



Semestrální práce z KIV/WEB

Plánování výletů pro skupiny

Martin Brom (A16B0009P)

30. ledna 2018

Obsah

1	Zadání	2
2	Popis aplikace	3
3	Využité technologie	3
4	Adresářová struktura	4
4.1	Adresář app	5
4.2	Adresář core	5
4.3	Adresář public	6
4.4	Adresář resources	7
5	Workflow aplikace	8
6	Uživatelská dokumentace	9
6.1	Nepřihlášený uživatel	9
6.2	Výletník	9
6.3	Organizátor	9
6.4	Majitel	9
6.5	Administrátor	9
6.6	Testování	10
7	Instalace	11
8	Závěr	11

1 Zadání

Nepřihlášený uživatel má přístup k několika úvodním informačním stránkám aplikace, přihlašovacímu a registračnímu formuláři a výletům (read-only), ke kterým zná unikátní veřejnou adresu.

Přihlášený uživatel vidí seznam svých nadcházejících, probíhajících a proběhlých výletů, a může si založit výlet vlastní. Dále může upravit své údaje, např. avatar, veřejné jméno nebo heslo...

Majitel výletu (přihlášený uživatel, který výlet založil) může spravovat uživatele s přístupem k výletu nebo pozvat uživatele (jak registrovaného tak nepřihlášeného přes email). Dále může výlet zveřejnit (vygenerovat veřejnou URL) nebo opět schovat.

Správce výletu, který je jmenován nebo zbaven funkce majitelem výletu nemůže spravovat správce výletu, ale může spravovat uživatele s přístupem. Stejně jako majitel, může manipulovat s výletem mnoha různými způsoby: např. změna pořadí dnů, přidání / rušení / editace dnů i akcí každého dne, správa nahraných souborů výletu...

Uživatel na výletě, unikátní kombinace přihlášeného uživatele a výletu může komentovat a prohlížet výlety, ke kterým má přístup.

Poslední role, administrátor aplikace, má přístup do všech výletů a může spravovat všechny výlety i uživatele v aplikaci.

2 Popis aplikace

TripHatch se skládá ze dvou slov, Trip (výlet) a Hatch (vylíhnout). Návrh aplikace vychází z vlastních potřeb, kdy při delších výletech s většími skupinami lidí nastává chaos jak v itineráři tak potřebných dokumentech. Tato aplikace by měla pomoci tyto problémy udržet v rozumné míře.

Aby byla aplikace kompletně připravená na nasazení do plného provozu, bylo by potřeba dořešit ještě několik funkcí, jako např. formulářovou validaci u klienta, zapamatování přihlášení, větší nabídku customizace jednotlivých akcí nebo branding aplikace.

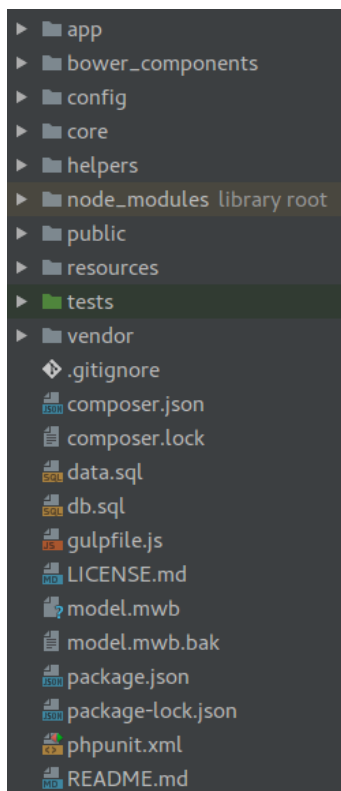
Celý zdrojový kód je dostupný na webové stránce GitHub: [TripHatch](#)

3 Využité technologie

Front-end aplikace je řešen tradičně s využitím HTML5, CSS3 a preprocesoru SASS. Dále je využita knihovna jQuery a templatovací engine Twig. Back-end využívá PHP 7.0, podle zadání bez čistě bez frameworku. O načtení tříd a dependencí projektu se stará composer a npm. Automatickou minifikaci JS a CSS souborů provádí Gulp.

4 Adresářová struktura

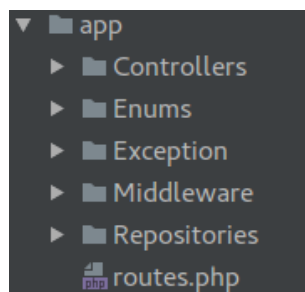
Adresářová struktura je částečně inspirována PHP frameworkem Laravel. Adresář `bower_components` obsahuje Bootstrap a jQuery, adresář `config` konfigurační soubory aplikace, adresář `helpers` soubory s pomocnými funkcemi, adresář `node_modules` balíky instalované pomocí npm, adresář `tests` několik testů aplikace a adresář `vendor` balíky instalované pomocí composeru. Dále se v kořenovém adresáři aplikace nachází soubor pro setup databáze, `setup.sql`, soubory pro instalaci potřebných balíčků aplikace `composer.json` a `package.json`, soubor `gulpfile.js` pro minifikaci a sbalení JS a CSS souborů a `model.mwb`, který představuje model databáze vytvořený v programu MySQL Workbench.



Obrázek 1: Hlavní adresářová struktura

4.1 Adresář app

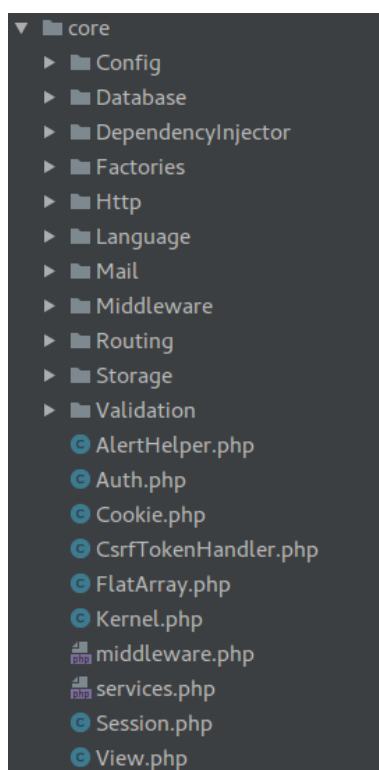
Adresář obsahuje převážně kontrolery, middleware a repozitáře.



Obrázek 2: Adresářová struktura složky app

4.2 Adresář core

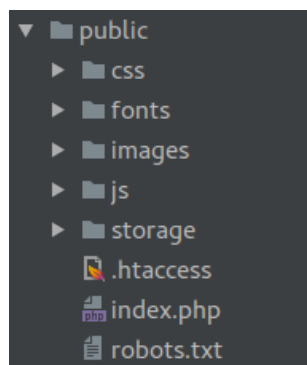
Adresář obsahuje celé jádro mnou vytvořeného "frameworku". Nacházejí se zde jak celé "knihovny", jako např. knihovna pro práci validaci nebo lokalizaci, tak i jednotlivé "base"soubory, jako např. základní kontroler nebo **Response**



Obrázek 3: Adresářová struktura složky core

4.3 Adresář public

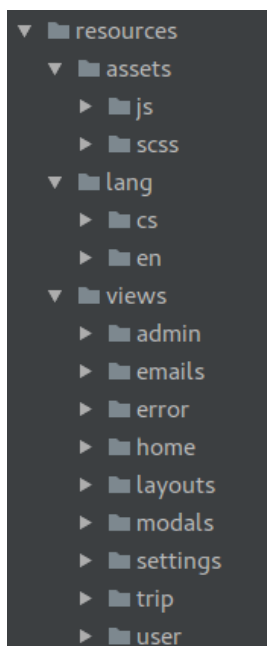
Adresář obsahuje soubory, které využívá prohlížeč pro zobrazení uživateli. Nacházejí se tu minifikované JS a CSS soubory, dále např. soubor `.htaccess` nebo obrázky webu.



Obrázek 4: Adresářová struktura složky public

4.4 Adresář resources

Adresář obsahuje "zdroje"souborů, ze kterých se následně tvoří "zdroje"ve složce public. Nachází se tu jak zdrojové CSS a JS soubory, tak např. šablony pro View nebo soubory lokalizace.



Obrázek 5: Adresářová struktura složky resources

5 Workflow aplikace

Pokusím se vysvětlit jak probíhá celý proces od odeslání požadavku na server až do vrácení odpovědi.

Veškeré URL aplikace jsou odkazovány na soubor `index.php` ve složce `public`. Tam se nastartuje session, třída `DependencyInjector` zajistí načtení všech "services" a dodání závislostí třídám. Poté se z URL, superglobálních proměnných a dalších dat vytvoří třída `Request` a načtou se všechny validní webové cesty aplikace. Tento požadavek je následně předán třídě `Kernel` ke zpracování.

`Kernel` nejprve zkontroluje, jestli se snažíme přistoupit na existující cestu, pokud ne, přesměruje nás na stránku 404. Dále jsou provedeny before-middleware třídy. Tyto třídy zajišťují potřebnou validaci přístupových práv nebo např. platnost CSRF tokenu. Pokud některý z before-middleware neproběhne v pořádku, je vytvořena třída `Response`, která nás s chybovou hláškou přesměruje zpátky odkud jsme požadavek odeslali. Po úspěšném provedení before-middleware je `Request` odeslán na zpracování.

Před vytvořením `Response` zavoláním metody na kontroleru; tyto dva parametry jsou určeny cestou aplikace na kterou se snažíme přistoupit, jsou zvalidovány superglobální proměnné třídou `Validator` s pravidly definovanými také cestou, kterou se snažíme přistoupit.

`Controller` dále manipuluje s modelovou vrstvou aplikace nebo nám jen pouze vrátí odpověď, která nás buď rovnou přesměruje na jinou cestu aplikace, nebo se v případě odpovědi s HTML kódem dostanou ke slovu ještě after-middleware. Ty už nemohou přerušit běh aplikace, pouze pozměňují odpověď vrácenou kontrolerem, např. přidáním validačních zpráv nebo informace o uživateli třídě `View`, která následně HTML obsah odpovědi vyrenderuje.

6 Uživatelská dokumentace

6.1 Nepřihlášený uživatel

Nepřihlášený uživatel má k dispozici pouze úvodní stránku, nebo i k některým výletům (pouze pro čtení) v případě, že mu někdo zaslal unikátní veřejnou adresu výletu. V pravém horním rohu se nachází tlačítko **Přihlásit se**. Na následující stránce je možné se buď zaregistrovat, přihlásit nebo si v případě zapomenutí hesla heslo obnovit.

6.2 Výletník

Výletníkem se rozumí nejnižší role výletu. S touto rolí máte možnost výlet pouze prohlížet, stahovat soubory výletu nebo výlet komentovat. Můžete si ale zároveň vytvořit výlet vlastní kliknutím na tlačítko **Vytvořit nový výlet** v horní části vaší nástěnky. Zde také můžete vidět všechny výlety, které vlastníte nebo k nim pouze patříte. Kliknutím na váš profilový obrázek v pravém horním rohu se dostanete na vlastní profil, kde si můžete změnit některé osobní údaje. Hned vedle se také nachází tlačítko pro odhlášení.

6.3 Organizátor

Organizátor je prostřední role výletu, nejste vlastníkem, ale zároveň můžete výlet téměř bez omezení upravovat. Můžete si povšimnout, že u výletu, které organizujete se nyní objeví ikonka tužky v pravém horním rohu každé akce a dne. Kliknutím na tuto ikonku můžete každou akci nebo den upravit nebo odstranit. Také se vám pod hlavní navigační lištou zobrazí další, světlejší lišta. Ta slouží k administraci výletu. Pomocí ní můžete přidat nový den, nebo např. výlet upravit nebo pozvat někoho nového. Nyní můžete i odstraňovat komentáře nebo přidávat a odstraňovat soubory výletu.

6.4 Majitel

Majitel, nebo vlastník je nejvyšší role výletu. Tuto roli automaticky získáte na vámi vytvořeném výletu, a administraci se už meze nekladou. Kromě práv organizátora můžete nyní výlet skrýt nebo zveřejnit (vytvořit veřejnou adresu výletu). Dále můžete povýšit některého z výletníků na organizátora, nebo mu tuto roli opět odebrat.

6.5 Administrátor

Administrátor je jediná role, která nepodléhá konkrétnímu výletu. Jako administrátor můžete přistupovat k veškerým výletům, jako byste byli jejich majitel. Dále můžete na administrátorské nástěnce pozorovat číselná data o aplikaci a mazat uživatele.

6.6 Testování

Pro testovací účely je v aplikaci několik účtů s emaily od testemail1@test.test do testemail5@test.test. Samozřejmě je možné si vytvořit účet vlastní. Účet s administrátorskými právy je účet testemail1@test.test

7 Instalace

Pro vytvoření databáze aplikace stačí spustit v MySQL soubor `setup.sql`. Poté je nutné nahrát soubory aplikace do kořenového adresáře serveru a spustit následující příkazy, které zajistí

```
composer install  
npm install  
npm run prod
```

8 Závěr

Ačkoliv se mi nepodařilo implementovat nahrávání a zpracování obrázků v aplikaci, myslím si, že jsem vytvořil kvalitní a rozšiřitelný kód. Tato práce mi zásadně pomohla pochopit, jak fungují větší PHP MVC frameworky.