



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PSK3-17

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 vzdělávací materiál
Typ vzdělávání:	Střední vzdělávání, 4. ročník, technické lyceum
Ověřeno:	VOŠ a SPŠE Olomouc; Třída: 4L
Zdroj:	Vlastní poznámky, Vilém Vychodil: Linux Příručka českého uživatele

Python jako motor webu -- Bottle

I

V dnešní době je nespočet technologií, které lze použít pro vytváření dynamických webových stránek. Jednou z možností je použít programovací jazyk Python. Vývoj webových aplikací v holém Pythonu by ale byl příliš složitý a pracný. Proto se většinou používají frameworky, které celou věc zjednoduší a umožňují aby se programátor soustředil na to podstatné.

Pro Python existuje web-frameworků celá řada. Za zmínku jistě stojí

- Django; Django
- Bottle
- Flask
- Pyramid

Pokud si chcete udělat přehled, jaké možnosti python má určitě si přečtete článek Znám jen PHP. Jak napíšu webovou aplikaci v Pythonu?.

Začínáme s Bottle

My se seznámíme s frameworkem Bottle. Ten je maximálně jednoduchý a k ničemu nás nenutí. První, malá, nic moc nedělající stránka/aplikace může vypadat například takto:

```
#!/usr/bin/python
# -*- coding: utf8 -*-
# Soubor: hello.py
# Datum: 07.02.2014 10:26
# Autor: Marek Nožka, nozka <@t> spseol <d.t> cz
# Licence: GNU/GPL
# Úloha: Hello World Bottle
#####

from bottle import route, run

@route('/')
@route('/ahoj')
def hello():
    return """<h1>Hurá Hurá! </h1>
<p>Funguje to! </p>"""

run(host='localhost', port=8090, debug=True, reloader=True)
```

`-> stáhnout

Vytvořený soubor uložíme na disk a spustíme. Spustí se tak malý vývojový web-server, který naslouchá na portu 8090.

```
$ python hello.py
```

Nyní se na náš první výtvar můžeme podívat ve webovém prohlížeči na adrese http://localhost:8090/. Pokud v souboru uděláme změny okamžitě se promítnou bez toho aniž bychom aplikaci (web-server) ukončovali a znovu spouštěli.

Úkol

1. Udělejte ve zdrojovém kódu syntaktickou chybu. Například `a=8+' 8'` a podívejte se jak na to spuštěná aplikace (web-server) zareaguje.
2. Otestujte adresu http://localhost:8090/nejaka_blbost
3. Otestujte adresu http://localhost:8090/ahoj

Routing

Každé adrese URL lze přiřadit kód, který se má provést.

```
1 #!/usr/bin/python
2 # -*- coding: utf8 -*-
3 # Soubor: name.py
4 # Datum: 07.02.2014 11:12
5 # Autor: Marek Nožka, nozka <@t> spseol <d.t> cz
6 # Licence: GNU/GPL
7 # Úloha: Hello World Bottle +
8 #####
9
10 from bottle import route, run
```

```

11 import datetime
12
13 @route('/')
14 @route('/ahoj')
15 def funguje():
16     return """<h1>Hurá Hurá! </h1>
17     <p>Funguje to! </p>"""
18
19 @route('/cas')
20 def cas():
21     q='<h1>Aktuální čas</h1>'
22
23     td=datetime.datetime.today()
24
25     q+="

Aktuální datum: {}, {}, {}</p>".format(td.day, td.month, td.year)
26     q+="

Aktuální čas: {}: {}: {}</p>".format(td.hour, td.minute, td.second)
27
28     return q
29
30 @route('/ahoj/<name>')
31 def ahoj(name):
32     return "<h1>Ahoj {}</h1><p>Funguje mi to! Hurá! Hurááááá.</p>".format(name)
33
34 @route('/nasobeni/<a>/<b>')
35 @route('/<a>*<b>')
36 def nasobeni(a,b):
37     return str(float(a)*float(b))
38
39 @route('/mocnina/<a: float>/<b: int>')
40 @route('/<a: float>^<b: int>')
41 def mocnina(a,b):
42     vysledek=a**b
43     return str(vysledek)
44
45 #####
46 run(host='localhost', port=8090, debug=True, reloader=True)


```

`--> [stáhnout](#)

Úkol

Vyzkoušejte si následující [URL](#) a prostudujte zdrojový kód, který je obhospodařuje.

1. <http://localhost:8090/ahoj>
2. <http://localhost:8090/ahoj/Karle>
3. <http://localhost:8090/ahoj/Tondo>
4. <http://localhost:8090/cas>
5. <http://localhost:8090/cas/blbost>
6. <http://localhost:8090/nasobeni/4/7>
7. http://localhost:8090/4*7
8. http://localhost:8090/4*abc
9. <http://localhost:8090/4^abc>
10. <http://localhost:8090/mocnina/4/abc>
11. <http://localhost:8090/4^3>
12. <http://localhost:8090/mocnina/4/3>

Více o routingu se dozvíte v dokumentaci:
<http://bottlepy.org/docs/dev/routing.html>

Zpracování dat z formulářů

Existuje více možností jak přistupovat k datům z formulářů. Záleží na tom, která metoda je použita (GET, POST). Více informací naleznete v

dokumentaci:

- [HTTP Request Methods](#)
- [HTML <form> Handling](#)

Atribut GET POST File Uploads

request.query	yes	no	no
request.forms	no	yes	no
request.files	no	no	yes
request.params	yes	yes	no
request.GET	yes	no	no
request.POST	no	yes	yes

```
1 #!/usr/bin/python
2 # -*- coding: utf8 -*-
3 # Soubor: forms.py
4 # Datum: 07.02.2014 11:59
5 # Autor: Marek Nožka, nozka <@t> spseol <d.t> cz
6 # Licence: GNU/GPL
7 # Úloha: Bottle a formuláře
8 #####
9
10 from bottle import run, route, get, post, request, redirect
11 import subprocess
12
13 # přesměrování
14 @route('/')
15 def redir():
16     redirect("/login")
17
18 @get('/login') # nebo @route('/login')
19 def login():
20     print '-----'
21     print request.method
22     print '-----'
23     return '''
24     <form action="/login" method="post">
25         Username: <input name="username" type="text" />
26         Password: <input name="password" type="password" />
27         <input value="Login" type="submit" />
28     </form>
29     '''
30
31 @post('/login') # nebo @route('/login', method='POST')
32 def do_login():
33     print '-----'
34     print request.method
35     # print dir(request.params)
36     for klic in request.params.keys():
37         print klic, '->', request.params[klic]
38     print '-----'
39     username = request.forms.get('username')
40     password = request.forms.get('password')
41     if 'abcd' in password:
42         return ""
43     <h1>{0}</h1>
44     <p>Hurá! Huráááá. Uhodl jsi</p>
45     """, format(username)
46     else:
47         return "<h1>Ne! Ne! Ne!!!</h1><p>Špatné heslo :-P</p>"
48
49
50 @route('/pwgen')
51 def pwgen():
52     h='<h1>Zapamatovatelné heslo?</h1>\n'
53     form='''<form method="get">
54 <p>
55     Délka hesla (od 5 do 40): <input name="length" type="text" /><br />
```

```

56     <input value="goo" type="submit" />
57 </p>
58 </form>
59 '''
60
61     length = request.query.length
62     if length :
63         try :
64             length=int(length)
65             length= 8 if length<5 or length>40 else length
66         except :
67             length=8
68         pswd= subprocess.check_output(['pwgen', '-Ccn', str(length), 1])
69         return h+'\n<p>Délka '+str(length)+' </p>\n<pre style="font-size:large;">\n'+pswd+' </pre>\n'+form
70     else :
71         return h+form
72
73 #####
74 run(host='localhost', port=8090, debug=True, reloader=True)

```

`--> stáhnout

Úkol

Opět vyzkoušejte jednotlivá URL a prozkoumejte zdrojový kód k nim příslušející.

- <http://localhost:8090/>
- <http://localhost:8090/login>
- <http://localhost:8090/pwgen>
- <http://localhost:8090/pwgen?length=12>
- <http://localhost:8090/pwgen?length=nejakaBlbost>

Odkazy

- <http://bottlepy.org/>
- <http://www.zdrojak.cz/clanky/jak-napisu-webovou-aplikaci-v-pythonu/>
- <http://www.zdrojak.cz/clanky/jak-napisu-webovou-aplikaci-v-pythonu-2/>
- <http://www.zdrojak.cz/serialy/webovy-framework-web-py/>
- <http://www.jakpsatweb.cz/server/http-protokol.html>

*

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Další odkazy na dokumentaci

- [Dynamic Routes](#)
- [HTTP Request Methods](#)
- [HTML <form> Handling](#)
- [Routing Static Files](#)
- [Request Routing](#)
- [Templates](#)
- [Error Pages](#)
- [Generating content](#)

- Static Files
- HTTP Errors and Redirects
- The Response Object
- Cookies
- Request Data
- File uploads
- JSON Content
- Plugins
- Development