

django-piston

RESTfulowe API w 5 minut

Konrad Hałas
konradhalas.blogspot.com
[@konradhalas](https://twitter.com/konradhalas)

Co to jest REST?

- REST - representational state transfer
- wzorzec architektoniczny
- sposób manipulowania zasobami
- wykorzystuje protokół HTTP
- wykorzystywany jako:
 - API dla zewnętrznych serwisów
 - API dla zapytań AJAX

Co to jest REST?

- URL do wskazania zasobu

`http://example.com/posts/1`

- metody HTTP do manipulacji danymi

`GET, POST, PUT, DELETE`

- popularne formaty danych

`XML, JSON, YAML`

```
api.sejmometr.pl/Donald-Tusk/info - Chromium
api.sejmometr.pl/Donal x
api.sejmometr.pl/Donald-Tusk/info
{
  id: "Donald-Tusk",
  nazwa: "Donald Tusk",
  plec: "M",
  klub_nazwa: "Platforma Obywatelska",
  klub_id: "P0",
  wiek: "54",
  nid: "396",
  data_wygasniecia: "0000-00-00",
  okreg_nr: "19",
  wojewodztwo_id: "7",
  ilosc_oswiadczen: "0",
  procent_nieobecnosci: "76.0",
  ilosc_slow: "110059",
  ilosc_wypowiedzi: "56",
  srednia_dlugosc_wypowiedzi: "1965",
  wyksztalcenie: "4",
  zawod: "historyk",
  ilosc_projektow: "1",
  ilosc_interpelacji: "0",
  liczba_glosow: "534241",
  data_wybrania: "2007-10-21",
  data_slubowania: "0000-00-00",
  data_urodzenia: "1957-04-22",
  miejsce_urodzenia: "Gdańsk",
  szkoła: "Uniwersytet Gdański,
  Wydział Humanistyczny
  - mgr historii",
  stan_cywilny: "2"
}
```



miniframework do tworzenia RESTfulowego API w Django

bitbucket.org/jespern/django-piston

django-piston

- współpracuje z wewnętrznymi mechanizmami Django
 - modele
 - formularze
- uwierzytelnianie:
 - HttpBasic
 - OAuth
- wspierane typy danych:
 - JSON
 - YAML
 - Python Pickle
 - XML
- throttling - ograniczenie liczby zapytań

przykład v1.0

```
1 from django.db import models
2 from django.contrib.auth.models import User
3
4 class MicroPost(models.Model):
5     author = models.ForeignKey(User)
6     text = models.CharField(max_length=140)
7     date = models.DateTimeField(auto_now_add=True)
```

microlog/blog/models.py


```
1 from django.conf.urls.defaults import patterns, include, url
2
3 urlpatterns = patterns('',
4     url(r'^api/', include('microlog.api.urls')),
5 )
```

microlog/urls.py

```
1 from django.conf.urls.defaults import patterns, url
2 from piston.resource import Resource
3 from microlog.api.handlers import MicroPostHandler
4
5 microposts = Resource(handler=MicroPostHandler)
6
7 urlpatterns = patterns('',
8     url(r'^posts/$', microposts),
9 )
```

microlog/api/urls.py

```
1 from piston.handler import BaseHandler
2 from microlog.blog.models import MicroPost
3
4 class MicroPostHandler(BaseHandler):
5     model = MicroPost
```

microlog/api/handlers.py

przykład v2.0

```
1 from django.conf.urls.defaults import patterns, url
2 from piston.resource import Resource
3 from microlog.api.handlers import MicroPostHandler
4
5 microposts = Resource(handler=MicroPostHandler)
6
7 urlpatterns = patterns('',
8     url(r'^posts/$', microposts),
9     url(r'^posts/(?P<id>\d+)$', microposts),
10 )
```

microlog/api/urls.py

```
1 from piston.handler import BaseHandler
2 from microlog.blog.models import MicroPost
3
4 class MicroPostHandler(BaseHandler):
5     model = MicroPost
6     fields = ('date', 'text', ('author', ('username',)))
```

microlog/api/handlers.py

przykład v3.0

HTTP vs. BaseHandler

HTTP	BaseHandler
GET	read
POST	create
PUT	update
DELETE	delete

```
1 from piston.handler import BaseHandler
2 from piston.utils import throttle, rc
3 from microlog.blog.models import MicroPost
4
5 class MicroPostHandler(BaseHandler):
6     model = MicroPost
7     fields = ('date', 'text', ('author', ('username',)))
8
9     @throttle(5, 10)
10    def read(self, request, id=None):
11        if id:
12            try:
13                return MicroPost.objects.get(id=id)
14            except MicroPost.DoesNotExist:
15                return rc.NOT_FOUND
16
17        return MicroPost.objects.order_by('-date')[:5]
```

Co poza tym?

- walidacja z wykorzystaniem formularzy
- metody zasobów
- uwierzytelnianie
- anonimowe zasoby
- generowanie dokumentacji
- strumieniowanie

Dziękuję!