

Programování v PHP

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

© Pavel Štěpán, Helena Wallenfelsová, 2014

Přechod k MVC pomocí frameworku F3 + ORM (controller)
BI-PHP



Soubor index.php

```
<?php
// Ukazka pouziti Fat Free Framework + jeho ORM
// (spise Active Record) - mappers
// Pouziva se database SQLite (Northwind)
// Cely program je v tomto souboru + views (zkraceni pouzitim ORM)
// Nejsou zde osetreny situace, pri kterych by nekolik uzivatelu
// najednou menilo data!!

// inicializace

$f3=require('lib/base.php');// lib obsahuje F3 (Fat Free Framework)

$f3->set('DEBUG',1);           // logovani files & lines
                              // (0 - potlaceni logu)

$f3->set('UI','views/');       // nastaveni cesty k views apod.

// $f3->config('config.ini'); // nastavuje config file - napr.
                              // pro externi urceni path k views
                              // zde nepouzito

// path pro autoload - zde nepouzito
// $f3->set("AUTOLOAD","classes/; models/");

// connection pro MySQL server (mozno pouzit i pole atributu)
// $conn = new
//         DB\SQL('mysql:host=localhost;port=3306;dbname=Northwind',
//                 'root','heslo');

// connection pro db SQLite
$conn = new DB\SQL('sqlite:northwind.sqlite');

// ORM (spis Active record) - na zaklade tabulky (view)
// vytvori objekt (prazdny - dry)

$zamObj = new DB\SQL\Mapper($conn,'EmpView');

// view je pouzito proto, aby se nenacitaly zbytecne sloupce
// (view vytvoreno predem v prilozene DB Northwind)

// takto by se nacetla data z DB do objektu - hydrate
// $zamObj->load(['userID=?',3]);
```

```

// controller + pristupy k DB pomoci ORM (natolik zjednoduseno,
// ze model neni oddelen do samostatného souboru)

// prvotni zobrazeni (pro adresu / )
$f3->route('GET /',
    function($f3){
        global $zamObj; // $zamObj - objekt, vytvoreny ORM z DB

        // inicializace
        // nacteni 1. zam (setrideno podle EmployeeID) z DB
        // nacteni kvuli 1. EmployeeID

        // vim, ze vsechna ID > 0 - autoincrement
        $zamObj->load(['EmployeeID > 0'], // load - nacteni z DB
            // podminky pro dotaz do DB:
            ['order'=>'EmployeeID','offset'=>0,'limit'=>1]);

        // nastaveni (a vytvoreni) globalnich promennych F3
        // frameworku, ktere budou pouzity v prislusnem view:
        $f3->set("aktualniID", $zamObj->EmployeeID);

        $f3->set("errMsg", "");
        $f3->set("okMsg", "");
        $f3->set("hledane", "");
        $osloveni = $f3->set("osloveni", "All");

        // nacteni vseh zamestnancu z DB
        // (setrideno podle EmployeeID)
        // find s prazdnym polem - prazdne where
        $zamestnanci = $zamObj->find([],["EmployeeID"=>1]);
        $f3->set("zamestnanci",$zamestnanci);

        // zpracovani view (template), vytvoreneho s
        // pouzitim template language
        echo Template::instance()->render('viewDisplay.htm');
    }
);

// ukonceni aplikace, zobrazeni zaverecne stranky
$f3->route('GET /end',
    function($f3) {
        // zpracovani view , vytvoreneho v cistem PHP
        echo View::instance()->render('viewEnd.php');
    }
);

// zobrazeni a filtrovani dat - prechod na 1. stranku
// pomoci GET (hyperlinku)
$f3->route('GET /display/@selOsloveni/@aktualniID',
    function($f3) {
        global $zamObj;
        // nacteni vstupnich dat (PARAMS)
        $osloveni = $f3->get("PARAMS.selOsloveni");
        $aktualniID = $f3->get("PARAMS.aktualniID");
    }
);

```

```

        // naplneni pole zamestnanci dle osloveni z DB
        // (setrideno podle EmployeeID)
        $cond = ($osloveni == "All")?[]: // All - zadna podm.
            // podminka pro find; pripojena tecka,
            // odstranena kvûli pouziti ve web URI
            ['TitleOfCourtesy=?',$osloveni."."];
        $zamestnanci = $zamObj->find($cond, ["EmployeeID">1]);

        $f3->set('zamestnanci',$zamestnanci);
        $f3->set("osloveni",$osloveni);
        $f3->set("aktualniID",$aktualniID);

        echo Template::instance()->render('viewDisplay.htm');
    }
);

// zobrazeni a filtrovani dat - prechod na 1. stranku
// pomoci POST (form + submit)
$f3->route('POST /display',
    function($f3) {
        global $zamObj;

        // lze psat i jako POST["selOsloveni"] - mapuje
        // standardni $_POST
        $osloveni = $f3->get("POST.selOsloveni");
        $aktualniID = $f3->get("POST.aktualniID");

        // naplneni pole zamestnanci dle osloveni z DB
        // (setrideno podle EmployeeID)
        $cond = ($osloveni == "All")?[]:
            ['TitleOfCourtesy=?',$osloveni."."];
        $zamestnanci = $zamObj->find($cond, ["EmployeeID">1]);

        $f3->set('zamestnanci',$zamestnanci);
        $f3->set("osloveni",$osloveni);
        $f3->set("aktualniID",$aktualniID);

        echo Template::instance()->render('viewDisplay.htm');
    }
);

// prechod na straku editace (pomoci hyperlinku)
$f3->route('GET /edit/@osloveni/@aktualniID',
    function($f3){
        global $zamObj;
        $selOsloveni = $f3->get("PARAMS.osloveni");
        $aktualniID = $f3->get("PARAMS.aktualniID");

        // nacteni aktualniho zamestnance
        $zamObj->load(['EmployeeID=?',$aktualniID]);
        renderEditView($f3,$zamObj,'','','',
            false,$selOsloveni,$aktualniID);
    }
);

```

```

// posuny po datech (cast adresy @direction v routingu = first,
// prev, next, last (hodnoty hyperlinku ve views)
$f3->route('GET /move/@direction/@selOsloveni/@aktualniID',
    function($f3){
        global $zamObj;
        // neosetruji chybné hodnoty direction
        $errMsg = "";
        $saveID = true; // uložit EmployeeID
        $orderLimit = " DESC"; // descending order
        $orderNeighbour = "";
        $operatorNeighbour = " > ?";
        $errNeighbour = "Jsme na posledním radku";
        $direction = $f3->get('PARAMS.direction');
        $osloveni = $f3->get('PARAMS.selOsloveni');
        $aktualniID = $f3->get('PARAMS.aktualniID');

        // @direction: first, prev, next, last
        switch ($direction){
            case "first":
                $orderLimit = ""; // ascending order
            case "last":
                // netestuji empty($zam) - předpokladám
                // neprázdnou tabulku
                // načtení prvního/posledního zaměstnance
                // z tabulky DB
                // (serazeno dle EmployeeID);
                // $orderLimit = " DESC" nebo ""
                $zamObj->load(null,
                    ['order'=>'EmployeeID'.$orderLimit,
                     'offset'=>0, 'limit'=>1]);
                break;
            case "prev":
                $orderNeighbour = " DESC";
                $operatorNeighbour = " < ?";
                $errNeighbour = "Jsme na prvním radku";
            case "next":
                // načtení předchozího/následujícího
                // zaměstnance z DB
                // $operatorNeighbor = " > ?" nebo " < ?",
                // $orderNeighbor = " DESC" nebo ""
                $zamObj->load(['EmployeeID'.$operatorNeighbour,
                             $aktualniID],
                    ['order'=>'EmployeeID'.$orderNeighbour,
                     'offset'=>0, 'limit'=>1]);
                // neexistuje další record - jsme na začátku
                if ($zamObj->dry()){
                    // Jsme na prvním/posledním radku
                    $errMsg = $errNeighbour;
                    // ponechat aktuálního Zam z DB
                    $zamObj->load(['EmployeeID = ?',
                                 $aktualniID]);
                    $saveID = false; // neukladat - nastaveno OK
                }
            }
        }
    }

```

```

        renderEditView($f3,$zamObj,'','',$errMsg,
                        $saveID,$osloveni,$aktualniID);
    }
);

// zmeny dat
$f3->route('GET /clear/@selOsloveni/@aktualniID',
function($f3){
    global $zamObj;

    $osloveni = $f3->get('PARAMS.selOsloveni');
    $aktualniID = $f3->get('PARAMS.aktualniID');
    $zamObj->LastName = $zamObj->FirstName =
    $zamObj->TitleOfCourtesy = $zamObj->City = "";
    renderEditView($f3,$zamObj,'','',$errMsg,
                    false,$osloveni,$aktualniID);
}

);

$f3->route('POST /edit', // find, update, insert
function($f3){
    global $zamObj,$conn;

    $hledane = $okMsg = $errMsg = '';
    $saveID = false; // neukladat EmployeeID

    $aktualniID = $f3->get("POST.aktualniID");
    $osloveni = $f3->get('POST.selOsloveni');

    // hledani
    // v praxi otestovat!!
    if ($f3->exists("POST.btnFind")){
        $hledane = $f3->get("POST.txtHledanePrijmeni");
        $zamObj->load(['LastName = ?',$hledane],
            // podminky
            ['order'=>'EmployeeID','offset'=>0,'limit'=>1]);

        if ($zamObj->dry()){ // jmeno nebylo nalezeno
            $errMsg = "Nebylo nalezeno hledane prijmeni";
            $zamObj->load(['EmployeeID = ?',$aktualniID]);
        }
        else {
            $saveID = true; // uložit EmployeeID
            $okMsg = "Hledany radek nalezen";
        }
    }
    // update
    elseif ($f3->exists("POST.btnUpdate")) {
        // symbolicky test vstupu
        $formID = $f3->get("POST.txtID");
        if (empty($formID) || !ctype_digit($formID)){
            $errMsg =
                "Nelze upravit - neni zadano platne ID";
        }
    }
}

```

```

        // nastavim puvodni hodnoty, které zobrazí form
        // EmployeeID nenastavuji - není platné
        // (zůstane prázdné)
        fillObjectZam($f3,$zamObj);
    }
    else {
        // nemohu použít přímo objekt generovaný z view
        // - v SQLite je view needitovatelné!!
        // Proto nový objekt z tabulky:
        $empObj = new DB\SQL\Mapper($conn,'Employees');
        // pro update nutně nacíst - jinak při save
        // provádí insert!!
        $empObj->load(['EmployeeID=?',$aktualniID]);

        // hodnoty z formu vložím do objektu - v praxi
        // otestovat!!
        // správné nastavení EmployeeID - kvůli
        // hackerům (stejně ale ošetří F3)
        fillObjectZam($f3,$empObj);
        $empObj->EmployeeID = (int)$aktualniID;
        $empObj->save(); // uložení do DB (update)

        $okMsg = "Změny byly provedeny";
        // objekt pro zobrazení ve formu
        $zamObj->load(['EmployeeID=?',$aktualniID]);
    }
}
// insert
elseif ($f3->exists("POST.btnInsert")) {
    // Pro insert nutně použít nový (nenacítaný) objekt
    // (jinak bychom při save prováděli update!!).
    // Nemohu použít přímo objekt generovaný z view.
    // Pro SQLite je view needitovatelné!! Proto nový
    // objekt z tabulky:

    $empObj = new DB\SQL\Mapper($conn,'Employees');
    // hodnoty z formu vložím do objektu - otestovat!!
    fillObjectZam($f3,$empObj);

    $empObj->save(); // vlastní uložení do DB (insert)
    // přidělené ID (autoincrement) - OK i pro pro SQL
    $prideleneID = $empObj->get('_id');
    $okMsg = "Vložení bylo provedeno";

    // objekt pro zobrazení ve formu } // load - nacíst
    $zamObj->load(['EmployeeID=?',$prideleneID]);
    $saveID = true; // uložit EmployeeID
}

renderEditView($f3,$zamObj,$hledane,$okMsg,$errMsg,
                $saveID,$osloveni,$aktualniID);
}
);

```

```

// vymaz recordu
$f3->route('GET /delete/@selOsloveni/@aktualniID',
    function($f3){
        global $zamObj,$conn;

        // zde jiz netestuji spravnost ID (vymazana apod.)
        $aktualniID = $f3->get('PARAMS.aktualniID');
        $osloveni = $f3->get('PARAMS.selOsloveni');

        // novy objekt z tabulky kvuli Read Only view v SQLite
        $empObj = new DB\SQL\Mapper($conn,'Employees');
        $empObj->load(['EmployeeID=?',$aktualniID]);
        $empObj->erase(); // vlastni vymazani z DB

        $okMsg = "Vymaz byl proveden";
        $errMsg = "";

        // zjistení radku, který se má zobrazit místo smazaného
        $zamObj->load(['EmployeeID > ?',$aktualniID],
            // nacíst dalšího řádek z DB
            ['order'=>'EmployeeID','offset'=>0,'limit'=>1]);

        // neexistuje další record - vymazali jsme poslední?
        if ($zamObj->dry()){
            $zamObj->load(['EmployeeID < ?',$aktualniID],
                // nacíst předchozí
                ['order'=>'EmployeeID DESC','offset'=>0,'limit'=>1]);
        }

        // není ošetřeno vymazání jediného řádku v tabulce

        renderEditView($f3,$zamObj,'',$okMsg,'',
            true,$osloveni,$aktualniID);
    }
);

// pomocná funkce (nastavení parametru, renderování view)
function renderEditView($f3,$zam,$hledane,$okMsg,$errMsg,
    $saveID,$osloveni,$aktualniID){
    $f3->set('zam',$zam);
    $f3->set('hledane',$hledane);
    $f3->set('okMsg',$okMsg);
    $f3->set('errMsg',$errMsg);
    $f3->set('selOsloveni',$osloveni);
    $f3->set('aktualniID',$aktualniID);
    // uložení ID zvoleného řádku:
    if ($saveID) $f3->set("aktualniID",$zam->EmployeeID);

    echo View::instance()->render('viewEdit.php');
}

```



```
// pomocna funkce (naplneni objektu Zam z poli formulare)
// Pole POST mapuje $_POST a lze pouzit jako objekt
function fillObjectZam($f3,$zam){
    // hodnoty v praxi otestovat!!
    // $zam->EmployeeID = $f3->get("POST.txtID");
    $zam->LastName = $f3->get("POST.txtPrijmeni");
    $zam->FirstName = $f3->get("POST.txtKrestni");
    $zam->TitleOfCourtesy = $f3->get("POST.txtOsloveni");
    $zam->City = $f3->get("POST.txtMesto");
}
```

```
$f3->run(); // spusteni frameworku
```

?>