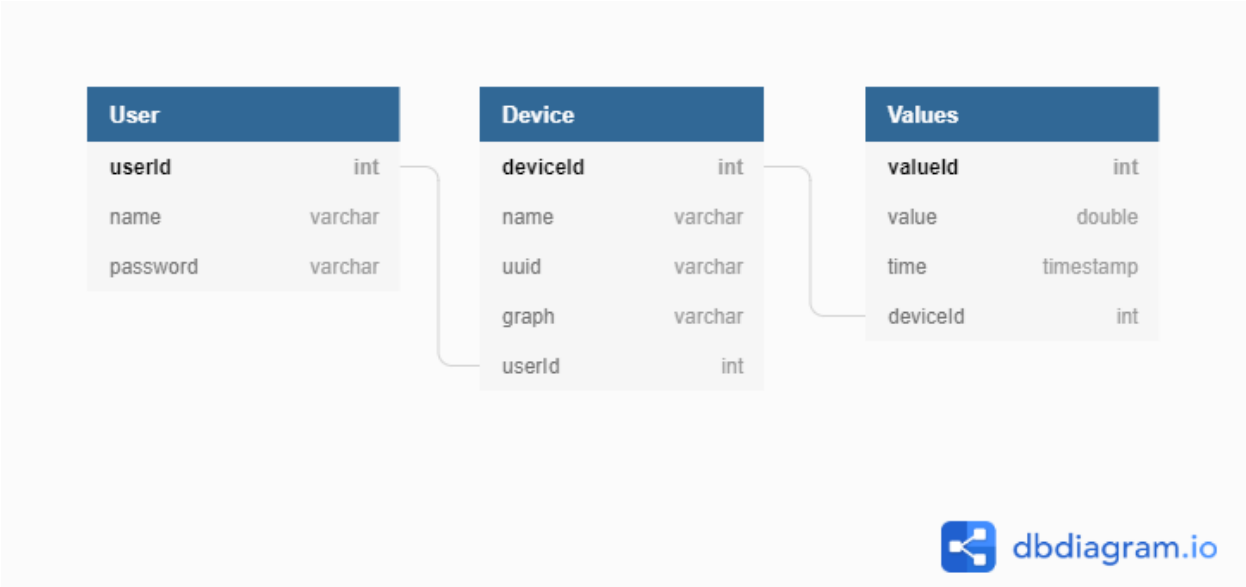
Aktuálny stav stránky:

- Približný rozvrh stránky
- Približný dizajn stránky
- Funkčné prihlasovanie
- Funkčná registrácia účtov
- Rozvrh autorizácie (kam má prístup človek, ktorý nie je prihlásený)

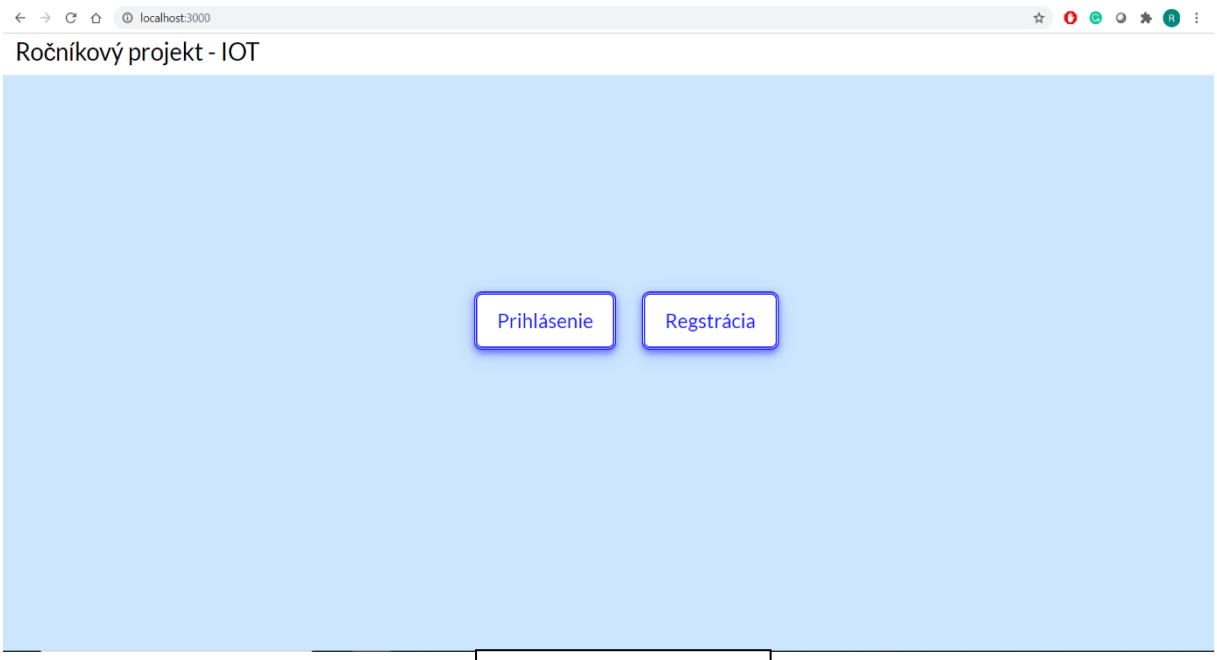
Popis vývoja projektu v letnom semestri:

Na začiatku letného semestra som sa rozhodol zmeniť zameranie webovej stránky. Cieľom novej webovej stránky je spracovanie údajov, ktoré získava zo zariadenia užívateľa, ktorý si zariadenie zaregistroval na stránke. Pre užívateľa umožňuje webová stránka okrem ukladania údajov aj vizualizáciu daných údajov v podobe grafov (momentálne 2 typov grafov). Najprv som pracoval na backende stránky, zvyšný čas som venoval frontendu (grafy) a opravovaniu chýb. Na backend som použil Haskell – Yesod framework, na frontend Elm.

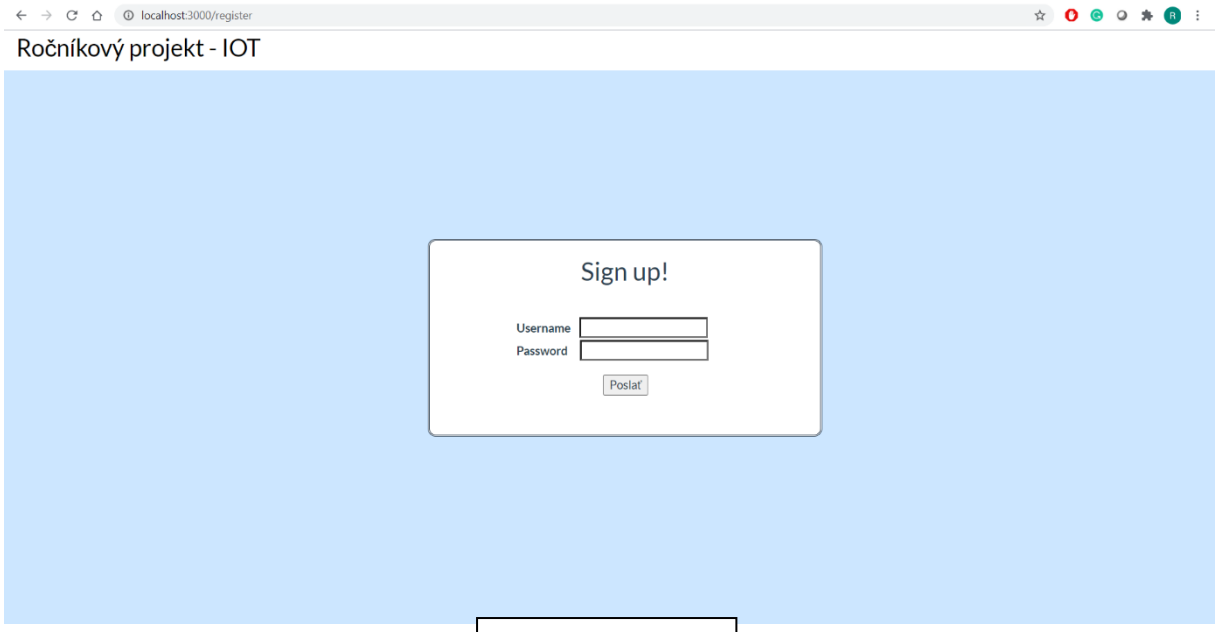
Ako databázu som si vybral Sqlite kvôli jednoduchosti a navrhol som model tabuliek a vzťahy medzi nimi nasledovne:



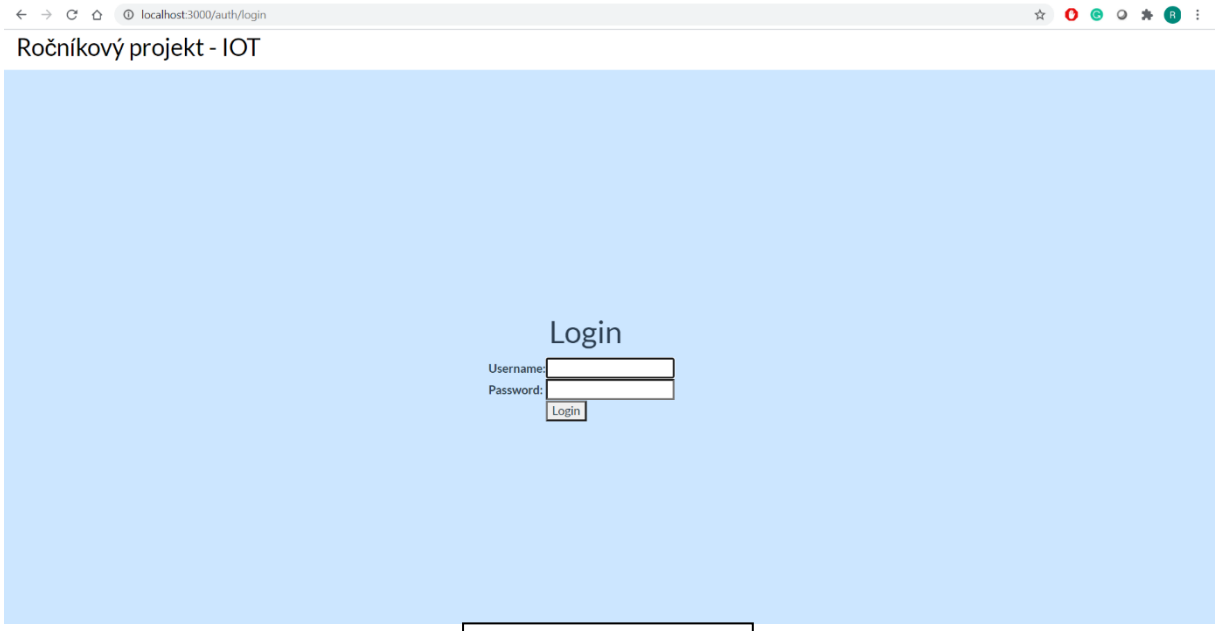
Prvou časťou stránky, ktorú som implementoval, bola registrácia a prihlásenie pre používateľa. Implementoval som pomocou OAuth knižnice, ktorá je súčasťou Yesod frameworku. Heslo používateľa je zahashované v databáze, po prihlásení je používateľ presmerovaný na hlavnú stránku a jeho ID je uložené v Sessione (po 2 hodinách neaktivity sa odhlasí automaticky), ktoré sa využíva na prístup k častiam stránky (prístupným len pre prihlásených používateľov) a na výber zariadení, ktoré patria používateľovi.



Úvodná strana



Registrácia

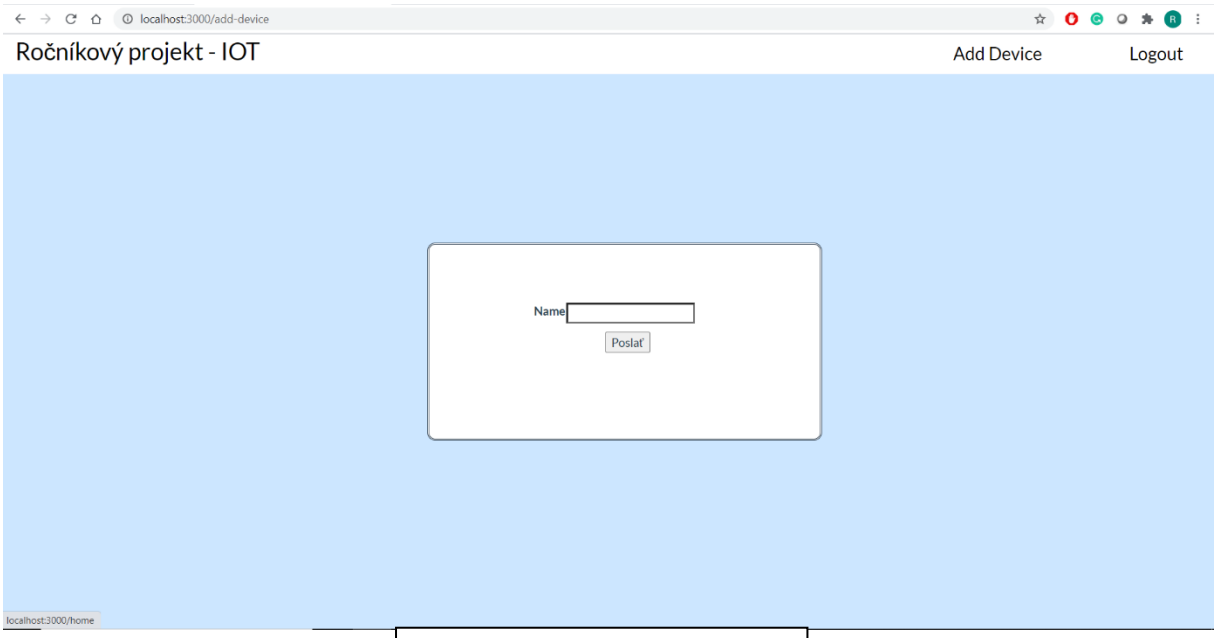


Prihlasovanie

Pokračoval som pridávaním zariadení a komunikáciou backend serveru s frontend klientom. Komunikáciu som zabezpečil pomocou JSON, pričom backend server pre dané zariadenie naformátuje svoje namerané hodnoty do JSONu a frontend klient ho dekóduje na pole dvojíc (dátum, nameraná hodnota).

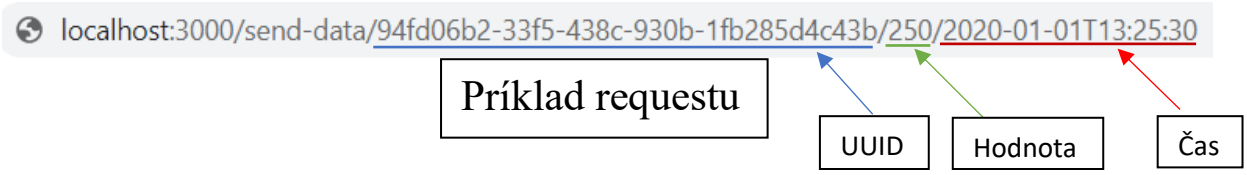
[{"time":"2020-01-01T00:00:00Z","value":100,"deviceId":1},{"time":"2020-08-02T18:35:40.5177826Z","value":200,"deviceId":1},{"time":"2020-08-02T18:36:08.6572016Z","value":100.5,"deviceId":1},{"time":"2020-08-02T18:57:20.3302065Z","value":-50,"deviceId":1}]

Príklad komunikácie



Pridávanie zariadenia

Údaje zo zariadenia sa ukladajú jednoduchým requestom na stránku. V requeste sa musí nachádzať UUID, ktorým sa zariadenie identifikuje, nameraná hodnota a čas (voliteľné), v ktorom nameral danú hodnotu. Ak chce zariadenie uviesť čas, musí mať formát OSI, teda YYYY-MM-DDThh:mm:ss, pričom T je písmeno (príklad 2020-01-01T13:25:30).



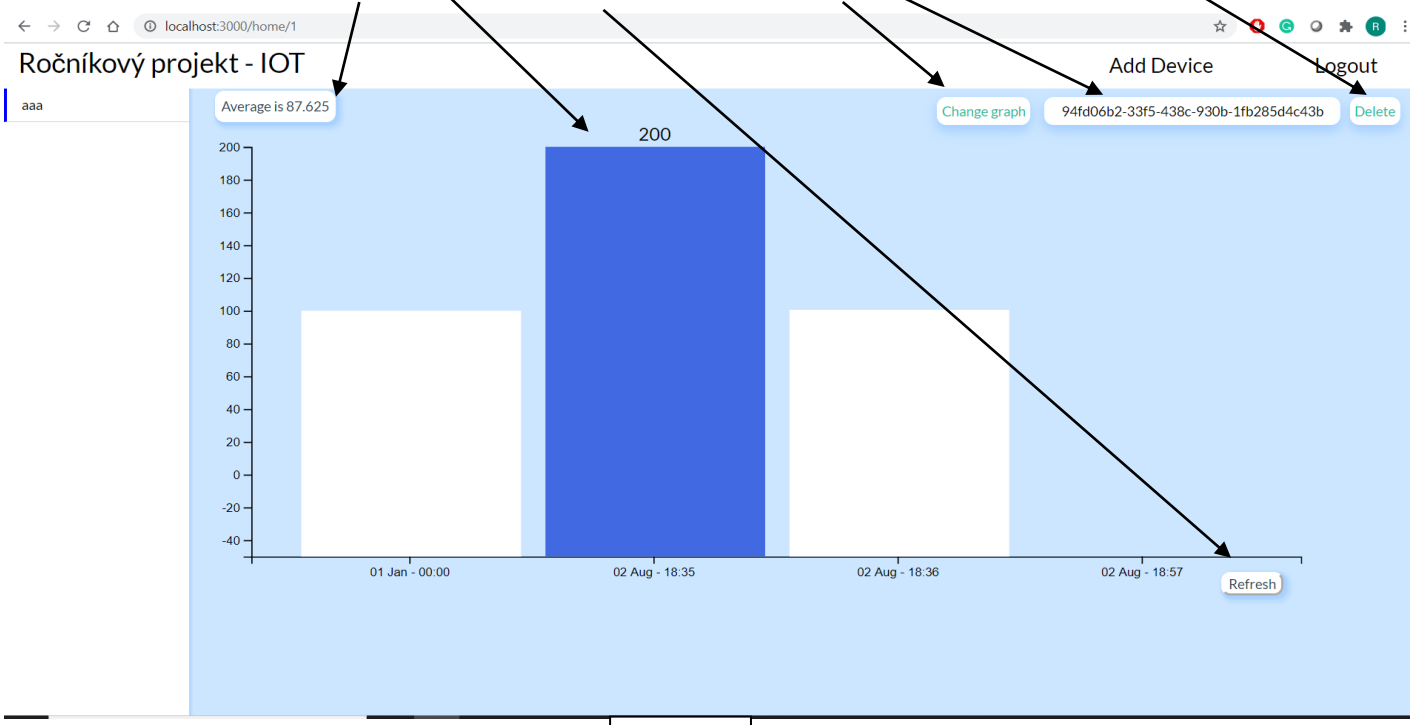
Príklad requestu

UUID

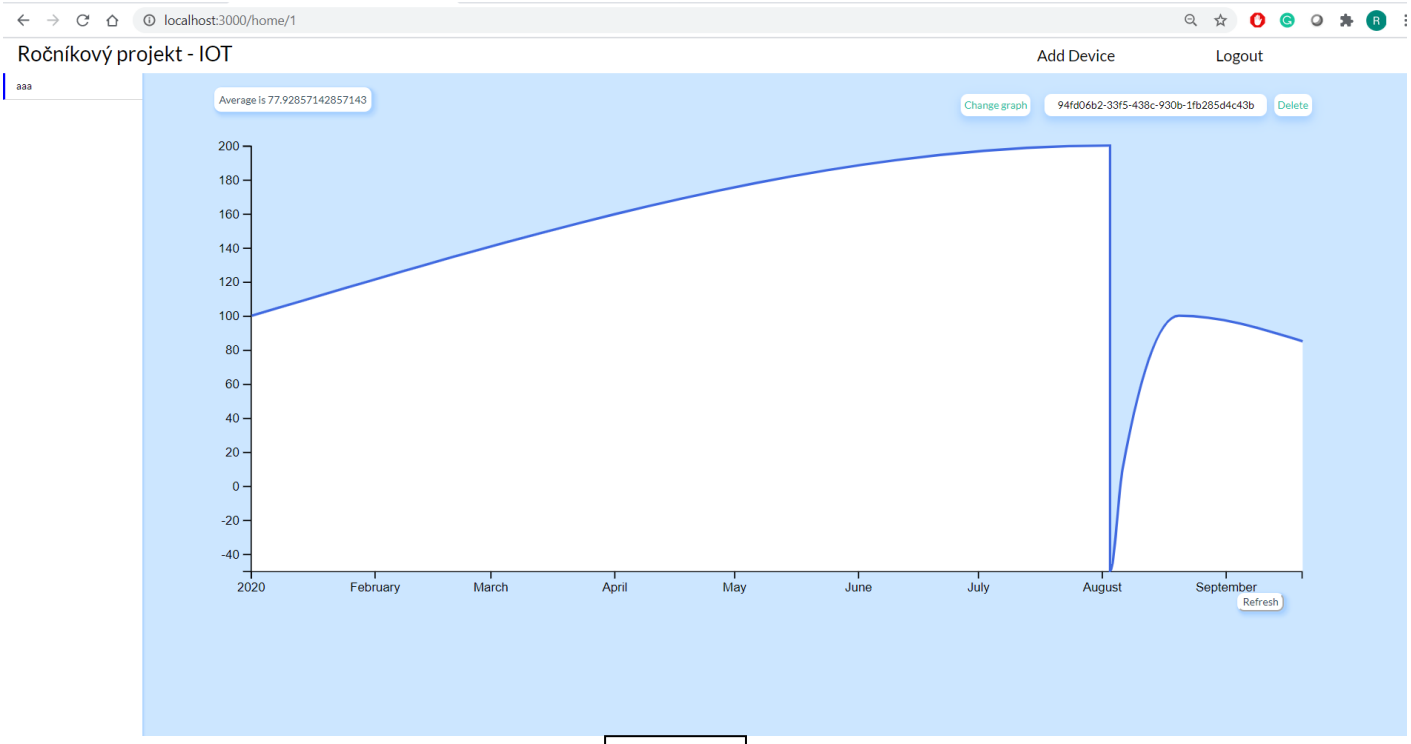
Hodnota

Čas

Po rozkliknutí zariadenia môžete vidieť UUID zariadenia, možnosť vymazať zariadenie, aritmetický priemer hodnôt, graf, možnosť na refresh grafu, možnosť zmeniť typ grafu.



1. graf



2. graf

Porovnanie tvorby webovej stránky pomocou funkcionálneho a imperatívneho programovania:

1. Learning curve – funkcionálne a imperatívne programovanie sa líšia aj formou nutnosti premýšľania programátora. Keďže nepatrí medzi jazyky, ktoré sa bežne vyučujú, musel som sa doučiť sám nové koncepty, syntax a nový spôsob, ako sa pozerať na danú problematiku. Tento proces však trvá oveľa dlhšie ako pri bežných imperatívnych jazykoch.
2. Dokumentácia k frameworkom - Framework Yesod má prehľadnú dokumentáciu, pomocou ktorej som sa vedel ľahko dostať k tvorbe stránky. Avšak čím viac špecifickejších informácií som potreboval, tým menej bola dokumentácia dostačujúca.
3. Komunitné otazky a riešenia - hľadanie odpovedí na moje otázky, ktoré neboli riešené v dokumentácii, bolo málokedy úspešné, pretože odpovede boli zastaralé, alebo chýbali. Toto sa pri imperatívnom jazyku nestáva, keďže väčšina programátorov programuje v ňom.
4. Chybové hlasenia na stránke – veľkou výhodou pri tvorbe stránky bolo, že pri zadaní hodnoty v inom tvare, ako očakávala stránka (napríklad pri requeste), Yesod daný request nespracoval, ale vyhlásil chybu, že daná hodnota je v zlom tvare.
5. Hlásenia chýb pri kompilácii – keď som niekde spravil malú chybu, pri kompilácii som dostal veľmi neprehľadné hlásenie chyby, a teda jediným východiskom bolo skúšať rôzne možnosti riešení. Pokiaľ chce človek efektívne využívať svoj čas, takéto hlásenia chýb ho môžu zastaviť na celé hodiny.