

# Tesseract LASAGNA: MVP PWA Framework

## Koncept

**Tesseract LASAGNA** je rychlý, moderní a modulární PHP OOP framework pro rychlé prototypování **Progresivních webových aplikací** (PWA). Tesseract používá *exporty CSV z Tabulek Google* jako datový vstup, vytváří model z vrstev CSV (odtud kódové označení LASAGNA). Abstraktní **Presentery** se používají ke zpracování **Modelu** a exportu výsledných dat ve formátech TEXT, JSON, XML nebo HTML5 (nebo v jakémkoli jiném vlastním formátu). **View** je vytvořen jako sada šablon a *partials* Mustache (Mustache lze také vykreslit v prohlížeči pomocí JavaScriptu). Tesseract je založen na *komponentách Composer*, Model definuje komplexní **RESTful API**, má *rozhraní příkazového řádku* (CLI) a zahrnuje testování *kontinuální integrace* (CI). Tesseract nepoužívá žádné klasické databázové modely a struktury, takže je celkem snadné implementovat všechny typy škálování a integrace. Přístupový model je založen na **šifrovaném cookie klíči**.

## Základy

### Index

Tesseract začíná běh v souboru **www/index.php**, který je cílem pro Apache server prostřednictvím konfiguračního souboru **.htaccess** s využitím *Mod\_rewrite*. **Index** může obsahovat různé definice konstant. **Index** poté načte soubor jádra **Bootstrap.php** z kořenové složky aplikace.

### Bootstrap

**Bootstrap** nastavuje základní konstanty a prostředí aplikace, za běhu se také nastavuje **Nette Debugger**. Bootstrap poté načte soubor jádra **App.php** ze složky **app/** (umístění lze přepsat pomocí konstanty).

### Aplikace

**App** zpracovává konfigurační soubory aplikace (veřejné a soukromé), nastavuje mechanismy ukládání do mezipaměti (volitelná databáze Redis), konfiguruje směrování URL, vydává hlavičky CSP a nastavuje základní **Model** (multidimenzionální pole). **Aplikace** poté načte odpovídající *prezentátor* na základě vyhodnocení URI pomocí routeru. Může také spustit *prezentátor CLI*, pokud je rozpoznáno CLI volání. Když *prezentátor* vrátí aktualizovaný model, výstup je zobrazen a nastaví se koncové hlavičky (včetně některých volitelných informací o ladění). Běh zde končí.

### Presentery

**Presentery** jsou instance podtříd založené na *abstraktní třídě* **APresenter.php** a definují alespoň metodu *process*, která je volána z **App**. Metoda *process* může buď zobrazit výsledná data, nebo data vrátit zapouzdřená uvnitř modelu zpět do **App** k zobrazení.

# Hierarchie souborového systému

- **apache/** - příklad konfigurace Apache
- **app/** - presentery a konfigurace NE-ON
- **bin/** - bash skripty pro Makefile
- **ci/** - protokoly průběžné integrace
- **data/** - soukromá data, šifrovací klíče, importy CSV atd.
- **doc/** - dokumentace vygenerovaná phpDocumentorem
- **docker/** - soubory, které mají být vloženy do Docker kontejneru
- **logy/** - systémové protokoly
- **node\_modules/** - moduly Node.js používané Gulpem
- **temp/** - dočasné soubory, zkompilované šablony Mustache
- **vendor/** - Composerem orchestrované třídy
- **www/** - statické soubory
  - **www/cdn-assets/** - hash verze úložiště odkazuje na www/
  - **www/css/** - CSS styly
  - **www/docs/** - odkaz na doc/
  - **www/downloads/** - soubory ke stažení
  - **www/epub/** - soubory ePub
  - **www/img/** - obrázky
  - **www/js/** - JavaScript soubory
  - **www/partials/** - části šablon Mustache
  - **www/summernote/** - editor Summernote
  - **www/templates/** - šablony Mustache
  - **www/upload/** - soubory nahrané přes administrační panel
  - **www/webfonts** - fonty

## Administrace

### Přihlášení a odhlášení

Přihlášení do Tesseractu je založeno výhradně na klientovi **Google OAuth 2.0**. Když se uživatel přihlásí, vytvoří se speciální šifrované cookie - hlavní klíč - a nastaví se pomocí protokolu HTTPS. Tento soubor cookie je chráněn před manipulací a jeho parametry lze upravovat v administračním panelu nebo vzdáleně prostřednictvím autentizovaných volání API. Tesseract nepoužívá žádnou databázi přihlášených uživatelů. Výchozí adresa URL pro přihlášení je **/login** a výchozí adresa URL pro odhlášení je **/logout**.

## Oprávnění

Tesseract má vestavěné tři základní úrovně oprávnění, které lze snadno rozšířit.

Základní úrovně jsou: 1) **admin** – superuživatel, 2) **editor** – může obnovovat data a upravovat články, 3) **tester** – žádná zvýšená oprávnění, 4) **ověřený uživatel** – práva stejná jako úroveň 3 a 5) **neautentizovaný uživatel** - neznámá identita.

## Základní funkce

### Soubory Sitemap

Tesseract generuje TXT a XML sitemapy na základě routovacích tabulek.

### Záhlaví CSP

V souboru **app/csp.neon** můžete definovat hlavičky pro zásady zabezpečení obsahu.

## Extra funkce

### QR obrázek

Základní routovací adresa je **qr/[s|m|l|x:velikost]/[:trailing]**. Příklad Hello World je následující:

### Čtečka elektronických knih EPUB

TBD

### WYSIWYG články

TBD

### Pingback Monitoring

Podívejte se na demo na této URL: [<https://lasagna.gscloud.cz/pingback>]