#

# Instalace webové aplikace komunikující s ABRA Web Service

Obsah

[Instalace webové aplikace komunikující s ABRA Web Service 1](#_Toc398560511)

[1. Úvod 3](#_Toc398560512)

[2. Instalace a konfigurace serveru Apache 3](#_Toc398560513)

[2.1. Stažení a instalace 3](#_Toc398560514)

[2.2. Konfigurace 8](#_Toc398560515)

[3. PHP 12](#_Toc398560516)

[3.1. Stažení a instalace 12](#_Toc398560517)

[3.2. Konfigurace PHP 17](#_Toc398560518)

[4. Dokončení celé instalace 18](#_Toc398560519)

[5. Nastavení webové aplikace 20](#_Toc398560520)

[6. Systémové URL 21](#_Toc398560521)

## Úvod

K rozfungování aplikace je potřeba následující:

1. Apache web server 2.2+
2. PHP 5.3.9+
3. Nastavení samotné aplikace

## Instalace a konfigurace serveru Apache

### Stažení a instalace

1. Na <http://httpd.apache.org/download.cgi> stáhneme instalátor

Win32 Binary without crypto (no mod\_ssl) (MSI Installer):

1. Proklikáme průvodce instalací:






Zde definujeme doménu společnosti, například „domena.cz“, nebo „abra.eu“





Instalovat budeme do „Program Files“ nebo „Program Files (x86)“ jak je běžně zvykem.








1. Nyní máme nainstalovaný webový server Apache, který běží na portu 80.
2. Pro funkčnost naší aplikace je však potřeba ještě v konfiguraci Apache:
	1. **Povolit modul mod\_rewrite**
	2. **Nastavit DocumentRoot**
	3. **Nastavit Virtual Host pro naši aplikaci**

### Konfigurace

V souboru **httpd.conf** se nachází nastavení Apache. Tento soubor je ve výchozí instalaci na této cestě:

c:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf

Na 64bit. systému bývá Apache nainstalován na této cestě:

c:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\httpd.conf

#### Povolení PHP souborů

1. Protože budeme používat Apache společně s PHP, musíme nastavit, aby Apache kromě souboru index.html poskytoval automaticky také soubory index.php
2. Najdeme sekci:

<IfModule dir\_module>

 DirectoryIndex index.html

</IfModule>

a změníme jí na:

<IfModule dir\_module>

 DirectoryIndex index.html index.php

</IfModule>

#### Povolení modulu mod\_rewrite

1. V souboru nalezneme řádek

#LoadModule rewrite\_module modules/mod\_rewrite.so

a odmažeme mu mřížku (tím ho odkomentujeme – povolíme modul)

LoadModule rewrite\_module modules/mod\_rewrite.so

#### Nastavení DocumentRoot

1. DocumentRoot je umístění v rámci souborového systému serveru, kde budou uloženy aplikace (stránky, chcete-li weby). Nastavuje se rovněž v souboru **httpd.conf**
2. Výchozím nastavením je:

DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"

doporučuji však DocumentRoot změnit na nějaké jiné umístění, například:

DocumentRoot "c:/Web/WWW/“

1. S tímto souvisí i nastavení pro adresář. Najděte následující sekci <Directory>…</Directory>**:**

#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">
.
.
.
</Directory>

a změňte jí na stejné umístění, kam jste nasměrovali DocumentRoot

<Directory "c:/Web/WWW/">

**Zároveň je důležité v této (<Directory>…</Directory>) sekci najít**

**AllowOverride None**

**A změnit ho na**

**AllowOverride All**

**To zajistí, aby naše aplikace mohla používat plně soubor .htaccess.**

#### VirtualHost pro aplikaci

1. Nejprve v souboru **httpd.conf** nalezneme následující odkaz na další konfigurační soubor s VirtualHost konfigurací a povolíme ho (odmazáním mřížky). Odkazy na další konfigurační soubory bývají v **httpd.conf** na samotném konci. Najděte:

# Virtual hosts

#Include conf/extra/httpd-vhosts.conf

A změňte za (odkomentujte):

# Virtual hosts

Include conf/extra/httpd-vhosts.conf

1. Nyní budeme editovat v předchozím kroku povolený soubor **httpd-vhosts.conf**

c:\Program Files\Apache Software Foundation\Apache2.2\conf\extra\httpd-vhosts.conf

1. Především se ujistíme, zda je odkomentovaný (bez mřížky na začátku) řádek

NameVirtualHost \*:80

1. Dále v souboru **httpd-vhosts.conf** již mohou být připravená vzorová data:

<VirtualHost \*:80>
 ServerAdmin webmaster@dummy-host.domena.cz
 DocumentRoot "C:/ … Apache2.2/docs/dummy-host.domena.cz"
 ServerName dummy-host.domena.cz
 ServerAlias www.dummy-host.domena.cz
 ErrorLog "logs/dummy-host.domena.cz-error.log"
 CustomLog "logs/dummy-host.domena.cz-access.log" common
</VirtualHost>

<VirtualHost \*:80>
 ServerAdmin webmaster@dummy-host2.domena.cz
 DocumentRoot "C:/ … Apache2.2/docs/dummy-host2.domena.cz"
 ServerName dummy-host2.domena.cz
 ErrorLog "logs/dummy-host2.domena.cz-error.log"
 CustomLog "logs/dummy-host2.domena.cz-access.log" common
</VirtualHost>

VirtualHost nám vlastně říká, na jaké adrese která aplikace (které stránky) poběží. Podle toho, o jakou doménu (jejíž DNS záznam je nasměrován na daný server) požádáme, Apache pozná, jaký VirtualHost je následně použít.

Pokud chceme naší aplikaci provozovat například na:

http://schvalovani.domena.cz/

upravíme konfiguraci VirtualHostů takto:

<VirtualHost \*:80>
 ServerAdmin webmaster@dummy-host.domena.cz
 DocumentRoot "c:/Web/WWW/"
 ServerName dummy-host.domena.cz
 ServerAlias www.dummy-host.domena.cz
</VirtualHost>

<VirtualHost \*:80>
 ServerAdmin webmaster@dummy-host2.domena.cz
 DocumentRoot "c:/Web/WWW/schvalovani/www/"
 ServerName schvalovani.domena.cz
 ServerAlias www.schvalovani.domena.cz
 ErrorLog "c:/Web/WWW/schvalovani/log/error.log"
 CustomLog "c:/Web/WWW/schvalovani/log/access.log" common
</VirtualHost>

Nejprve si všimněme druhého VirtualHostu:

* **ServerAdmin** určuje pouze e-mail zodpovědné osoby
* **DocumentRoot** určuje kořenový adresář pro naší aplikaci (neplést s DocumentRootem celého Serveru). Nasměrovat na podadresář /www v rámci aplikace
* **ServerName** je doména (v tomto případě subdoména – což bude asi nejčastější případ), na které aplikace poběží
* V direktivě **ServerAlias** je možno definovat i aliasy, ale zejména v případě subdomény je možné ji zcela vynechat
* **ErrorLog** a **CustomLog** jsou cesty k logům aplikace. Nasměrovat na podadresář /log v rámci aplikace. Pojmenování error.log a access.log, může takto zůstat.

První VirtualHost se definuje z toho důvodu, že pokud se na stránku přistoupí přes doménu / subdoménu, která nemá odpovídající VirtualHost, Apache by nevěděl, o jaký web (jakou aplikaci) je požádáno. Proto automaticky zvolí první VirtualHost, na který v definici narazí a ten použije. Je proto dobré nasměrovat ho např. do DocumentRootu webu, nebo na stránku s chybovým hlášením apod. V konfiguraci výše je směrován do DocumentRootu webu.

1. Nyní máme nakonfigurovaný Apache a můžeme přistoupit k instalaci PHP

**Poznámka: Pro zajištění funkčnosti adresy „schvalovani.domena.cz“, kde poběží aplikace je nutné zabezpečit, aby DNS záznam pro danou subdoménu byl nasměrován na IP adresu stroje, kde aplikace běží. Toto nastavení provádí zpravidla IT technik konkrétní společnosti.**

## PHP

### Stažení a instalace

1. Na URL <http://windows.php.net/download/> je dostupný instalátor PHP
V době psaní tohoto dokumentu konkrétně ve verzi 5.3.13
2. Zajímá nás Thread Safe verze

<http://windows.php.net/downloads/releases/php-5.3.13-Win32-VC9-x86.msi>

*Poznámka: Verze PHP 5.4, 5.5 a 5.6 nemá instalátor, ale aplikaci je možné i s touto verzí PHP bez problémů provozovat.*

1. Proklikáme průvodce instalací:





Instalovat budeme opět do „Program Files“ resp. „Program Files (x86)“



PHP se instaluje jako modul serveru Apache, zvolíme první možnost



Dále nastavíme cestu k adresáři, kde je naše konfigurace Apache, instalátor PHP ji sám upraví.



Dále zvolíme, jaké rozšíření PHP nainstalovat. Kromě základních se budou hodit FileInfo a Multi-Byte String, viz obrázek níže:


Zvolíme u nich možnost „Will be installed on local hard drive“







1. Nyní máme nainstalované PHP.
instalátor nám na konec konfiguračního souboru Apache **httpd.conf** přidal následující:

#BEGIN PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL

PHPIniDir "C:\Program Files\PHP\"

LoadModule php5\_module "C:\Program Files\PHP\php5apache2\_2.dll"

#END PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL

### Konfigurace PHP

1. I instalaci PHP je potřeba přinejmenším zkontrolovat
2. Nastavení PHP se provádí v souboru php.ini, který se po instalaci nachází zde:

„c:\Program Files\PHP\php.ini“

1. Nalezneme řádek s definicí adresáře s rozšířeními PHP a ujistíme se, že ukazuje na správný podadrsář s extensions :

extension\_dir="C:\Program Files\PHP\ext"

1. Na konci soboru jsou pak uvedena všechna povolená rozšíření. Překontrolujeme, zda je mezi nimi CURL (bez středníku na začátku – tedy nezakomentovaná):

[PHP\_CURL]

extension=php\_curl.dll

## Dokončení celé instalace

1. Teď je čas nahrát naší aplikaci do adresáře, kam jsme nasměrovali DocumentRoot pro náš VirtualHost. Tedy do:

c:\Web\WWW\schvalovani\

Ujistíme se, že v adresáři aplikace existují následující podadresáře:

c:\Web\WWW\schvalovani\www

c:\Web\WWW\schvalovani\app

c:\Web\WWW\schvalovani\log

c:\Web\WWW\schvalovani\temp

c:\Web\WWW\schvalovani\libs

**Pro adrsář „log“ a „temp“ musejí být nastavená maximální práva pro zápis.**

1. Nakonec je potřeba restartovat server Apache, aby načetl upravenou konfiguraci a modul pro PHP.
Učiníme tak přes aplikaci Monitor Apache Servers, která je nám po instalaci serveru Apache, schovala do hlavního panelu.

Rozklikneme tam ikonku 

V následujícím okně dáme Stop a Start (Měl by stačit Restart, ale tohle je Sichr ☺)



Po restartu (nebo spíše dvou) by se mělo ve StatusBaru okna objevit PHP/5.3.13



**POZOR: protože PHP přidává do proměnné prostředí PATH cestu ke svým knihovnám, je potřeba většinou restartovat celý server, aby správně fungovalo zejména rozšíření CURL pro PHP. Proto v tento okamžik raději server restartujte.**

## Nastavení webové aplikace

1. Nastavení samotné aplikace provedeme už celkem snadno
2. Jediné, co bývá nutné nastavit, je “kde běží webové služby”
3. Nastavení aplikace se provádí v souboru **custom.neon.**
	1. Soubor **custom.neon** ve výchozí instalaci neexistuje. Je zde pouze **\_custom.neon**
	2. Soubor **\_custom.neon** přejmenujte na **custom.neon**
	3. Souborem **custom.neon** je nyní možné přetížit všechny nastavitelné hodnoty ze souboru **config.neon**
	4. Needitujte proto **config.neon**, ale pouze **custom.neon** a udržujte v něm pouze ta nastavení, která jsou oproti **config.neon** různá.

Celá cesta je:

c:\Web\WWW\schvalovani\app\config\custom.neon

**Zde se provádí nastavení v sekcích:**

parameters – connections

1. zde je definice spojení (shodné s URL které bývá obsluhováno ABRA module pro Apache)
2. nastavení je vždy ve formátu:

**spojeni**: **“Popis spojení”**

**spojeni2**: **“Popis spojení 2”**

parameters – service

* 1. Je potřeba definovat **host**, na kterém běží webové služby (může to být stejný počítač, nebo I jiný). Patří sem tedy **IP adresa**, nebo **hostname** počítače, kde jste konfigurovali webové služby.
	2. Pomocí **path** je možné zadat I cestu, pokud služby neběží v rootu webu, to však nebývá nutné.
	3. Zároveň je možné zadat I **port** (pokud služby běží na jiném portu než 80)
	4. **web\_service\_name** je název webových služeb tak, jak jsou pojmenované v ABŘE

Z takového nastavení se pak skládá URL na webovou službu jako:

http://<host>/<path>:<port>/<spojeni>/<web\_service\_name>/

## Systémové URL

1. Pokud chcete aplikaci na chvíli vypnout (zobrazit maintenance stránku), aby ji uživatelé nemohli používat, zadejte:

http://<URL\_APLIKACE>/system/debug/mon

 poté maintenance vypnete touto URL:

http://<URL\_APLIKACE>/system/debug/moff

1. Je-li potřeba promazat Cache číselníků a Entit z ABRY, zavolejte:

http://<URL\_APLIKACE>/clearcache

1. Zapnutí Nette laděnky, která zobrazuje všechny dotazy do WS ABRA:

http://<URL\_APLIKACE>/system/debug/on

 analogicky k tomu vypnutí:

http://<URL\_APLIKACE>/system/debug/off