

Rozšíření logon - přihlašování a přesměrování

Uživatele potřebujeme nejen přihlašovat, ale po odeslání formuláře i nasměřovat na správné cílové stránky.

Základní situace

Máme sekci po přihlášení, která je běžným uživatelům skrytá. Typickým příkladem jsou:

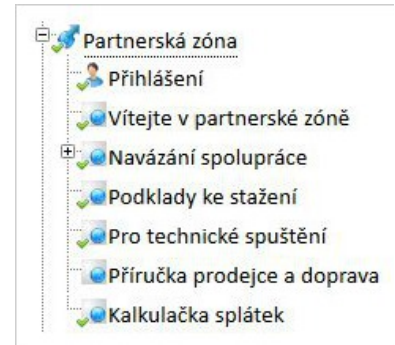
- partnerská zóna,
- zprávy pro VIP klienty,
- sekce pro zaměstnance aj.

Struktura dokumentů může mít strukturu jako na obrázku vpravo. Hyperlink míří na stránku "Vítejte...", nezobrazuje se v menu (nezobrazovat v navigaci). Případní návštěvníci přichází přes odkaz (ať už interní na našem webu nebo z externího zdroje).

Může proběhnout několik scénářů:

- **Nepřihlášený** uživatel zadá **správné** údaje pro přihlášení.
- **Nepřihlášený** uživatel zadá **nesprávné** údaje.
- **Přihlášený** uživatel **se odhlásí**.
- Nastane chyba aplikace.

Všechny situace umí vyřešit standardní obsah Přihlašování (Logon), který je v jádru Viva. Uživatele můžeme podle provedené akce přesměrovat na různé stránky. Nastavení je možné provést přes Vivo. Pokud není některá z adres zadaná, je dotyčný přesměrování zpátky na přihlašovací formulář a dostane zprávu o výsledku poslední akce.



Hlavní obsah	+
Verze obsahu	Verze 1.0: Přihlášení
URL pro úspěšné přihlášení	/partnerska-zona/vitejte-v-partnerske-zone/
URL pro chybné přihlášení	
URL pro odhlášení	/

Uživatel se vrací do skryté části webu

Samozřejmě pohyb uživatele po webu nekončí v přihlašovacím formuláři. Situaci řešíme tak, že **nepřihlášený uživatel** má k dispozici **odkaz na přihlášení** a na stejném místě má **přihlášený uživatel** k dispozici odkaz, který směřuje **na uvítací stránku** zabezpečené části webu.

Přihlášenému uživateli zobrazujeme položku v menu „Partnerská zóna“ s obsahem typu **Hyperlink**, který míří na **uvítací stránku**.

Nepřihlášenému uživateli v menu nezobrazujeme nic, má k dispozici pouze odkaz.

Zamčená sekce je viditelná v menu

Někdy může přijít požadavek, aby partnerská zóna byla viditelná v menu vždy. Uživatel se přihlásí, odejde do „normální“ části webu a pak se vrátí. Výše uvedená logika s takovou situací nepočítá. Nenastane chyba, uživatel dostane zprávu o tom, že je již přihlášen.

Výše popsaná situace není v zásadě špatná, ale není to user friendly. Uživatel přece ví, že je přihlášený a chce se dostat rovnou k obsahu. Teď bychom potřebovali aby hyperlink:

1. **přihlášeného** uživatele přesměroval na uvítací stránku/nějaký reálný obsah,
2. **nepřihlášeného** uživatele přesměroval na **přihlašovací formulář**.

To hyperlink neumí, ikdyž bychom ho to možná mohli naučit.

Každopádně, neměli logiku umístěnou v hyperlinku a pak ještě v přihlašování, je výhodnější rozšířit model Logon. Ten pak řeší všechny situace přihlašování, odhlásování a přesměrování uživatele. Případně ho v budoucnu budeme dále rozšiřovat podle nových požadavků.

Standardní model Logon ve Viva má 3 proměnné.

```
public $logon_url;  
public $error_url;  
public $logoff_url;
```

Doplníme model o další proměnnou, kterou pak využijeme při přesměrování.

```
public $logged_url;
```

Odpovídajícím způsobem model o statickou proměnnou, doplníme konstruktor a UI. Ve View nahradíme zprávu pro přihlášeného uživatele za metodu, která zajistí požadované přesměrování.

```
$this->redirect($this->content->logged_url);
```

V článku nebudeme vypisovat kompletní kód. Všechny součásti včetně *messages* jsou připravené ke stažení, viz nástěnka níže.

Ke stažení

- [logon extended .zip](#)
Modul, UI, View, messages v .zip balíku.
- [logon extended .tgz](#)
Modul, UI, View, messages v .tgz balíku.

TODO

1. Nebylo by dobré umístit komponentu do jádra viva?
2. Sepsat variantu, kdy je více obsahu - *param->param->main* nebo *param->2* + množstvi param.

Související články

- [Release process of Vivo release](#)
- [getSentry Service | nastveni modulu raven](#)
- [Specifický title stránky](#)
- [Nasazení Vivo 2 na NGINX+HHVM](#)
- [Zprovoznění Vivo pod open_basedir](#)