



Webové aplikace

Ing. Přemysl Brada, MSc., Ph.D.
Ing. Martin Dostal

Katedra informatiky a výpočetní techniky, FAV, ZČU v Plzni

Harmonogram

Dopolední blok

9:00 – 12:30

- Ing. Dostal – Úvod, XHTML + CSS
- Ing. Brada, Ph.D. – Java

Oběd

12:30 – 13:30

Odpolední blok

13:00 – 15:00

- Ing. Dostal – Php, ukázky webových aplikací

Webová aplikace vs. webová služba

- Co to je webová aplikace?
- Co to je webová služba?
- Jaký je hlavní rozdíl?

Architektura klient - server

Výhody:

- data jsou uložena na jednom místě
- nejsou redundantní
- není problém s aktuálností dat
- k datům můžeme přistupovat z libovolného pc

Webový server

- je program, který vyřizuje požadavky klientů
- je připojen k počítačové síti
- komunikuje s využitím protokolu HTTP



Zdroj: webserver.com

Architektura webových aplikací

Jak dělat webové aplikace pořádně

Motivační příklad – Školní IS: známky studentů

index.php (úvodní strana aplikace)

```
<? php
```

```
include ("settings.inc.php"); // konfigurace
```

```
// načtení obsahu z db MySQL5
```

```
$query = "select * from t where s='uvod'";
```

```
?>
```

```
// začátek html stránky
```

```
<html>
```

```
<head>...</head>
```

```
<body><div id="obsah">
```

```
<?
```

```
    // výpis úvodní stránky
```

```
    echo "<div
```

```
        class="uvod">$uvod</div>";
```

```
?>
```

```
</div></body></html>
```

znamky.php (známky studenta)

```
<? php
```

```
include ("settings.inc.php"); // konfigurace
```

```
// načtení známek pro studenta z MySQL5
```

```
$query = "select * from z where id=1"; ...
```

```
?>
```

```
// začátek html stránky
```

```
<html>
```

```
<head>...</head>
```

```
<body><div id="obsah">
```

```
<?
```

```
    // výpis úvodní stránky
```

```
    echo "<div
```

```
        class="znamky">$zn</div>";
```

```
?>
```

```
</div></body></html>
```

Architektura webových aplikací

Základní typy:

- Bez použití vrstev = vše v jednom souboru – viz. předchozí ukázka
- Dvouvrstvá architektura – oddělení vzhledu a aplikace
- Třívrstvá architektura (MVC)
 - Model – např. databáze
 - View – zobrazení stránky uživateli
 - Controller – obsluha požadavků
- Vícevrstvá (dělení view, webservice vrstva, fasáda pro integrace s jinými systémy)

3-vrstvá architektura - realizace



Architektura
= nápad,
myšlenka

Existují různá řešení:

- Všechny 3 vrstvy jsou na 1 webovém serveru:
 - **prezentační vrstva** – připravuje html výstup pro klienta
 - **aplikační vrstva** – zpracovává požadavky od klienta
 - **databázová vrstva** – načítá data z databáze
- Všechny 3 vrstvy mohou být na různých místech:
 - **prezentační vrstva** – může být použit sw na klientském pc
 - **aplikační vrstva** = samostatný aplikační server
 - **databázová vrstva** = specializovaný databázový server

Logické vrstvy

Fyzické vrstvy

Motivační příklad – řešení

index.php (úvodní strana aplikace)

```
<? php
```

```
include ("settings.inc.php"); // konfigurace
```

```
// načtení obsahu z db MySQL5
```

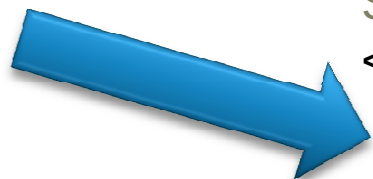
```
$uvod = $aplikace->Nacti("uvod");
```

```
$uvod_s_html = $prezentacni  
->VlozObsah($uvod);
```

```
$page->nastavObsah($uvod_s_html );
```

```
$page->show();
```

```
?>
```



Šablona:

```
<html>
```

```
<head></head>
```

```
<body>
```

```
<container>{obsah}</container>
```

```
</body>
```

```
</html>
```

znamky.php (známky studenta)

```
<? php
```

```
include ("settings.inc.php"); // konfigurace
```

```
// načtení známek pro studenta z MySQL5
```

```
$znamky= $aplikace->Nacti("znamky", 5);
```

```
$znamky_s_html = $prezentacni  
->VlozObsah ($znamky);
```

```
$page->nastavObsah($znamky_s_html);
```

```
$page->show();
```

```
?>
```



Protokol HTTP (HyperText Transfer Protocol)

- protokol pro přenos objektů libovolného typu (stránky HTML, obrázky, PDF) mezi webovým serverem a prohlížečem (různé „mime typy“)
- bezstavový protokol modelu požadavek/odpověď – problémy:
 - server nemá stálé spojení s klienty a nemůže je proto jednoznačně identifikovat
 - řešení: SESSION (viz. dále)

PROCES KOMUNIKACE:

1. navázání spojení
2. zaslání požadavku klientem
3. zaslání odpovědi serverem
4. uzavření spojení

HTTP – ukázka komunikace

Dotaz:

Logické místo a SEO

GET /clanky/obsah.html HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT)

Host: www.server.cz

Odpověď:

Stavový kód

HTTP/1.1 **200 OK**

Server: Microsoft-IIS/5.0

Date: Wed, 06 Dec 2000 13:37:40 GMT

X-Powered-By: PHP/4.0.3pl1

Content-type: text/html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<html><head> <title>Titulek</title>
```

```
<link rel="stylesheet" type="text/css" href="„styl.css"> ...
```

HTTP – stavové kódy

1xx - informativní kód

2xx - úspěšné vyřízení požadavku

3xx - přesměrování

4xx - chyba klienta

5xx - chyba na straně serveru

Přenos dat od uživatele na server

Existují 2 základní metody:

- **GET** – vložení proměnných do URL
- **POST** – odeslání vyplněných HTML formulářů

Obě metody je možné použít k **navigaci** v rámci webové aplikace

GET – např. index.php?page=uvod

Výhody:

- tyto stránky lze uložit do záložek v prohlížeči

Nevýhody:

- není vhodné pro přenos např. celého textu článku

POST – např. index.php

Výhody:

- vhodné pro přenos např. textu článku

Nevýhody:

- stránky nelze uložit do záložek prohlížeče

Ukázka HTTP headers – data z formuláře

- Data jsou otevřená a snadno čitelná/manipulovatelná
- Pokud použiji protokol HTTP = celá komunikace je čitelná a lze snadno odchytit např. uživatelské jméno a heslo
- Mohu však použít protokol HTTPS a pak je celá komunikace šifrovaná (použití např. v internetovém bankovníctví)
- **Ukázka:** odeslání formuláře s loginem a heslem
- **Ukázka2:** šifrované spojení – např. u připojení na portál

Postup:

1. Google chrome: Nástroje – Nástroje pro vývojáře – záložka Network a pak Headers
2. Vyplnit a odeslat form – bude to vidět v sekci Headers/Form data
3. V Request headers: seznam.cz a portal.zcu.cz (šifr.)