Aplikace METES

Popis architektury

# Úvod

## Účel

Účelem tohoto dokumentu je popsat technickou architekturu aplikace METES.

## Rozsah

Tento dokument pokrývá celou architekturu systému METES. Je však použitelný i pro jiné podobně malé CRUD aplikace.

## Definice a zkratky

METES – Methodology Evaluation System.  
CRUD - Create, Read, Update, Delete - Typ aplikace, kde se převážně vytváří, čtou, upravují a mazají data.  
OOP - Objektově orientované programování.  
ORM - Object-Relational Mapping.  
JSON - JavaScript Object Notation.  
REST - Representational State Transfer  
SPA - Single Page Application

## Odkazy

Bitbucket - <http://bitbucket.org/xmlal00/metes>  
PHP Framework Laravel - <https://laravel.com>  
React - <http://facebook.github.io/react>  
Entrust - <https://github.com/Zizaco/entrust>  
BrowserStack - [https://www.browserstack.com](https://www.browserstack.com/)  
REST API - <https://cs.wikipedia.org/wiki/Representational_State_Transfer>

## Historie verzí

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Verze | Popis | Autor |
| 28. 2. 2016 | 1.0.0 | Sepsání prvotní verze dokumentu | Lukáš Mladý |
| 05.042016 | 1.1.0 | Doplnění informací a grafiky | Lukáš Mladý |
|  |  |  |  |

# Základní vymezení

## Požadavky a omezení

Architektura vychází z požadavků uvedených v dokumentu **Požadavky**.

## Předpoklady

* Pokročilá znalost programovacího jazyka PHP.
* Základy běžných PHP frameworků jako Laravel, Zend Framework, Symfony či Yii.
* Základy unit testování v PHP.
* Základní znalost HTML a CSS.
* Pokročilá znalost JavaScriptu.
* Znalost principů OOP.
* Znalost návrhových vzorů Singleton, Dependency Injection, Observer, Factory.
* Obeznámení s ORM.
* Zkušenost s verzováním softwaru v Gitu nebo SVN.
* Základní znalost databázového systému MySQL.
* Principy práce s virtuálními stroji (VirtualBox).

## Diagram architektury

## laravel-mvc-components.png

# architektonická rozhodnutí a jejich vysvětlení

Aplikace bude mít klasickou třívrstvou architekturu realizovanou ve webovém prostředí.

## Architektonické rozhodnutí a vysvětlení

Backend aplikace bude napsán pomocí jazyka PHP a frameworku Laravel. Rozhraní aplikace (API) bude realizováno pomocí principů REST a data bude primárně vracet ve formátu JSON (s možností pozdějšího přidání jiného formátu výstupu jako např. XML).

Autentifikace bude realizována skrze standardní komponentu Laravelu. Pro autorizaci pomocí oprávnění, rolí a uživatelů do nich příslušících bude použita externí komponenta Entrust.

Díky této architektuře bude pro aplikaci možné vytvořit klienta naprosto nezávisle na daných technologiích (např. mobilní aplikace či aplikace třetí strany).

Důvody:

* PHP je jedním z nejpoužívanějších programovacích jazyků pro web. Členové týmu v něm pasivně či aktivně programují a někteří i absolvovali kurz PHP vyučovaný na VŠE.
* Framework Laravel funguje jako agregátor PHP komponent a spojuje je do frameworku. Obsahuje lokalizace, ORM, Dependency Injection, Unit testy, databázové migrace a seedy, composer pro řešení závislostí komponent a přehlednou dokumentaci.
* REST API - standard při tvorbě webových rozhraní.
* JSON - standardní datový formát v moderních webových aplikacích.

## Architektonické rozhodnutí a vysvětlení

Databázové tabulky a jejich změny budou realizovány na principu migrace a seedových (počátečních) dat. To dovolí celému týmu efektivně řídit a sdílet změny v databázi.

Pro mapování databázových tabulek do PHP bude použit ORM frameworku Laravel s názvem Eloquent.

## Architektonické rozhodnutí a vysvětlení

Zdrojový kód bude sdílen skrze privátní repozitář na veřejné službě Bitbucket a verzován pomocí Gitu. Do repozitáře budou mít přístup všichni členové týmu. Git byl vybrán pro obecnou akceptaci v týmu a jeho popularitě v open source komunitě.

## Architektonické rozhodnutí a vysvětlení

Vývojové prostředí bude realizováno skrze jednotně nastavený virtuální stroj Homestead, který byl vytvořen komunitou okolo PHP frameworku Laravel. Bude tak zajištěna konzistence a jednotnost vývojového prostředí nehledě na primární OS a aplikační vybavení členů týmu.

## Architektonické rozhodnutí a vysvětlení

Frontend aplikace bude napsán pomocí jazyků HTML, CSS a JavaScript. Konkrétněji pak framework Bootstrap pro CSS a React.js knihovna a její další moduly pro JavaScript. Bude se tedy jednat o tlustou klientskou SPA aplikaci, která bude komunikovat s backendem skrze REST API pomocí JSON formátu. Díky tomu lze dosáhnout svižného moderního uživatelského prostředí, které se svou rychlostí a použitelností vyrovná desktopovým aplikacím.

## Architektonické rozhodnutí a vysvětlení

Unit testy backendu aplikace budou programovány zároveň s hlavním vývojem aplikace a to v prostředí frameworku Laravel pomocí PHP Unit.

Integrační testy budou realizovány skrze Selenium.

Testování frontendu aplikace proběhne pomocí unit testů v prostředí Mocha a kompatibilita prohlížečů bude kontrolována v nutných případech pomocí služby BrowserStack.

## Architektonické rozhodnutí a vysvětlení

Aplikace bude primárně psána v českém jazyce. Bude však připravena na lokalizaci pomocí standardní knihovny frameworku Laravel.