

# Programování v PHP

Katedra softwarového inženýrství  
Fakulta informačních technologií  
České vysoké učení technické v Praze

© Pavel Štěpán, Helena Wallenfelsová 2014

**Vytvoření MVC + REST aplikace pomocí frameworku F3 (popis)**  
BI-PHP



## Popis programu, vytvořeného ve stylu MVC pomocí frameworku Fat-Free Framework

### MVC

MVC (Model-View-Controller) představuje strukturu (architekturu) pro navrhování (zejména) web aplikací. Skládá se z **Controlleru** - části, která **přijímá vstupy** od uživatele a na jejich základě volá **Model** (např. pro načítání a úpravy dat z DB). Dale určuje, které **View se zobrazí uživateli** a případně mu předává nějaká data pro zobrazení (obvykle získána na základě vstupu a dat v DB, načtených pomocí Modelu). Controller tedy provádí řízení aplikace. View může volat Model pro získání dat ke zobrazení. **Model sam představuje business logiku aplikace**, která zejména pracuje s daty (ale nemusí to být pouze nadstavba nad DB). V MVC by však Model nikdy neměl volat View nebo Controller.

(Větsí) MVC programy jsou obvykle podstatně lépe udržitelné, testovatelné a snadnější se vyvíjejí než programy, které tuto strukturu nepoužívají.

-----

Programy, které ilustrují použití frameworku, jsou funkčně stejné jako jejich předchozí verze - provádějí tedy zobrazování, vyhledávání a úpravy dat z DB. Jsou však vytvořeny pomocí frameworku **Fat-Free Framework (dále f3)**. Tento Framework je značně rozšířen, je rychlý, malý a přitom poměrně dobře vybavený (je někde na mezi microframeworky a "velkými" frameworky).

Získáte jej na adrese **<http://fatfreeframework.com>**. Ovšem pozor - stažená verze je kompletní aplikace nad tímto frameworkem. **Vlastní framework je uložen v adresáři lib** - ten nakopírujte do své aplikace (např. do podadresáře dané aplikace).

Tento framework je (na rozdíl od jiných) velmi pružný - nevyžaduje "tuhou" strukturu adresáře. Lze nad ním vytvořit různé typy aplikací - například zde strukturu MVC. Tomu je přizpůsobena v těchto příkladech zvolená struktura adresáře:

- **v kořenu aplikace jsou soubory index.php** (startovací stránka aplikace) a nepovinné: **.htaccess** (pro prepis adres), **config.ini** (konfigurační soubor) a pro jednoduchost je zde i databázový soubor **northwind.sqlite** (praktická aplikace by se připojovala ke "skutečné" databázi - např. MySQL).
- **podadresáře jsou lib** (vlastní Framework f3), **views**, a **models**. Pomocné jsou **tmp** ("přeložená" views), **css** a **classes**, kam ukládáme pomocné třídy. Bylo by možné mít i adresář **controllers**, ale zde je řízení programu natolik jednoduché, že je celé realizováno v souboru **index.php**

Pokud chceme "user and SEO friendly" adresy, je možné dát do kořenu aplikace soubor **.htaccess** pro server Apache (v serveru Apache je třeba mít povoleno prepisování adres)

Na závěr: tato ukázka používá framework f3, protože se s ním dobře pracuje a jeho "instalace" je snadná, takže je vhodný pro výuku i pro praktické používání. Bylo by ovšem možné použít mnoho jiných frameworků ke stejnému účelu. Rozhodně tento příklad nemá sloužit k přesvědčování, že f3 je "jediný a nejlepší" framework - to určíte ne! Volba frameworku je (kromě objektivních hledisek) z velké části záležitostí individuální - jak komu který framework "sedí"! A mnohdy je lepší nepoužít žádný - MVC lze vytvořit i v čistém PHP.

## REST

Tento framework však umožňuje nejenom vytvářet MVC aplikace, ale jde i o tzv. RESTful (Representational State Transfer) framework. REST technologie chape web jako sadu zdrojů (resources) - ne nutně webových stránek, kde každý zdroj je identifikován svým URI. REST definuje pro tyto zdroje čtyři základní operace - POST, GET, PUT a DELETE (odpovídají CRUD - CREATE, READ (resp. RETRIEVE), UPDATE a DELETE).

V praxi ovšem webový software často nepodporuje operace PUT a DELETE, proto bývají realizovány pomocí zbývajících dvou a předávaných dat. Data (parametry) těchto operací se obvykle předávají jako další část adresy (URI).

Například: `http://....move/first` může znamenat požadavek na zobrazení prvního zaměstnance.

Operace GET je vyvolána při kliknutí na hyperlink nebo odesláním submitem z formu při nastavení atributu method ve formu na GET.

Právě na základě REST je v f3 prováděno celé řízení toku programu (controllers), což je vidět ve stránce `index.php`. Zde je pro každou adresu (a její parametry) definována funkce, která bude při přijetí takovéto adresy zavolána v závislosti na použité metodě PUT, GET.

Například: v metodě `route` našeho frameworku je vždy prvním parametrem použita metoda a zdroj (`'GET /move/@direction'`), druhým parametrem je funkce příslušná tomuto požadavku.

### Varianty programu:

Bylo naší snahou ukázat různé varianty daného programu pro ilustraci vývoje MVC a REST aplikace s použitím i bez použití frameworku. Jde pochopitelně o podstatně větší rozsah, než je možno probrat na cvičení. Proto jsme přiložili i trochu rozsáhlejší verze, ty jsou ale ve formě .zip souboru (z technických důvodů jsou uváděny jako .pdf - před použitím je přejmenujte).

- „kompaktní“ verze - celý program je v jednom souboru
- verze MVC, vytvořena bez použití frameworku (.zip)
- velmi zjednodušená verze, která slouží pro seznámení se základy frameworku f3
- plná verze vytvořena pomocí f3, která pro přístup k DB používá standardní PDO (.zip)
- další plná verze vytvořena pomocí f3, která přistupuje k DB pomocí ORM (Object Relational Mapping), zabudovaného v f3

Obe plné verze odpovídají předchozí verzi MVC programu, vytvořeného bez použití frameworku i verzi „kompaktní“.

Varianty, kde je přímo uveden kód jako PDF soubory (ne-zip), jsou pro zprovoznění doplněny databází Northwind.sqlite (uložena v předchozích materiálech), kterou umístíte do kořene aplikace. A samozřejmě tam umístíte i vlastní framework f3 - adresář lib (ze souboru, staženého ze stránek f3). Pro verzi s ORM je možno přidat ještě podadresář css, který obsahuje formátovací soubor (je obsazen v předchozím .zip).