

# Úvod do praktického programování v Nette

David Grudl



FRAMEWORK

# Motivace

- rok 2004 – žádný dostatečný framework
- nesmí být závislé na JavaScriptu
  - přístupnost, SEO
- nesmí používat PostBack
  - metoda „minimální reprezentace“
- komponenty vs. MVC?

# Vlastnosti frameworku

- Komponenty
- Událostmi řízený model
- „MVC“
- Vrstva URL adres
- Podpora jmenných prostorů
- Samostatně použitelné jednotky
- Persistence parametrů

# Jmenné prostory

- novinka PHP 5.3
- první framework s podporou namespaces?

```
namespace Nette::Web;
```

```
require_once dirname(__FILE__) . '/../Object.php';
```

```
/**
```

```
 * HTML helper.
```

```
*/
```

```
class Html extends Nette::Object implements ::ArrayAccess, ::Countable, ::IteratorAggregate
```

```
{
```

```
    /** @var string element's name */
```

```
    private $name;
```

```
    /** @var bool is element empty? */
```

```
    private $isEmpty;
```

```
    /** @var array element's attributes */
```

```
    public $attrs = array();
```

# Jmenné prostory

- Termín PHP 5.3 odsunut: Q2/Q3 2008
- Nette běží v kompatibilním režimu

```
/*namespace Nette::Web;*/

require_once dirname(__FILE__) . '/../Object.php';

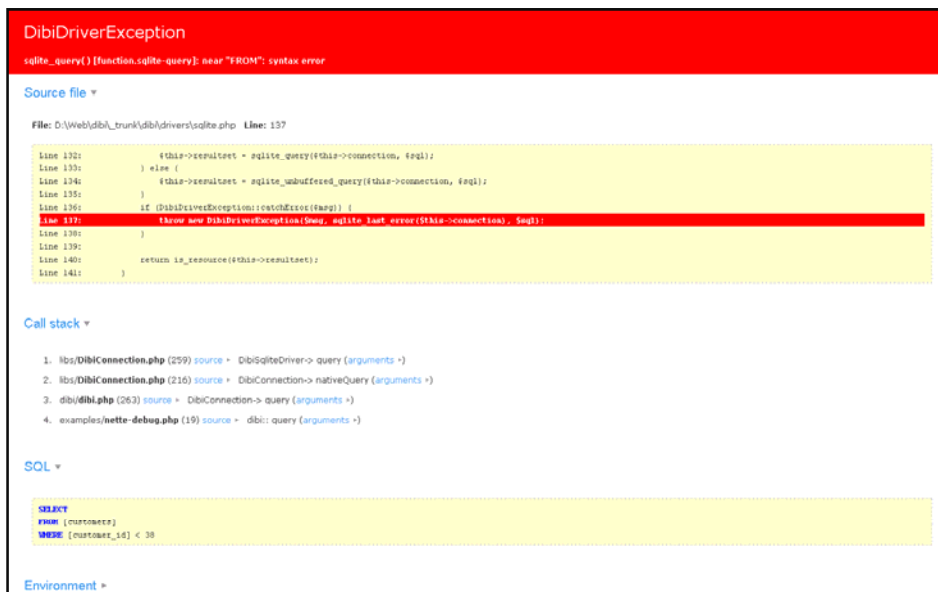
/**
 * HTML helper.
 */
class Html extends /*Nette::*Object implements */i>ArrayAccess, Countable, IteratorAggregate
{
    /** @var string element's name */
    private $name;

    /** @var bool is element empty? */
    private $isEmpty;

    /** @var array element's attributes */
    public $attrs = array();
}
```

# Nette::Debug

- Aby chybové zprávy potěšily a oku lahodily
- Lze přidávat vlastní panely (IDebuggable)
- Zobrazit / Uložit / Odeslat emailem



The screenshot shows a detailed error report for a `DibiDriverException`. At the top, the error message is displayed: `sqlite_query() [function.sqlite_query]: near 'FROM': syntax error`. Below this, the 'Source file' section shows the code context in `D:\Web\diba_trunk\dibi\drivers\sqlite.php` at line 137. The code snippet includes a conditional check for `DibiDriverException::catchErrors($arg)`, and line 137 is highlighted in red, showing the error being thrown: `throw new DibiDriverException($msg, $sqlite->last_error($this->connection), $arg);`. The 'Call stack' section lists four frames, with the top one being `libs/DibiConnection.php (259) source - DibiSqliteDriver->query (arguments ->)`. The 'SQL' section shows the executed query: `SELECT FROM [customers] WHERE [customers_id] < 38`. The 'Environment' section is partially visible at the bottom.

```
DibiDriverException
sqlite_query() [function.sqlite_query]: near 'FROM': syntax error

Source file ▾
File: D:\Web\diba_trunk\dibi\drivers\sqlite.php Line: 137
Line 132:         $this->resultSet = $sqlite_query($this->connection, $sql);
Line 133:     } else {
Line 134:         $this->resultSet = $sqlite_unbuffered_query($this->connection, $sql);
Line 135:     }
Line 136:     if (DibiDriverException::catchErrors($arg)) {
Line 137:         throw new DibiDriverException($msg, $sqlite->last_error($this->connection), $arg);
Line 138:     }
Line 139: }
Line 140:     return $resource($this->resultSet);
Line 141: }
```

```
Call stack ▾
1. libs/DibiConnection.php (259) source - DibiSqliteDriver->query (arguments ->)
2. libs/DibiConnection.php (216) source - DibiConnection->nativeQuery (arguments ->)
3. dibi/dibi.php (263) source - DibiConnection->query (arguments ->)
4. examples/nette-debug.php (19) source - dibi::query (arguments ->)
```

```
SQL ▾
SELECT
FROM [customers]
WHERE [customers_id] < 38
```

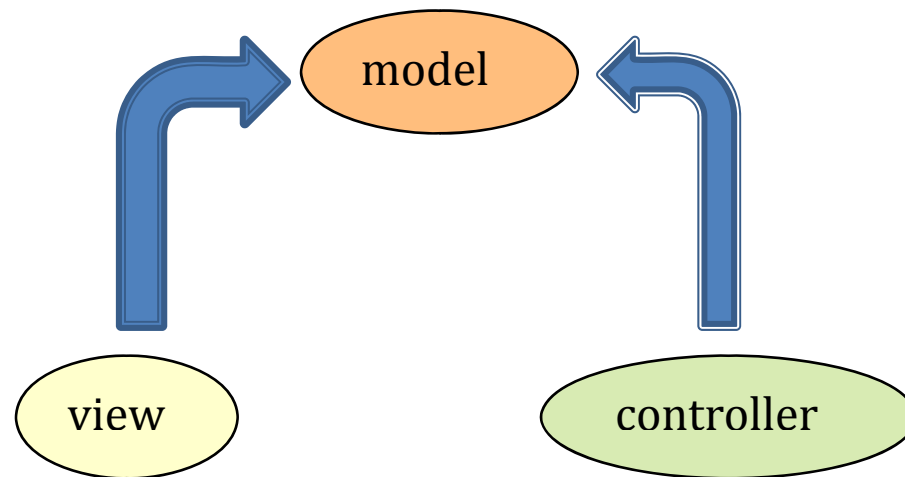
```
Environment ▾
```

# Nette::Collections

- rozhraní
  - [ICollection](#)
    - [IList](#)
    - [IMap](#)
    - [Iset](#)
- výchozí implementace
  - [Nette::Object](#)
    - [Collection](#) (implements ICollection)
      - [ArrayList](#) (implements IList) (Nette::Application::MultiRouter)
      - [Hashtable](#) (implements IMap) (Nette::Config)
      - [Set](#) (implements ISet)

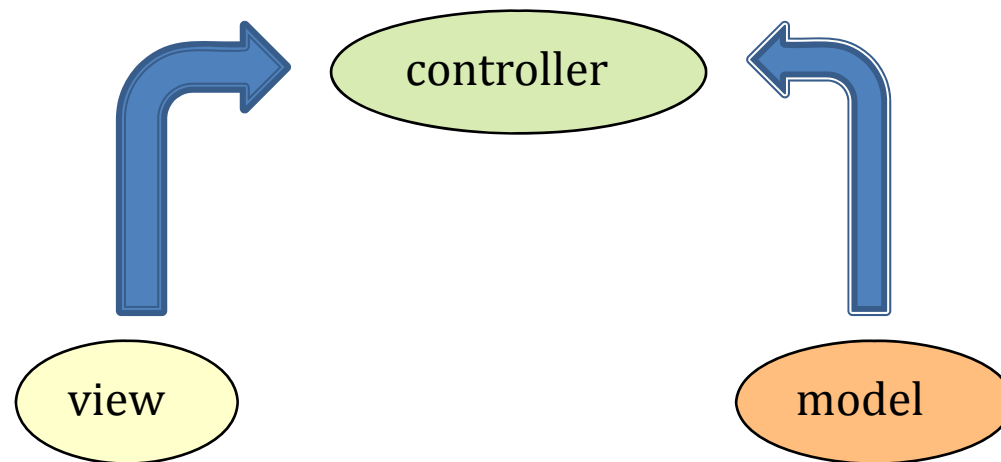
# MVC

- z konce 70. let minulého století (MVC i já)
- V 80. letech překonán



# Jakože-MVC

- Model → databáze
- View → šablona
- Controller → kanál mezi databází a šablonou

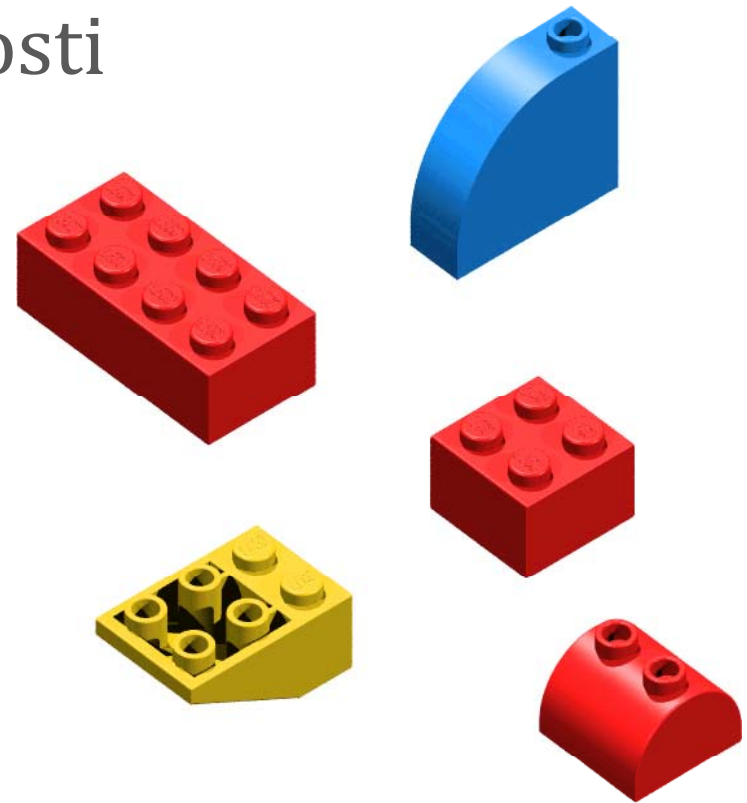


# Nový název pro „Controller“

- Příliš mnoho con...
  - Control
  - Controller
  - Front Controller
  - Container, Component, Config, Collection, Cornflakes, ...
- „Page“ – hezké, leč nepřesné
- „Presenter“ – od tvůrců MVP

# Komponenty

- Pevně dané rozhraní (IComponent)
- Jednotky znovupoužitelnosti
- Představte si kostky Lega

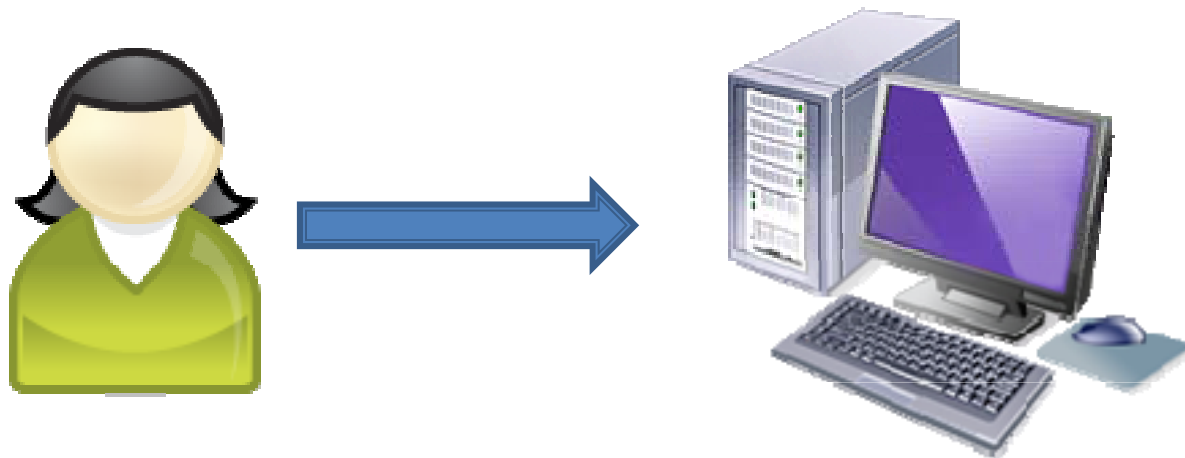


# Základní pojmy

- Component (komponenta)
- Service (služba)
  - dependency injection
  - singleton pattern
- Presenter
  - komponenta IComponent
  - kontejner IContainer

# Komunikace s webovou aplikací

## 1. prohlížení



# Komunikace s webovou aplikací

1. prohlížení
2. změny



# Životní cyklus

- Aplikace (FrontController)
  - routování → PresenterRequest
  - presenter factory → objekt Presenter
- Presenter
  - inicializace
  - exekuce (nebojte)
  - renderování

# Start aplikace – index.php

```
<?php
```

```
// absolute filesystem path to the web root  
define('WWW_DIR', dirname(__FILE__) . '/');
```

```
// absolute filesystem path to the application root  
define('APP_DIR', WWW_DIR . '../');
```

```
// load bootstrap file  
require APP_DIR . 'bootstrap.php';
```

# Start aplikace – bootstrap.php

```
require_once 'Nette/loader.php';

/**
 * Configure PHP
 */
Environment::loadConfig();

/**
 * Setup Nette Application
 */
$application = Environment::getApplication();
$router = $application->getRouter();

$router[] = new Route('<presenter>/<view>/<id>', array(
    'presenter' => 'admin',
    'view' => 'default',
    'id' => NULL,
));

/**
 * Run!
 */
$application->run();
```

# Nette::Environment

- název prostředí (development, production)
- proměnné (appDir, templatesDir)
- služby (Nette::ServiceLocator)
  - Environment::getService(NazevSluzby)
  - Environment::getHttpRequest()
  - Environment::getApplication()
- konfigurace
  - pasivní / aktivní

# Start aplikace – config.ini

```
[common]
var> tempDir = %appDir%/cache
var> foo = %bar% world
var> bar = hello

set> date.timezone = "Europe/Prague"
set> iconv.internal_encoding = "%encoding%"
set> mbstring.internal_encoding = "%encoding%"
set> include_path = "%appDir%/../_trunk;%appDir%/libs"

; Production site configuration data
[production < common]
service> User = "Nette::Security::User"

webhost                = www.example.com
database> params> host  = db.example.com
database> params> username = root
database> params> password = secret
database> params> dbname  = dbname

; overrides values as necessary
[development < production]
database> params> host      = dev.example.com
database> params> username = dev
database> params> password = devsecret
```

# Routování

## Obousměrný router

1. Převést HTTP požadavek na PresenterRequest
2. Převést PresenterRequest na URL adresu

# Routování

## interface Nette::Application::IRouter

```
/**
 *
 * @package Nette::Application
 */
interface IRouter
{
    /**
     * Maps HTTP request to a PresenterRequest object.
     * @param Nette::Web::IHttpRequest
     * @return PresenterRequest|FALSE
     */
    function match(Nette::Web::IHttpRequest $context);

    /**
     * Constructs URL path from PresenterRequest object.
     * @param Nette::Web::IHttpRequest
     * @param PresenterRequest
     * @return string|FALSE
     */
    function constructUrl(PresenterRequest $request, Nette::Web::IHttpRequest $context);
}
```

# Routování

- SimpleRouter
  - `index.php?presenter=home&page=10`
- MultiRouter
  - ArrayList pro prvky IRouter
  - Ideální v kombinaci s objekty Route

# Routování a třída Route

- Příklad použití

```
$router[] = new Route('<presenter>/<view>/<id>', array(  
    'presenter' => 'home',  
    'view' => 'default',  
    'id' => NULL,  
));
```

---

- /products/edit/12/
- /products/edit/12
- /products
- ....

# Routování a třída Route

- Složitější kousky

```
$router[] = new Route('/<presenter>/<view>/<id>', array(
    'presenter' => 'home',
    'view' => 'default',
    'id' -> NULL,
));
```

```
$router[] = new Route('index.php', array(
    'presenter' => 'catalog',
), Route::ONE_WAY);
```

```
new Route('<lang [a-z][a-z]>/<page [a-z0-9./-]+?> ? action=<presenter>', ... );
```

```
new Route('//<host>.taxy.<domain>/<path>', ...);
```

# Routování a třída Route

- Maska se přeloží na jeden optimalizovaný regulární výraz (rychlé jako .htaccess)
- Celý objekt lze uložit do cache
- Pravidla jsou velmi srozumitelná

# Hierarchie tříd Presenter

- BasePresenter
  - HomepagePresenter
  - CatalogPresenter
    - ProductPresenter
- Admin::BasePresenter
  - Admin::CatalogPresenter

# Hierarchie tříd Presenter

- BasePresenter (lang, shopId)
  - HomepagePresenter
  - CatalogPresenter (catalogId)
    - ProductPresenter (productId)
- Admin::BasePresenter (shopId, userId)
  - Admin::CatalogPresenter

# Hierarchie tříd Presenter

- BasePresenter
  - (hlavička, patička, postranní panel)
- HomepagePresenter (obsah)
- CatalogPresenter (obsah, postranní panel)
  - ProductPresenter (obsah)

Namísto masterpages

# Navigační model

Volání mezi presentery

- presenter může mít více „view“
  - `renderDefault()`
  - `renderEdit($id)`
  - `renderDelete($id, $force = NULL)`
- předávání parametrů

# Navigační model

```
<table>
<?php foreach($albums as $album) : ?>
<tr>
  <td><?= $escape($album->title);?></td>
  <td><?= $escape($album->artist);?></td>
  <td>
    <a href="<?= $component->link('edit', $album->id);?>">Edit</a>
    <a href="<?= $component->link('delete', $album->id);?>">Delete</a>
  </td>
</tr>
<?php endforeach; ?>
</table>
```

# Navigační model

```
public function renderEdit($id = 0)
{
    $this->template->title = "Edit Album";
    $album = new Albums();
    if ($id > 0) {
        $this->template->album = $album->fetch($id);
    } else {
        $this->template->album = $album->createBlank();
    }

    // additional view fields required by form
    $this->template->action = $this->link('save!', $id);
    $this->template->buttonText = 'Update';
}
```

# Persistence

- implementuje IStatePersistent

```
interface IStatePersistent
{
    /**
     * Loads state informations.
     * @param array
     * @return void
     */
    function loadState(array $params);

    /**
     * Saves state informations for next request.
     * @param array
     * @return void
     */
    function saveState(array & $params);
}
```

# Persistence

- Presenter automaticky udržuje persistentní parametry označené `@persistent`
- dědičnost

```
class FifteenControl extends Control
{
    /** @persistent array */
    public $order;

    /** @persistent int */
    public $round = 0;

    /** @var bool */
    public $useAjax = FALSE;

    ...
}
```

# Persistence

- Lze uživatelsky zpracovat

```
/**
 * Save params
 * @param array
 * @return void
 */
public function saveState(array & $params)
{
    parent::saveState($params);
    if (isset($params['order'])) {
        $params['order'] = implode('.', $params['order']);
    }
}
```

# Command layer

- implementuje ICommandExecutor

```
/**
 *
 * @package Nette::Application
 */
interface ICommandExecutor
{

    /**
     * @param string
     * @return void
     */
    function executeCommand($command);

}
```

# Command layer

- v šabloně zapíšeme...

```
<a href="<?= $this->link('Click', $x, $y) ?>"><img ....></a>
```

# Command layer

- ...a zavoláme metodu komponenty

```
public function commandClick($x, $y)
{
    if (!$this->isClickable($x, $y)) {
        throw new Exception('Action not allowed');
    }

    $this->move($x, $y);
    $this->round++;
    $this->onAfterClick($this);

    if ($this->order == range(0, self::MAX)) {
        $this->onGameOver($this, $this->round);
    }
}
```

# Command layer

- zAJAXovatění

```
<a href="<?= $this->link('Click', $x, $y) ?>"  
  onclick="return !nette.action(this.href, this)"><img ....></a>
```

```
public function render()  
{  
    if (!$this->beginPartial()) return;  
  
    .... vykresleni komponenty ...  
  
    $this->endPartial();  
}
```

# Událostmi řízený návrh

- Opravdu véééélmi stručně

```
public function renderDefault()
{
    $form = new Nette::Forms::Form;
    $form->addButton('Click me')->onClick[] = array($this, 'formSubmitted');
    ...
    $this->template->form = $form;
}
```

```
public function formSubmitted($sender)
{
    ....
}
```

# Soutěžní otázka

- ???

# Klepnutím zde vložíte nadpis!!!

- <http://nettephp.com>