



Projeto Final 1.0

Prof. Paulo Junior

AMONTADA
VALLEY
TECNOLOGIA E EMPREENDEDORISMO

AV RSS

O que é RSS?

RSS é a sigla em inglês para Rich Site Summary, ou seja, uma forma simplificada de apresentar o conteúdo de um site. Um documento RSS é feito na linguagem XML/JSON e geralmente exibe o grande volume de informações existente em uma página na internet de forma resumida. Pela característica de alimentar de notícias, os documentos RSS também são chamados de Feeds.

AV RSS

Um RSS trabalha com:

- `channel(canal)`
- `item(etiquetas)`

Dentro do `channel` você terá um `item` que por sua vez apresentará as notícias/informações do site.

TECNOLOGIA E EMPREENDEDORISMO

AV RSS

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>RSS Title</title>
  <description>This is an example of an RSS feed</description>
  <link>http://www.example.com/main.html</link>
  <lastBuildDate>Mon, 06 Sep 2019 00:01:00 +0000 </lastBuildDate>
  <pubDate>Sun, 06 Sep 2019 16:20:00 +0000</pubDate>
  <ttl>1800</ttl>

  <item>
    <title>Example entry</title>
    <description>Here is some text containing an interesting
description.</description>
    <link>http://www.example.com/blog/post/1</link>
    <guid isPermaLink="false">7bd204c6-1655-4c27-aeee-53f933c5395f</guid>
    <pubDate>Sun, 06 Sep 2019 16:20:00 +0000</pubDate>
  </item>

</channel>
</rss>
```

TECNOLOGIA E EMPREENDEDORISMO

Configuração do ambiente

```
### criar o diretório do seu projeto master  
mkdir av_setembro_back
```

```
### JÁ NA PASTA QUE VAI FICAR O PROJETO ###
```

```
## instale o VIRTUALENV  
pip install virtualenv
```

```
## inicie um ambiente virtual  
virtualenv .
```

```
## ative o ambiente virtual  
.\Scripts\activate
```

```
## instale os módulos necessários  
pip install django  
Pip install djangorestframework  
pip install markdown
```

Configurando o setup

```
## iniciar o projeto  
django-admin startproject setup .
```

```
## abrir o arquivo de SETTINGS  
code .\setup\settings.py
```

```
## alterar as informações  
'en-us' >>> 'pt-br'  
'UTC' >>> 'America/Fortaleza'
```

Configurando o av_rss

```
## criar app
```

```
python .\manage.py startapp av_rss
```

```
## acessar o settings.py e add os apps
```

```
INSTALLED_APPS = [
```

```
    (...)
```

```
    'rest_framework',
```

```
    'av_rss',
```

```
]
```

TECNOLOGIA E EMPREENDEDORISMO

Entenda o ambiente

Para trabalhar com o REST API vamos ter que montar a estrutura de cinco arquivo:

- `models.py`
- `serializer.py`
- `admin.py`
- `views.py`
- `urls.py`

Models

É o principal arquivo do ambiente.

Faz a ponte do API com o DB

As definições da estrutura toda são feitas nele

Deve conter uma função `__str__` para mostrar quem vai representar o cada model

TECNOLOGIA E EMPREENDEDORISMO

Models

```
from django.db import models

class Canal(models.Model):
    titulo = models.CharField(max_length=255)
    descricao = models.TextField(blank=True, null=True)
    link = models.URLField()

    # Essa função é feita para mostrar quem vai representar o
cada canal
    def __str__(self):
        return self.titulo

class Etiqueta(models.Model):
    canal = models.ForeignKey(Canal, on_delete=models.CASCADE)
    titulo = models.CharField(max_length=255)
    descricao = models.TextField(blank=True, null=True)
    link = models.URLField()
    pub_date = models.DateTimeField()

    # Essa função é feita para mostrar quem vai representar o
cada etiqueta
    def __str__(self):
        return self.titulo
```

Models

Após a criação dos seus models faz necessário criar todo ambiente do banco de dados.

`python .\manage.py makemigrations` - para gerar as migrações do ADMIN e do MODELS

`python .\manage.py migrate` - para executar as migrações

`python .\manage.py createsuperuser` - para criar o usuário de acesso ao ADM

`python .\manage.py runserver` - para levantar o serviço na porta

Serializer

É o responsável por transformar as informações para um formato que a API vai entender

Pode ser configurada para JSON ou XML

Cria uma ponte entre as informações escritas em PYTHON e JSON/XML

Serializer

```
from rest_framework import serializers
from av_rss.models import Canal, Etiqueta

class CanalSerializer(serializers.ModelSerializer):
    class Meta:
        model = Canal
        fields = [
            "descricao",
            "link",
            "titulo",
        ]

class EtiquetaSerializer(serializers.ModelSerializer):
    class Meta:
        model = Etiqueta
        fields = '__all__'
```

Admin

Para gerir e fazer a administração do ambiente DJANGO REST

Com ele é possível fazer o CRUD dos elementos configurados

AMONTADA
VALLEY

TECNOLOGIA E EMPREENDEDORISMO

Admin

```
from django.contrib import admin
from av_rss.models import Canal, Etiqueta

class Canais(admin.ModelAdmin):
    list_display = ('id', 'titulo', 'descricao', 'link')
    list_display_links = ('id', 'titulo')
    search_fields = ('titulo',)
    list_per_page = 20

admin.site.register(Canal, Canais)

class Etiquetas(admin.ModelAdmin):
    list_display = ('id', 'canal', 'titulo', 'descricao',
                    'link', 'pub_date')
    list_display_links = ('id', 'canal')
    search_fields = ('canal',)

admin.site.register(Etiqueta, Etiquetas)
```

Views

Responsável por receber os dados da API

É ele que defini o tipo de serialização que será utilizada na API

A estrutura REST permite que utilizemos uma estrutura de abstração chamadas VIEWSET

O VIEWSET permite que as pessoas que forem utilizar a API possa trabalhar bem com a modelagem das nossas informações.

Deixa a construção da URL automaticamente criada com as convenções de API mais comuns

Views

```
from rest_framework import viewsets
from av_rss.models import Canal, Etiqueta
from av_rss.serializer import CanalSerializer,
EtiquetaSerializer
```

```
class CanaisViewSet(viewsets.ModelViewSet):
    """Exibindo todas os canais"""
    queryset = Canal.objects.all()
    serializer_class = CanalSerializer
```

```
class EtiquetasViewSet(viewsets.ModelViewSet):
    """Exibindo todos as etiquetas"""
    queryset = Etiqueta.objects.all()
    serializer_class = EtiquetaSerializer
```

Ur1s

Trabalha com as rotas do ambiente REST

Configurar as rotas para cada VIEWSET



AMONTADA
VALLEY

TECNOLOGIA E EMPREENDEDORISMO

Urls

```
from django.contrib import admin
from django.urls import path,include
from av_rss.views import CanaisViewSet,
EtiquetasViewSet
from rest_framework import routers

router = routers.DefaultRouter()
router.register('canais', CanaisViewSet,
basename='Canais')
router.register('etiquetas', EtiquetasViewSet,
basename='Etiquetas')

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(router.urls) )
]
```