

Bottle -- příklad

V tomto příkladu se pokusíme vytvořit malou aplikaci umožňující psát jednoduché poznámky.

Databáze

Nejprve je třeba vytvořit v databázovém serveru **uživatele** (pokud už není vytvořen) a také **databázi**, do které budeme ukládat data. Já nazvu uživatele `marek` a databázi `notes`.

```
-- uživatel
CREATE USER 'marek'@'localhost' IDENTIFIED BY 'tojetajny';
GRANT USAGE ON *.* TO 'marek'@'localhost' IDENTIFIED BY 'tojetajny'
WITH
    MAX_QUERIES_PER_HOUR 0
    MAX_CONNECTIONS_PER_HOUR 0
    MAX_UPDATES_PER_HOUR 0
    MAX_USER_CONNECTIONS 0;

-- databáze
CREATE DATABASE IF NOT EXISTS `notes`
    CHARACTER SET utf8
    COLLATE utf8_general_ci;
GRANT ALL PRIVILEGES ON `notes`.* TO 'marek'@'localhost';
```

`--> [stáhnout](#)

Dále vytvoříme v databázi tabulku `poznamky`, do které budeme ukládat jednotlivé záznamy. Tabulka bude obsahovat následující sloupce:

id

Jedinečný identifikátor řádku: Bude typu `INT` s `AUTO_INCREMENT` a nastavíme ho jako primární klíč

titulek

Titulek poznámky:

obsah

Samotný obsah poznámky:

```
CREATE TABLE poznamky (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    titulek VARCHAR(100),
    obsah VARCHAR(10000)
);
```

`--> [stáhnout](#)

Testovací data

Dále by bylo vhodné vložit do tabulky dva řádky pro účely testování:

```
INSERT INTO notes.poznamky (`titulek`, `obsah`) VALUES
    ('Miluju Python',
     'Python je nejlepší programovací jazyk na světě. Svůj život už si bez něj nedokážu představit.'),
    ('To je jedno',
     'Je úplně jedno, co sem napíšu.'
```

```
);
```

`--> [stáhnout](#)

URL

Jednotlivé akce budou probíhat na jim přidělených URL adresách:

/

seznam poznámek

/poznamka/*ID*

jednotlivé poznámky (ID je její číselný identifikátor)

/edi t/*ID*

editace poznámky

Jednotlivým adresám odpovídají také šablony, které se budou vykreslovat.

Šablony

Budeme potřebovat tři základní šablony: pro seznam poznámek, pro každou zobrazení poznámky a pro editaci poznámky. Kostra HTML stánky může vypadat asi takto:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Notes</title>

  <style type="text/css">
    body {
      width: 80%;
      margin-left: auto;
      margin-right: auto;
    }
    #hlavni {
      color: green;
    }
  </style>

</head>
<body>

  {{sem budu dávat obsah stránky}}

</body>
</html>
```

`--> [stáhnout](#)

Tato základní kostra by se měla rozpadnou na tři šablony:

1. seznam.tpl bude vypisovat seznam poznámek. Myslím, že by bylo vhodné předat seznam poznámek do šablony opravdu jako seznam. Každá položka seznamu bude obsahovat další seznam s id a ti tulkem:

```
<h1><a href="/">Moje poznámky</a></h1> <hr />
<ol>
```

```

% for id,titulek in seznam_poznamek:
    <li>
        <a href="poznamka/{{id}}">{{titulek}}</a>
        :-
        <a href="edit/{{id}}">edituj</a>
    </li>
% end
</ol>
<hr /> <h1>Přidej poznámku</h1>
<form method="post">
    <p>
        <input type="text" name="titulek" placeholder="Nový titulek" />
        <input type="submit" name="odeslat" value="Odeslat" />
    </p>
</form>

```

`--> [stáhnout](#)

2. poznamka.tpl bude zobrazovat konkrétní poznámku:

```

<h1><a href="/">Moje poznámky</a></h1> <hr />
<h1 id="hlavni">{{titulek}}</h1> <hr />
{{! obsah}}
<hr />
<p><a href="/edit/{{id}}">edituj</a></p>

```

`--> [stáhnout](#)

3. edi t. tpl bude sloužit k editaci poznámky:

```

<h1><a href="/">Moje poznámky</a></h1> <hr />
<h1>{{titulek}}</h1>
<hr />
<form method="post">
    <p>
        <input type="hidden" name="id" value="{{id}}">
        <textarea rows="30" cols="80" name="obsah">{{obsah}}</textarea>
    </p>
    <p>
        <input type="submit" value="odeslat">
    </p>
</form>

```

`--> [stáhnout](#)

Bottle

Seznam poznámek

Základ pro náš zdrojový kód může vypadat například takto:

```

#!/usr/bin/python
# -*- coding: utf8 -*-
# Soubor: notes.py
# Datum: 23. 03. 2014 23:00
# Autor: Marek Nožka, nozka <@> spseol <d.t> cz
# Licence: GNU/GPL
#####
"""
Webový notýsek
"""
from __future__ import unicode_literals
from bottle import run, get, post, redirect, template
import MySQLdb

```

```

# passwd='eek6eud1',
db = MySQLdb.connect(host='localhost',
                    user='marek',
                    passwd='tojetajny',
                    charset='utf8',
                    db='notes')

@get('/')
@get('/seznam')
def seznam():
    cursor = db.cursor()
    cursor.execute('SELECT id, titulek FROM poznamky')
    data = cursor.fetchall()
    cursor.close()
    return template('seznam', seznam_poznamek=data)

#####
run(host='0.0.0.0', port=8090, reloader=True)

db.close()

```

`--> [stáhnout](#)

Tím by měla být obstarán seznam poznámek.

V dolní části stránky je formulář pro vytvoření nové poznámky. Tento formulář je třeba "obsloužit". Příjem dat pro vytvoření nové poznámky se děje pomocí metody POST.

```

@post('/')
@post('/seznam')
def pridej():
    titulek = request.forms.titulek
    cursor = db.cursor()
    cursor.execute('INSERT INTO poznamky (titulek, obsah) '
                  "VALUES ('{}','NOVE')".format(titulek))
    db.commit()
    cursor.close()
    redirect('/')

```

`--> [stáhnout](#)

Dále je třeba vytvořit kód pro zobrazení a editaci poznámky.

Zobrazení poznámky

Pro jednoduché vykreslení a editaci použijeme jazyk Markdown, konkrétně tedy python knihovnu markdown. Tento jednoduchý značkovací jazyk umožňuje velice přehlednou editaci a převod do HTML. Samotný převod se děje pomocí funkce `markdown.markdown()`.

Ukázka značkovacího jazyka Markdown

```

Nadpis
=====

* toto
* je
* seznam

Menší naspis
-----

```

```

1. toto se
2. čísluje

Sem píšu tenějaká text
a prázdný řádek

udělá nový odstavec. Ještě odkaz na
[dokumentaci](http://daringfireball.net/projects/markdown/syntax)

<p>Dle libosti <em>můžu používat</em> html</p>

nebo obrázek
![[Alt text]](image/600px/svg_to_png/192053/remington_typewriter.png)

```

`--> [stáhnout](#)

O samotné vykreslení poznámek by se mohl starat následující kód:

```

import markdown

@get('/poznamka/<id:int>')
def ukaz(id):
    cursor = db.cursor()
    cursor.execute('SELECT titulek, obsah FROM poznamky '
                  'WHERE id = {}'.format(id)
                  )
    titulek, obsah = cursor.fetchone()
    cursor.close()
    return template('poznamka', id=id, titulek=titulek,
                  obsah=markdown.markdown(obsah))

```

`--> [stáhnout](#)

Úprava poznámky

Nejprve potřebujeme poznámku vytáhnou z databáze a vložit ji do šablony. Tento kód se tedy velice podobá kódu pro zobrazení poznámky.

```

@get('/edit/<id:int>')
def edit(id):
    cursor = db.cursor()
    cursor.execute('SELECT titulek, obsah FROM poznamky '
                  'WHERE id = {}'.format(id)
                  )
    titulek, obsah = cursor.fetchone()
    cursor.close()
    return template('edit', id=id, titulek=titulek, obsah=obsah)

```

`--> [stáhnout](#)

Editovaná poznámka přichází do naší aplikace metodou POST. Poté co aktualizovaný text poznámky vložíme do databáze poznámku zobrazíme. To se děje přesměrováním požadavku na stránku /poznamka/ID.

```

@post('/edit/<id:int>')
def update(id):
    obsah = request.forms.obsah
    cursor = db.cursor()
    cursor.execute("UPDATE poznamky SET obsah='{}' "

```

```
        "WHERE id='{1}'".format(obsah, id))
db.commit()
cursor.close()
redirect('/poznamka/{1}'.format(id))
```

`--> [stáhnout](#)

Závěrečný úkol

Přidejte do šablon a do aplikace kód, který zajistí **vymazání** dané **poznáky**. SQL může vypadat asi takto:

```
DELETE FROM poznamky WHERE id='cislo';
```

`--> [stáhnout](#)



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PSK3-19

Název školy:	Vyšší odborná škola a Střední průmyslová škola, Božetěchova 3
Autor:	Ing. Marek Nožka
Anotace:	web framework Bottle.py
Vzdělávací oblast:	Informační a komunikační technologie
Předmět:	Počítačové sítě a komunikační technika (PSK)
Tematická oblast:	Operační systém Linux/Unix
Výsledky vzdělávání:	Žák vytvoří jednoduchou webovou aplikaci pomocí Bottle.py
Klíčová slova:	Linux, Unix, Apache, Python, Bottle.py, web framework
Druh učebního materiálu:	Online návod pro samostatnou práci
Typ vzdělávání:	Střední vzdělávání, 4. ročník, technické lyceum
Ověřeno:	VOŠ a SPŠE Olomouc; Třída: 4L
Zdroj:	Vlastní poznámky, Vilém Vychodil: Linux Příručka českého uživatele