

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Лабораторная работа №6  
по Дисциплине "Технологии программирования"

Выполнил: Юсуюмбели В. И.  
Плохотнюк В. С.  
Толстухин М.С.  
Группа: Р3411  
Преподаватель: Оголюк А. А.

Санкт-Петербург  
27 ноября 2019 г.

```

1  from http.server import BaseHTTPRequestHandler, HTTPServer
2  from urllib.parse import parse_qs
3  from urllib.parse import urlparse
4
5  import feedparser
6  from bson.json_util import dumps, loads
7  from bson.objectid import ObjectId
8  from pymongo import MongoClient
9
10
11 class RecordsHTTPServer(BaseHTTPRequestHandler):
12     def do_GET(self, rsss, db):
13         params = parse_qs(urlparse(self.path).query)
14         page = int(params['page'][0])
15         size = int(params['size'][0])
16         rss_id = params['rss'][0]
17         rss = rsss.find_one({'_id': ObjectId(rss_id)})
18         response = '{"records":{0}, "pagination":{1}}'
19         if rss is None:
20             print('rss {} not found'.format(rss_id))
21         else:
22             records = db['{}_records'.format(rss_id)]
23             all_records = list(records.find())
24             for record in all_records:
25                 del record['_id']
26             total = len(all_records)
27             pagination = '{"page":{0}, "size":{1}, "total":{2}}'.format(page, size, total)
28             self.send_response(200)
29             self.send_header('Content-type', 'application/json')
30             self.end_headers()
31             if page * size > total:
32                 self.wfile.write(bytes(response.format(dumps(all_records), pagination),
33                 ↪ encoding='utf-8'))
34             else:
35                 self.wfile.write(
36                     bytes(response.format(dumps(all_records[(page * size):(page * size +
37                     ↪ size))], pagination),
38                     encoding='utf-8'))
39
40     def do_RELOAD(self, rsss, db):
41         rss_id = parse_qs(urlparse(self.path).query)['rss'][0]
42         rss = rsss.find_one({'_id': ObjectId(rss_id)})
43         if rss is None:
44             print('rss {} not found'.format(rss_id))
45         else:
46             records = db['{}_records'.format(rss_id)]
47             new_records = feedparser.parse(rss['url'])['entries']
48             for record in new_records:
49                 if 'id' not in record:
50                     record['id'] = record['link']
51                 if records.find_one({'id': record['id']}) is None:
52                     records.insert_one(
53                         {"id": record['id'], "link": record['link'], "published":
54                         ↪ record['published'],
55                         "summary": record['summary']})
56             self.send_response(200)
57             self.send_header('Content-type', 'application/json')
58             self.end_headers()
59             self.wfile.write(bytes("{}", encoding='utf-8'))
60
61 class TypesHTTPServer(RecordsHTTPServer):
62     def do_GET(self):

```

```

61 client = MongoClient('localhost', 27017)
62 db = client['rss-scroller']
63 rsss = db['rsss']
64
65 try:
66     if self.path.startswith('/get-rss-list'):
67         self.send_response(200)
68         self.send_header('Content-type', 'application/json')
69         self.end_headers()
70         self.wfile.write(bytes(dumps(rsss.find()), encoding='utf-8'))
71
72     if self.path.startswith('/get-records'):
73         RecordsHTTPServer.do_GET(self, rsss, db)
74
75     if self.path.startswith('/reload-rss'):
76         RecordsHTTPServer.do_RELOAD(self, rsss, db)
77
78     return
79 except IOError:
80     self.send_error(404, 'Not Found: %s' % self.path)
81
82 def do_POST(self):
83     client = MongoClient('localhost', 27017)
84     db = client['rss-scroller']
85     rsss = db['rsss']
86     if self.path.startswith('/add-rss'):
87         self.add_rss(rsss,
88             ↪ loads(self.rfile.read(int(self.headers['Content-Length'])).decode("utf-8")))
89         self.send_response(200)
90         self.send_header('Content-type', 'application/json')
91         self.end_headers()
92         self.wfile.write(bytes(dumps(rsss.find()), encoding='utf-8'))
93
94 def add_rss(self, _rsss, rss):
95     if _rsss.find_one({'url': rss['url'], 'title': rss['title']}) is None:
96         _rsss.insert_one(rss)
97
98 def run():
99     httpd = HTTPServer(('192.168.43.94', 8080), TypesHTTPServer)
100     print('Application successfully started')
101     httpd.serve_forever()
102
103
104 if __name__ == '__main__':
105     run()

```