



# Nette Framework

~~2.3.5~~

~~2.3.6~~

~~2.3.7~~

2.3.8

Martin Zlámal – bořič mýtů v ANT studio s.r.o.  
KIV/WEB – 2015

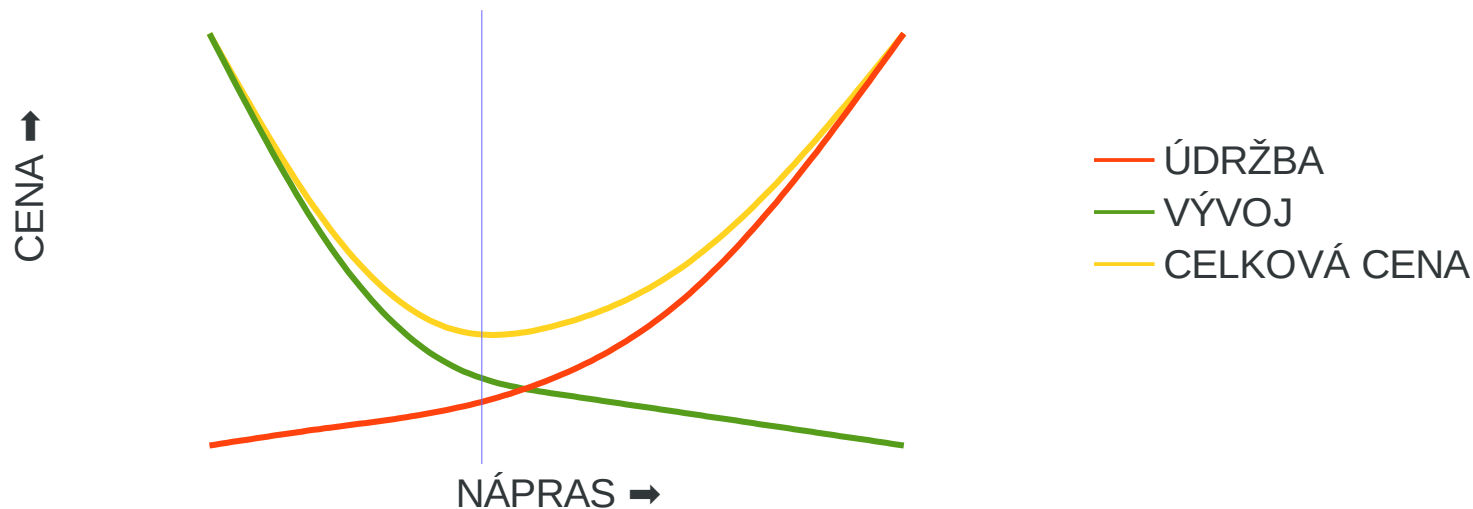
# Třetí nejpoblárnější framework

Best PHP Framework for 2015 ([Sitepoint.com](http://Sitepoint.com))

ČSFD, StartupJobs, Bandzone.cz, Rádio Impuls,  
**Blueboard.cz**, Megapixel, Balíkonoš.cz, **Uložto**,  
Deník.cz, **Lupa.cz**, Testomato, **Mall.cz**, Noviny.sk,  
Ženy.cz, Měšec.cz, **DameJidlo.cz**, Podnikatel.cz,  
**Primat.cz**, E15.cz, dTest, **Rohlik.cz**, Růžový slon.cz,  
**Slevomat.cz**, Vysoké školy.cz, **Root.cz**,  
EDNA.cz, a mnoho dalších... ([zdroj](#))

# Proč framework?

Seznamte se s *křivkou náprasu* (via [@veproza](#) - Marcel Šulek):



# Nette Components

- Dnešní trend je jasný – komponenty vedou
- Místo o frameworku se tak bavíme spíše o jednotlivých komponentách (21):

**Application**, **Bootstrap**, **Caching**, **Component Model**,  
**DI**, **Database**, **Finder**, **Forms**, **Http**, **Latte**, **Mail**, **Neon**,  
**PHP Generator**, **Reflection**, **Robot Loader**, **Safe Stream**,  
**Security**, **Tester**, **Tokenizer**, **Tracy**, **Utils**

# Composer

- „Must have“ každého PHP programátora
- [Packagist](#): The PHP Package repository

Dvě rady:

1. Nechodte do firem, které ho neznají
2. Nechodte do firem, pokud jej neznáte



Buzzword time

# **M**odel **V**iew **P**resenter

Passive View / Supervising Controller

The background is a solid green color. It features several white squares of various sizes, some of which are slightly tilted. These white squares are primarily located in the top-left and bottom-right corners. In the center of the image, there are several overlapping, semi-transparent green squares. The word "MODEL" is written in a white, sans-serif font, centered horizontally and partially overlaid by the central green squares.

MODEL

# Co si představit?

- Za model je většinou považována aplikační logika
- Většina programování se odehrává právě zde
- Nedá se jednoduše ohraničit, je to logická část aplikace
- **Vrstva nezávislá na zobrazení dat**
- Správný návrh je klíčový pro *křivku náprasu*
- Model je nutné dostatečně otestovat



Undefined variable \$appRequest, did you mean \$appRequest? [search](#) [skip error](#)

## Source file ▼

File: .../html/web-project/app/presenters/NanoPresenter.php:37

```
27:      *
28:      * @return Nette\Application\IResponse
29:      */
30:      public function run(Application\Request $appRequest)
31:      {
32:          $this->em->getRepository(Address::class)->findAll();
33:          $result = $this->cache->load('key', function () {
34:              $qb = $this->em->getRepository(Address::class)->createQueryBuilder('a');
35:              return $qb->getQuery()->getResult(\Doctrine\ORM\AbstractQuery::HYDRATE_ARRAY);
36:          });
37:          \Tracy\Debugger::log($appRequest);
38:          return new Responses\JsonResponse($result);
39:      }
40:
41:  }
```

Queries: 1 , time: 0.768 ms, host: 127.0.0.1/workshop

ms	SQL Statement
0.768	<pre>SELECT t0.id AS id_1, t0.street AS street_2, t0.post t0.country AS country_5, t0.of_user AS of_user_6 FROM address t0 .../presenters/NanoPresenter.php</pre>

## System info

Execution time	39.9 ms
Peak of allocated memory	2.80 MB
Included files	252
Classes + interfaces + traits	368 + 70 + 2
Your IP	127.0.0.1
Server IP	127.0.0.1
PHP	7.0.0-1~dotdeb+8.1
Tracy	2.3.6
Server	Apache/2.4.10 (Debian)
Nette Framework	2.4-dev

## Call stack ▼

- .../nette/application/src/Application/Application.php:141 [source](#) ▶ App\Presenters\NanoPresenter->run(arguments ▶)
- .../nette/application/src/Application/Application.php:81 [source](#) ▶ Nette\Application\Application->processRequest(arguments ▶)
- ./var/www/html/web-project/www/index.php:9 [source](#) ▶ Nette\Application\Application->run()

## Variables ▶

## Exception ▶

localhost/web-project/www/?\_tracy\_skip\_error

## Nano:nano

Mask / Class	Defaults	Matched as
api/v1/addresses[/<id [0-9]+>]	presenter = Nano action = nano id = NULL	
✓ [ <path> ]	presenter = Nano action = nano path = NULL	Nano:nano path = NULL
= <presenter>/<action>[/<id>]	presenter = Homepage action = default id = NULL	Homepage:default id = NULL

GET http://localhost/web-project/www/

App\Presenters\NanoPresenter

# Nette\Object

*„PHP gives a huge freedom to developers, which makes it a perfect language for making mistakes.“*

- Striktní třídy
- Syntaktický cukr (automatické settery/gettery)
- Události (**`$form->onSuccess[] = ...`**)
- Rozšiřující metody + sebereflexe

# Nette\DI

- Víte o čem je Dependency Injection?
- Hollywood principle (*Don't call us, we'll call you.*)

Kde vezmeme potřebnou závislost?  
Necháme si ji předat!

- Nevýhoda? Když DI nechápete, tak je to WTF...

# Metaprogramování

- Každý programátor by měl být dostatečně **líný**
- Každý programátor by měl být dostatečně **mazaný**
- Nette\Neon – (lze použít i PHP konfiguraci)

## REÁLNÁ UKÁZKA (1, 2)

Řekne více než tisíc slov...

# Rozšíření pro DIC

- Nette je rozdělené na komponenty – to už víme
- Každá komponenta je ale spíš rozšíření DIC
- Umožňují připravit aplikaci v „*compile-time*“

Composer require → registrace rozšíření → PROFIT

# Nette\Database

```
use Nette\Database\Context;
```

```
$database = new Context($connection);
```

```
$database->query('INSERT INTO users', array( // parametr může být pole
    'name' => 'Jim',
    'created' => new DateTime, // nebo objekt DateTime
    'avatar' => fopen('image.gif', 'r'), // nebo soubor
), ...); // je možné také provést více vložení najednou
```

```
$database->query('UPDATE users SET ? WHERE id=?', $data, $id);
```

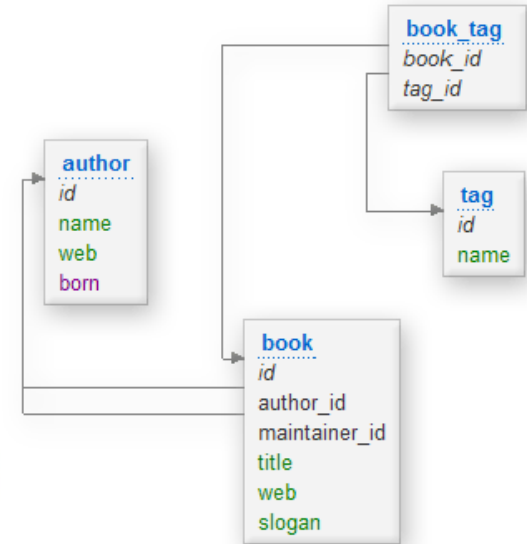
```
$database->query('SELECT * FROM categories WHERE id=?', 123)->dump();
```

# Nette\Database\Table

```
$books = $context->table('book');

foreach ($books as $book) {
    echo 'title:      ' . $book->title;
    echo 'written by: ' . $book->author->name;

    echo 'tags: ';
    foreach ($book->related('book_tag') as $bookTag) {
        echo $bookTag->tag->name . ', ';
    }
}
```



# Doctrine maybe?

- Určitě doporučuji jakoukoliv formu ORM
- Doctrine vychází z Hibernate (možná znáte spíš)
- Teoreticky pomůže s lepším návrhem modelu
- Ale prakticky? Ano, jen pozor na anemický návrh...

Do Nette existuje oblíbené rozšíření **Kdyby\Doctrine** ([link](#))



Model je však o něčem jiném

Na to teď není čas. Někdy příště...

Podívejte se také na další zajímavé části frameworku:

Caching, Finder, Http, Mail,  
Safe stream, Security, Tester, ...

The image features a solid green background. In the top-left and bottom-right corners, there are clusters of small, white, slightly tilted squares. Scattered across the center are several larger, rounded green squares, some of which have thin yellow outlines. The word "VIEW" is centered in a white, sans-serif font, with the letters partially overlapping the central green squares.

VIEW

# Motivace – umíte escapovat?

Ok, jak se escapují následující kontexty?

"xxx'yyy</img<!--XSS--" ← původní XSS vektor

```
"xxx'yyy</img<!--XSS--  
"xxx'yyy&lt;/img&lt;!--XSS--  
<a>"xxx'yyy&lt;/img&lt;!--XSS--</a>  
<a "&quot;xxx'yyy&lt;/img&lt;!--XSS--"></a>  
<a href="&quot;xxx'yyy&lt;/img&lt;!--XSS--"></a>  
<a style="\&quot;xxx\'yyy&lt;/img&lt;!--XSS--"></a>  
<script>"\"xxx'yyy</img&lt;!--XSS--"</script>  
<style>"\"xxx\'yyy</img&lt;!--XSS--</style>  
<!-- "xxx'yyy</img<!--XSS-- -->
```

# HTML, XML – htmlspecialchars

- Speciální znaky: < & " ' >
- Gotchas: komentáře, kódování, maska (ENT\_\*)

XML only



```
$string = preg_replace('#[\x00-\x08\x0B\x0C\x0E-\x1F]+#', '', $string)  
htmlspecialchars($string, ENT_QUOTES, 'UTF-8');
```

# A co další kontexty?

CSS:

```
addslashes($string, "\x00..\x1F!\"#$%&'()*+,-./:;<=>?@[\\]^`{|}~");
```

JS:

```
$json = json_encode($string, PHP_VERSION_ID >= 50400 ?  
                        JSON_UNESCAPED_UNICODE : 0);  
str_replace(["\xe2\x80\xa8", "\xe2\x80\xa9", ']]>', '<!'],  
            ['\u2028', '\u2029', ']]\x3E', '\x3C!'], $json);
```

## DOST!

# Latte\Latte

- Řešením je Context-Aware Escaping
- „*Less code, more security*“
- Ale není to jen o escapování...
- Latte je plnohodnotný šablonovací systém

Nikdy nesrovnávejte Latte se Smarty. Prosím.

```
<ul n:if="$items">
{foreach $items as $item}
    <li id="item-{$iterator->counter}">{$item|capitalize}</li>
{/foreach}
</ul>
```

# VS



```
<?php if ($items): ?>
    <?php $counter = 1 ?>
    <ul>
    <?php foreach ($items as $item): ?>
        <li id="item-<?php echo $counter++ ?>"><?php
        echo htmlspecialchars(mb_convert_case($item, MB_CASE_TITLE)) ?>
        </li>
    <?php endforeach ?>
    </ul>
<?php endif?>
```



## (n:)?makra, filtry

- `{foreach $x as $y} ... {/foreach}`
- `<p n:foreach="$x as $y"> ... </p>`
- `<p n:inner-foreach="$x as $y"> ... </p>`
- `{$heading|lower|capitalize|truncate:20}`

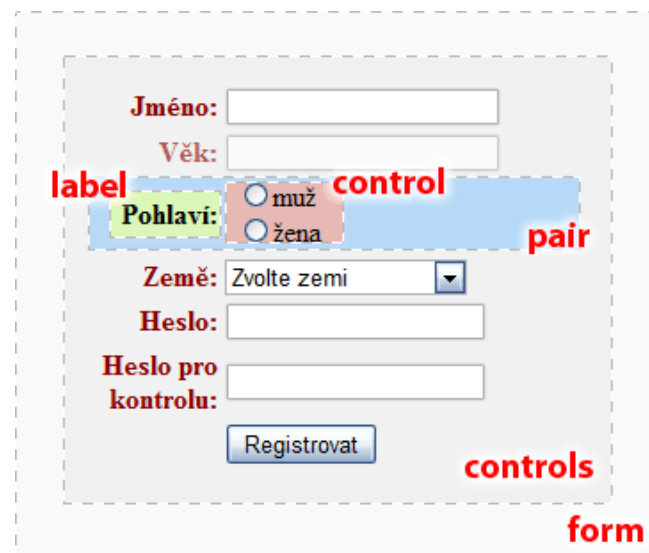
Makra a n:makra nejsou 1:1 (**n:href**)



# Nette\Forms

Formuláře jsou bohužel složité a nudné:

- Validace na serveru a u klienta
- Zpracování těch správných hodnot (!)
- Ochrana před CSRF (víte co a jak?)
- Překlady, HTML5 atd...
- A hlavně – na nic nezapomenout!



The diagram illustrates a registration form with various annotations:

- label**: Points to the 'Pohlaví' label.
- control**: Points to the radio buttons for 'muž' and 'žena'.
- pair**: Points to the 'muž' and 'žena' radio buttons.
- controls**: Points to the 'Registrovat' button.
- form**: Points to the entire form structure.

The form fields include:

- Jméno:
- Věk:
- Pohlaví: ☐ muž ☐ žena
- Země: Zvolte zemi
- Heslo:
- Heslo pro kontrolu:
- Registrovat



# Komponenty

(ted' nemyslím části frameworku)

# Komponenty jsou nutnost

- Komponenta je znovupoužitelná část aplikace
- Měla by být úplně nezávislá
- Často se používají jen pro přesunutí kódu z presenteru
- Komponenty je možné libovolně zanořovat
- Vlastní signály, flashmessages, odkazy, ...
- Možno vytvořit až v presenteru (pouze pro jednoduché)

# UI\Control

- Umí se samostatně vykreslit (`{control xyz}`)
- [addons.nette.org](https://addons.nette.org) (většinou formou rozšíření)
- Perzistentní parametry
- Jednoduchá práce s AJAXem
- Komponenty se chovají jako „lazy“

**Jen rychlé představení, jdeme dál...**

The background is a solid green color. It is decorated with several white squares of varying sizes, some of which are slightly tilted. There are also several larger, semi-transparent green squares with thin yellow outlines scattered across the background. The word "PRESENTER" is centered in the middle of the image in a white, serif font. The letter "P" is significantly larger and bolder than the other letters.

**P**RESENTER

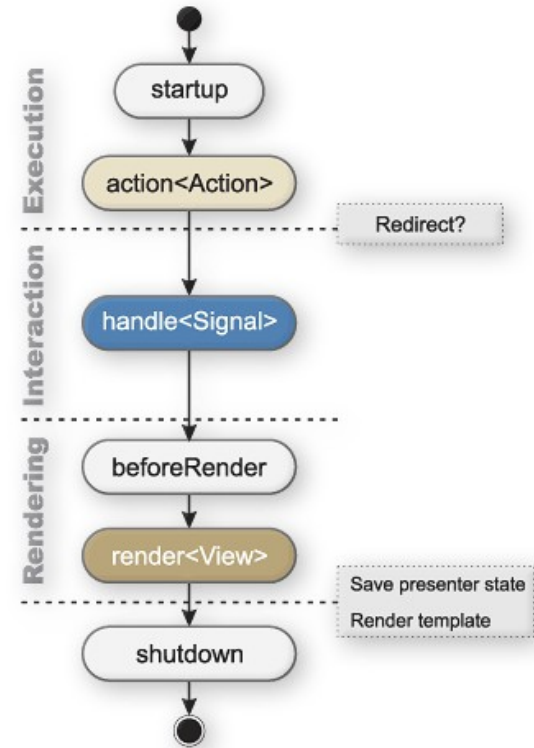
# Presenter (Nette\Application)

- Presenter je vlastně také komponenta (UI\Control)
- Implementuje metodu **run()** z rozhraní IPresenter
- **Neobsahuje** aplikační logiku – na to je model
- Na presenter dopadají všechny requesty
- Snažte se udržovat presenter jednoduchý

(viz křivka náprasu)

# Životní cyklus UI\Presenter

- Nejedná se o cyklus – už z principu
- V action je ještě čas změnit view
- Pozor na to, že (už) v handle\* se zpracovává formulář
- Ve skutečnosti existuje ještě **afterRender()**
- Neplatí pro všechny IPresentery!



# MicroPresenter implements IPresenter

- Nemá životní cyklus! (a spoustu dalších věcí)
- Samotný presenter je velmi mocný – ne vždy je potřeba
- Stejně tak je možné implementovat si vlastní
- Je sice jednoduchý, ale hodí se na rychlé odpovědi

NanoPresenter + (Doctrine) + Cache → JSON output  
Celkem za ~10 ms, min. 8.91 ms



# Obousměrné routování URL

1. `HttpRequest` → `ApplicationRequest`
2. `ApplicationRequest` → absolutní URL

```
<a n:href="Product:detail $productId">
```

Detail produktu

```
</a>
```

```
http://example.com/product/detail/123
```

```
new Route('<presenter>/<action>[/<id>]', 'Homepage:default');
```

# AJAX zjednodušeně

- Doporučuji použít **nette.ajax.js** (nepovinné)
- Každý potomek `UI\Control` se umí překreslit
- `{snippet xyz} ... {/snippet}`

*Událost na straně klienta → obsluha signálu (handle) →  
→ vrácení dat → překreslení části stránky*

# Kdyby CMS

- Rychlá integrace cizích knihoven do Nette
- Nepatří pod Nette jako takové

[help.kdyby.org](http://help.kdyby.org)

Doctrine, Redis, Console, Events, Translation, AOP, ElasticSearch,  
RabbitMQ, Facebook, Monolog, Autowired, Wkhtmltopdf, Money, ...

# Kam a jak dál?



- Existuje [dokumentace](#), [fórum](#), [školení](#) v Praze
- Používejte [API](#)
- Můžete štěbetat na [gitteru](#)
- Pošlete PR na [github](#)
- Přijďte na [Poslední sobotu](#) (leden 2016)
- Kontaktujte přímo [mě](#), zajdeme na pivo

The background is a solid green color. It is decorated with several white squares of varying sizes, some of which are slightly tilted. There are also several larger, semi-transparent green squares with thin white borders, some of which are tilted. The text "Nette PRO" is centered in the middle of the image.

Nette **PRO**

# Nette PRO[fesionálně]

- Řeší otázku jak vlastně financovat Nette Framework
- Na dobrovolnické bázi to dlouhodobě nefunguje
- Dva balíčky: pro jednotlivce a pro firmy
- Cílem je získat peníze (viz definice profesionála)
- Účastníci mají možnost ovlivňovat budoucnost Nette
- Pro běžné uživatele se nic nemění



Hledáme ty nejlepší z  
nejlepších...

[antstudio.cz/kariera](http://antstudio.cz/kariera)



Děkuji za pozornost

zlml.cz, @mrtnzlml



BONUS: Jak funguje start Nette aplikace?

## REÁLNÁ UKÁZKA

Řekne více než deset tisíc slov...