

 無料電子ブック

学習

tastypie

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#tastypie

.....	1
<b>1: tastypie</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
<b>2: Tastypie</b> .....	<b>3</b>
.....	3
Examples.....	3
.....	3
Tastypie.....	3
Tastypie.....	4
Tastypie.....	5
<b>myapp / api.py</b> .....	<b>6</b>
<b>urls.py</b> .....	<b>6</b>
.....	7
<b>myapp / api.py</b> .....	<b>7</b>
.....	8

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [tastypie](#)

It is an unofficial and free tastypie ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official tastypie.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# 1: tastypieをいめる

このセクションでは、tastypieのと、がそれをいたいについてします。

それはまた、ののきなをし、するトピックにリンクするがあります。tastypieのドキュメンテーションはしいので、これらのトピックのバージョンをするがあります。

## Examples

インストールまたはセットアップ

タパスタイプのまたはインストールにするな。

オンラインでtastypieをいめるをむ <https://riptutorial.com/ja/tastypie/topic/9618/tastypieをいめる>

## 2: Tastypieによるコーディング

き

TastypieはDjangoのWebサービスAPIフレームワークです。これは、RESTスタイルのインターフェイスをするためののですが、にカスタマイズなをします。Tastypieをすると、モデルをにできますが、するをにできるため、なだけデータベースをできます。Tastypieは、ORMのデータソースとのものにします。したがって、TastypieはORMおよびNon-ORMデータベースでできます。

### Examples

クイックスタート

1. tastypieをINSTALLED\_APPSにします。
2. のinit.pyをしてアプリケーションにapiディレクトリをします。
3. <my\_app>/api/resources.pyファイルをし、のファイルをします。

```
from tastypie.resources import ModelResource
from my_app.models import MyModel

class MyModelResource(ModelResource):
    class Meta:
        queryset = MyModel.objects.all()
        allowed_methods = ['get']
```

rootのURLconfにのコードをしますコードの。

```
from django.conf.urls import url, include
from tastypie.api import Api
from my_app.api.resources import MyModelResource

v1_api = Api(api_name='v1')
v1_api.register(MyModelResource())

urlpatterns = [
    # ...more URLconf bits here...
    # Then add:
    url(r'^api/', include(v1_api.urls)),
]
```

### Tastypieのインストール

TastypieはPythonパッケージ、つまりpipでインストールすることも、Githubからコードをチェックアウトすることもできます

1. pipインストールdjango-tastypie

## 2. Githubからのチェックアウト

### なぜTastypieですか

DjangoにはにもAPIフレームワークがあります。なオプションをし、でするがあります。つまり、ここにはののがいくつかあります。

- あなたはRESTfulでHTTPをうまくったAPIがです。
- あなたはいをサポートしたい。
- をしくうためののシリアライザをするはありません。
- しのをい、にがあり、のドメインにうまくマップできるAPIフレームワークがです。
- JSONとにわれるXMLシリアルがですYAMLもそこにあります。

DjangoをとってAPIをするには、くのがあります。なオプションは、 **Tastypie**と **Django Rest Framework DRF**です。

まともな**API**フレームワークをるには

これらの

1. ページネーション
2. によるデータの
3. クエリーセットにうメタデータの
4. APIディスカバリー
5. なHTTPレスポンス
6. キャッシング
- 7.
8. スロットリング
- 9.
- 10.

なAPIフレームワークにもがです。

11. コードのにいテストカバレッジ
12. まともなパフォーマンス
13. ドキュメンテーション
14. フレームワークをめ、サポートするためのなコミュニティ

これらのをった、では、のあるAPIフレームワークは2つしかありませんユーザービューにします、 **django-tastypie**と **django-rest-framework**です。

どちらがいいですか **django-tastypie**または**django-rest-framework**

はらがだとう。あなたはどちらかでってくことはできません。それらののにあるとコミュニティはアクティブであり、コードはされており、テストされています。そして、ここにのについての

のなえがあります

## Tastypie

- にめることができ、なをするOOBそのまま
- ほとんどの、CBVやFormsなどのなDjangoのコンセプトをうことはありません
- みやすいコードとないマジック
- あなたのモデルがORMのは、それをめてください。
- には、Djangoにうものではありませんよくえているpythonとdjangoのはくなります
- APIをカスタマイズするにはししいでしょう
- O-Authなし

## DRF

- のdjangoにつてください。あなたがdjangoをつていて、CBVににれていれば、いなくフォームなどがそれになります
- ModelViewSetをして、すぐにできるRESTをします。に、CustomSerializer、APIView、GenericViewsなどをしてカスタマイズするためのれたコントロールをします。
- よりい。
- カスタムクラスをにできます。
- のライブラリやOAuthとさせることはにうまくいており、なことです。
- DJANGO-REST-AUTHはAuth / SocialAuthentication / RegistrationのLIBRARYにするがあります。
- あなたがDjangoをよくつていないなら、これにはかないでください。
- いくつかのはにしいをする。
- それはそのかなりなジャンゴのCBVのにかれているからです。
- ながあります。

## Tastypie

Tastypieはなアプリケーションですつまり、のコードのみにし、RESTスタイルのAPIのみをすることにをいています。そのアプリケーションのソースをすることなく、のアプリケーションにAPIをするのにしています。

すべてのニーズがじではないので、Tastypieは、それがどのようにするかをきしたりしたりするためのフックをたくさんしています。

たとえば、なブログアプリケーションにAPIをします。

のは、あなたの**INSTALLED\_APPS**に '**tastypie**' をすることです。Tastypieには2つのモデルしか

ありませんが、いかなので、これはにはではありません。

ここに **myapp / models.py** があります

```
from tastypie.utils.timezone import now
from django.contrib.auth.models import User
from django.db import models
from django.utils.text import slugify

class Entry(models.Model):
    user = models.ForeignKey(User)
    pub_date = models.DateTimeField(default=now)
    title = models.CharField(max_length=200)
    slug = models.SlugField(null=True, blank=True)
    body = models.TextField()
```

リソースの

RESTスタイルのアーキテクチャはリソースについてりますので、Tastypieとくほどするにはリソースクラスをするがあります。なアプリケーションでは、 **myapp / api.py** でこれらのファイルをしますが、アプリケーションのどこにでもくことができます。

---

## myapp / api.py

```
from tastypie.resources import ModelResource
from myapp.models import Entry

class EntryResource(ModelResource):
    class Meta:
        queryset = Entry.objects.all()
        resource_name = 'entry'
```

RESTではすべてがリソースであり、オブジェクトがリソースとしてされていることをします。したがって、Tastypie REST APIをするには、リソースクラスをするがあります。リソースをするには、ModelResourceクラスをサブクラスするがあります。このEntryResourceクラスは、Entryモデルのすべてのリレーショナルフィールドをチェックし、のApiフィールドをします。これは、DjangoフォームのModelFormのようにします。

リソースをフックアップするリソースをするAfrer、Djangoに、EntryResourceをURLにフックしてリソースをするがあることをえるがあります。

---

## urls.py

```
from django.conf.urls import url, include
from myapp.api import EntryResource
```

```
entry_resource = EntryResource()

urlpatterns = [
    url(r'^blog/', include('myapp.urls')),
    url(r'^api/', include(entry_resource.urls)),
]
```

ここで**localhost8000 / api / entry /**をチェックすると、APIのがられます。

メタクラスのリソースはです。しない、にリソースのすべてのインスタンスをし、をに、クラスをオフにされます。だから、EntryResourceだけのエントリになる。URLで、々は**./entry**されたです/

タスティピー

までたちはリクエストをけようとしてしました。のHTTPをすがあるは、*curl*や*postman*のようなAPIテストツールをしてください

POST / PUT / DELETEをリソースにしようとする、**「401 Unauthorized」**エラーがします。のため、Tastypieには、クラスがReadOnlyAuthorizationにされています。これにより、Webですることはですが、POST / PUT / DELETEをすることはできません。それらをにしましょう。

---

## myapp / api.py

```
from tastypie.authorization import Authorization
from tastypie.resources import ModelResource
from myapp.models import Entry

class EntryResource(ModelResource):
    class Meta:
        queryset = Entry.objects.all()
        resource_name = 'entry'
        authorization = Authorization()
```

オンラインでTastypieによるコーディングをむ

<https://riptutorial.com/ja/tastypie/topic/10640/tastypieによるコーディング>

## クレジット

S. No		Contributors
1	tastypieをいめる	<a href="#">Community</a>
2	Tastypieによるコーディング	<a href="#">AKHIL MATHEW</a>