Front-Controller

Obsah

- O co se jedná
- Motivace
- Popis
- Příklad
- Webové návrhové vzory
- Frameworky

O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace

O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace
- "Řídí průjezd" stránkou

O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace
- "Řídí průjezd" stránkou
- Poskytuje jednotný vstup do stránky

O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace
- "Řídí průjezd" stránkou
- Poskytuje jednotný vstup do stránky

Uživatel: www.stranka.cz/kontakt

./products.php

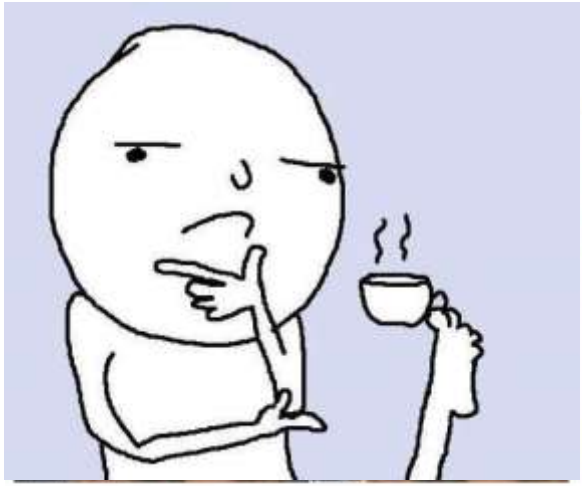
./contact.php

./about-us.php

O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace
- "Řídí průjezd" stránkou
- Poskytuje jednotný vstup do stránky

Uživatel: www.stranka.cz/kontakt



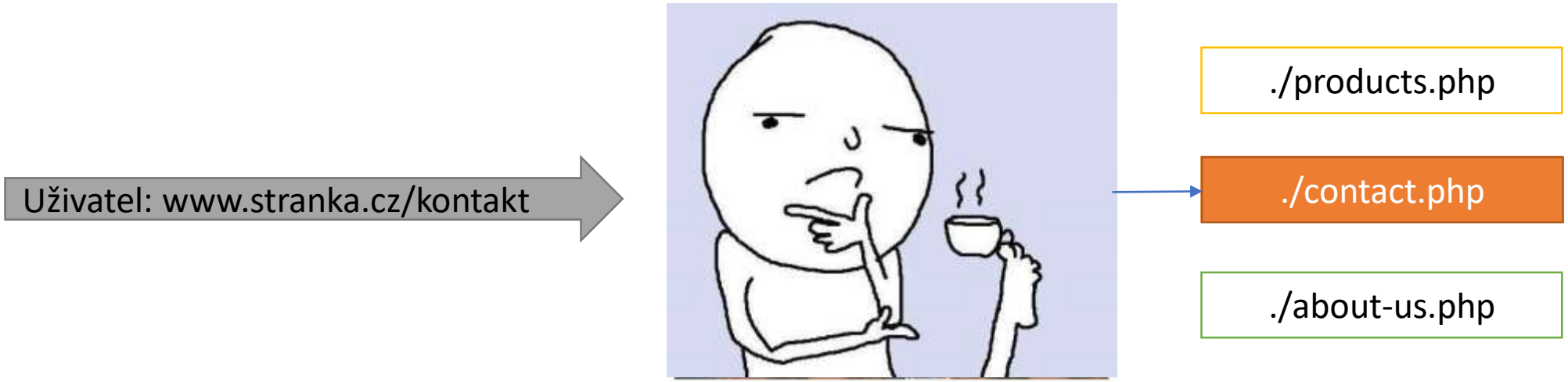
./products.php

./contact.php

./about-us.php

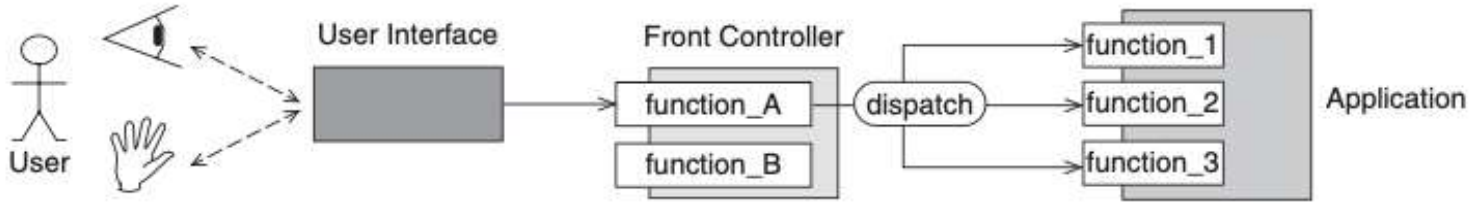
O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace
- "Řídí průjezd" stránkou
- Poskytuje jednotný vstup do stránky



O co se jedná?

- (Rozšiřující) návrhový vzor pro webové aplikace
- "Řídí průjezd" stránkou
- Poskytuje jednotný vstup do stránky



Zdroj: Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing

Motivace

- "**DRY**" *(Don't repeat yourself)*
  - Obvykle se na webu něco opakuje
    - Import hlavičky, patičky, ...
- Rozumný způsob, jak navrhovat webové aplikace
  - "Renderování" podstránek na jednom místě
- /objednavka/2
  - Bude mít možná něco společného s /objednavka/1

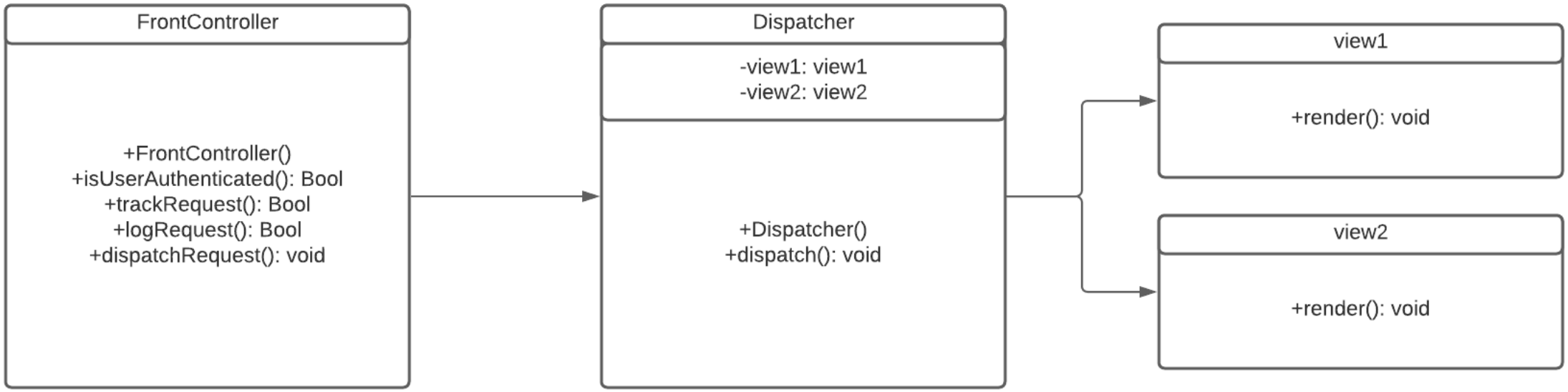
Popis

- Návrhový vzor
- Zpracování požadavků z jednoho místa
- Možnost, jak na jednom místě:
  - Zalogovat požadavek
  - Ověřit uživatele
  - Předat požadavek "dále"
  - Ověřit/sanitizovat vstup

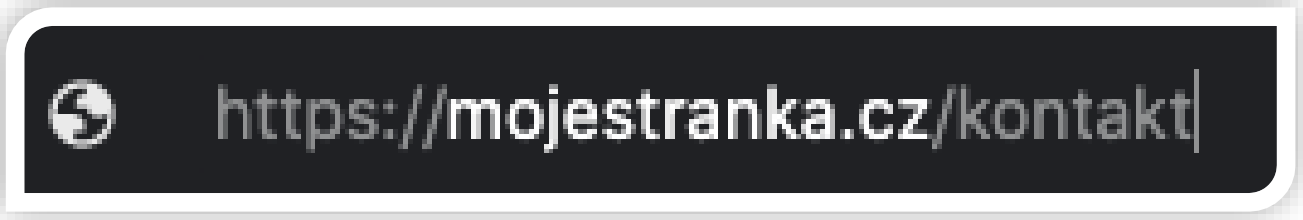
Popis (možný návrh)

- Třída FrontController
  - Zabezpečuje základní operace před "vstupem"
- Třída Dispatcher
  - Řeší, co chtěl uživatel zobrazit a zobrazí

Popis (možný návrh)

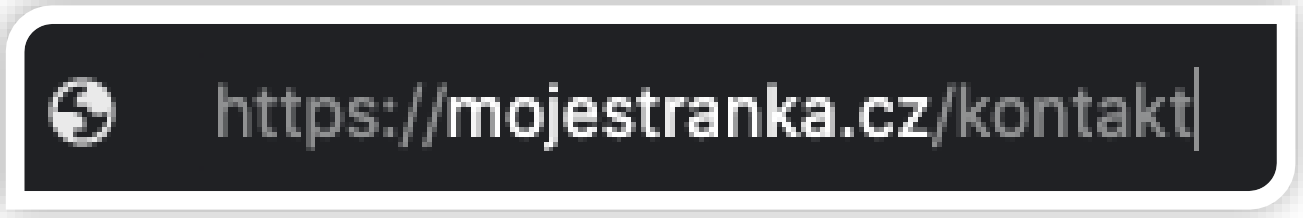


Příklad (Úroveň webserveru)



- Na úrovni webserveru
    1. Vezme **/kontakt**
    2. Upraví cestu na např.: index.php?page=kontakt
        - Index.php představuje Front Controller

Příklad (Úroveň aplikace)



- Na úrovni aplikace
  1. FrontController dostane požadovanou stránku v parametru
  2. Provede sanitizaci
  3. Ověří uživatele
  4. Předá data dispatcheru
  5. Dispatcher "sestaví" stránku

Možná implementace

```
class FrontController{

    private $_page_param;
    private $_template_context;
    //this function handles rendering header
    private function show_header() {…
    }
    //this function handles rendering footer
    private function show_footer() {…
    }
    //$_GET['page'] parser
    private function handle_page_parameter(){…
    }
    //page parser
    private function parse_page(){…
    }
    //only public function of class FrontController
    public function run(){…
    }

}
```

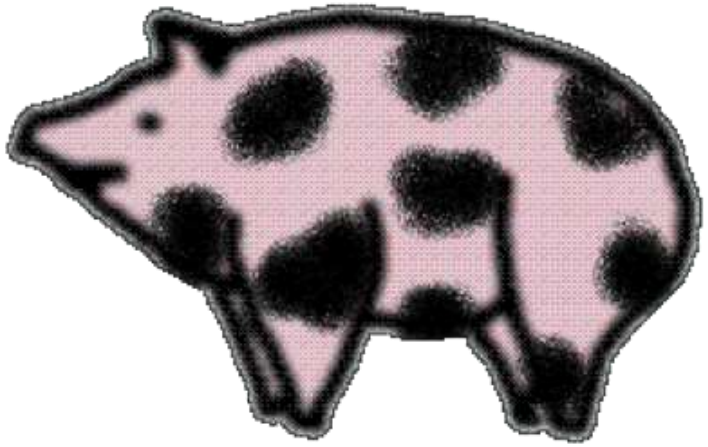
Ne světlé písmo na tmavém pozadí !!!

Kód ne jako obrázky, ale jako text !

Příklad (Úroveň aplikace)

- Příkazový přístup

```php
<?php
switch ($_GET['page']) {
    case 'home':
        $page = 'home.php';
        break;
    case 'contact':
        $page = 'contact.php';
        break;
    case 'about-us':
        $page = 'aboutus.php';
        break;
}
```



Příklad (Úroveň aplikace)

- Deklarativní přístup

```php
<?php
$pages = [
    'contact' => 'contact.php',
    'home' => 'home.php',
    'about-us' => 'aboutus.php'
];

$page = $pages[$_GET['page']];
```

Možná implementace

```php
//$_GET['page'] parser
private function handle_page_parameter(){
    $response = NULL;
    //if page is not set or it's invalid, function will throw 404
    $page = isset($_GET['page']) ? $_GET['page'] : NULL;
    if($page===NULL || $page=== ""){
        $response = 400;
        return $response;
    }
    //regex check if the string is eg:
    //a/a/a, a, a/a
    // and not //a, //, /, ../, _a/, etc.
    if(! preg_match('/^(([a-zA-Z]+)([\/]?)([a-zA-Z]+))+$/', $page)) {
        $page = NULL;
        $response = 400;
        return $response;
    }
    //let check if it's page, or dir
    if($page){
        if(is_dir(__DIR__.'/templates/'.$page)){
            //it's a directory, let's append 'index.php'
            $page = $page.'/index.php';
        } elseif(is_file(__DIR__.'/templates/'.$page.'.php')){
            //it's a file, let's append '.php' to the end
            $page = $page.'.php';
        } else {
            //it's some weird crap that is invalid - 404
            $page = NULL;
        }
    }
    if(!$page || !is_file(__DIR__.'/templates/'.$page)){
        //page after all sanitization did not pass
        $this->_page_param = NULL;
        $response = 404;
        return $response;
    }
    //page exists
    $this->_page_param = $page;
    $response = 200;
    return $response;

}
```

Výhody/Nevýhody

- + Výhody
  - Nenutí programátora se opakovat
    - Logger, Request tracker, Dispatcher, autentikace, autorizace
  - Snižuje chybovost
  - Zvyšuje bezpečnost
  - Snižuje paměťovou náročnost aplikace

- - Nevýhody
  - Horší škálovatelnost
    - Díky "jednotnému" vstupu
    - Může zapříčinit hrdlo requestů

Frameworky

- Frameworky využívající Front-Controller:
  - ASP.NET MVC (C#)
  - Spring Framework (Java)
  - Drupal (PHP)

Q/A

Je zmíněno, že tento vzor je málo (nebo vůbec) škálovatelný, existují tedy nějaké rozšíření/modifikace tohoto vzoru, které by umožňovaly škálovat?

- problém je v tom, že všechny requesty procházejí jedním vstupem
- webová app bude mít jen jeden skript, který bude delegovat požadavky mezi jednotlivé stránky
- tedy škálovatelnost ve smyslu delegace requestů mezi více skriptů není možná v logice (front-controller)

Možná by bylo dobré uvést spolupráci Front Controlleru s ostatními podcontrollery.

- Pokud to odpoví na otázku, tak na posledním slide bude MVC x Front-Controller vztah

….A ještě mi není jasné v čem znamená horší škálovatelnost. Server má na starosti přijímání requestu a tedy by se měl o škálovatelnost starat on si myslím (nejakým paralelním zpracováním dotazu napr.).

- viz předpředchozí otázka
- server přijímá requesty a předává je naší aplikaci, ta je ale přijímá pouze na jednom místě (front-controller)

Ako by sa riešilo ak by ten page mal viacero úrovní? V prezentácii je príklad mojestranka.cz/kontakt, ale co ak by tam bolo mojestranka.cz/kontakt/konkretny_kontakt/...?

- Implementace Front-Controlleru, které jsem viděl třeba ve webových aplikacích, zpravidla tu cestu /kontakt/kontkretni_kontakt vzdy nejprve rozparsovaly na nějaký URL parametr.
- Pomocí toho parametru se poté vyrenderuje konkrétní stránka, takže to záleží jen na logice toho controlleru - možné to samozřejmě je

Aký je medzi MVC a Front Controller nejaký hlbší rozdiel?

- MVC je obecný koncept
- Koncept MVC mohu realizovat např. prostřednictvím front-controlleru
- Model-View-*(Front-)*Controller