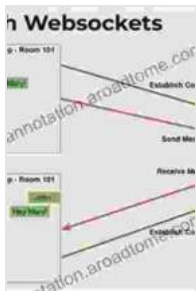


# Lightweight Django Using REST Websockets and Backbone

In this article, we'll build a lightweight Django application that uses RESTful websockets and Backbone.js for real-time communication and a responsive user interface. We'll cover the following topics:



## Lightweight Django: Using REST, WebSockets, and Backbone

by Julia Elman

★★★★☆ 4.1 out of 5

Language : English  
File size : 1710 KB  
Text-to-Speech : Enabled  
Screen Reader : Supported  
Enhanced typesetting : Enabled  
Print length : 346 pages



- Setting up a Django project
- Creating a REST API with Django REST Framework
- Implementing websockets with Django Channels
- Building a Backbone.js application
- Integrating the Django REST API and Backbone.js application

## Prerequisites

Before we begin, you'll need the following installed on your system:

- Python 3.6 or higher
- Django 2.2 or higher
- Django REST Framework 3.9 or higher
- Django Channels 2.3 or higher
- Node.js 10 or higher
- Backbone.js 1.4 or higher

## Setting Up a Django Project

First, let's create a new Django project. Open your terminal and run the following command:

```
bash django-admin startproject myproject
```

This will create a new directory called **myproject** with the following structure:

```
myproject/ ├── manage.py ├── myproject/ | ├── __init__.py |  
           ├── settings.py | ├── urls.py | ├── wsgi.py | ├── README.md
```

## Creating a REST API with Django REST Framework

Next, let's create a REST API with Django REST Framework. Open the **settings.py** file and add the following line to the **INSTALLED\_APPS** list:

```
python INSTALLED_APPS = [ ... 'rest_framework', ]
```

Now, let's create a new app for our API. Open your terminal and run the following command:

```
bash python manage.py startapp api
```

This will create a new directory called **api** with the following structure:

```
api/ ├── __init__.py ├── admin.py ├── apps.py ├── migrations/
     ├── models.py ├── serializers.py ├── tests.py └── views.py
```

Open the **models.py** file and define a new model called **Message** :

```
python from django.db import models
```

```
class Message(models.Model): text = models.TextField() timestamp =
models.DateTimeField(auto_now_add=True)
```

Next, open the **serializers.py** file and create a serializer for the **Message** model:

```
python from rest_framework import serializers
```

```
class MessageSerializer(serializers.ModelSerializer): class Meta: model =
Message fields = ['id', 'text', 'timestamp']
```

Finally, open the **views.py** file and create a viewset for the **Message** model:

```
python from rest_framework import viewsets
```

```
class MessageViewSet(viewsets.ModelViewSet): queryset =
Message.objects.all() serializer_class = MessageSerializer
```

Now, let's add the viewset to the **urls.py** file:

```
python from django.urls import path, include
```

```
urlpatterns = [ ... path('api/', include('api.urls')),]
```

And in the `api/urls.py` file:

```
python from django.urls import path from . import views
```

```
urlpatterns = [ path('messages/', views.MessageViewSet.as_view({'get':  
'list', 'post': 'create'}),name='message-list'),]
```

## Implementing Websockets with Django Channels

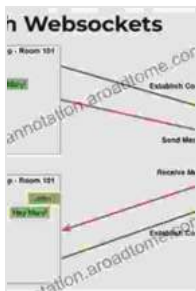
Now, let's implement websockets with Django Channels. Open the `settings.py` file and add the following line to the `INSTALLED_APPS` list:

```
python INSTALLED_APPS = [ ... 'channels', ]
```

Next, create a new directory called `chat` in the `myproject` directory. This directory will contain our websocket code.

Open the `chat/routing.py` file and add the following code:

```
python
```



## Lightweight Django: Using REST, WebSockets, and Backbone

by Julia Elman

★★★★☆ 4.1 out of 5

Language : English

File size : 1710 KB

Text-to-Speech : Enabled

Screen Reader : Supported

Enhanced typesetting : Enabled

Print length : 346 pages

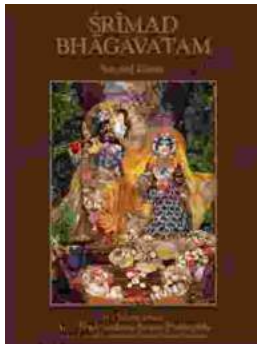
FREE

DOWNLOAD E-BOOK



## Java Learn Java In Days: Your Fast-Track to Programming Proficiency

Are you ready to embark on an extraordinary journey into the world of programming with Java? David Chang, the acclaimed author and programming expert, brings...



## Srimad Bhagavatam Second Canto by Jeff Birkby: A Literary Masterpiece

In the vast tapestry of ancient Indian literature, the Srimad Bhagavatam stands as a towering masterpiece, an inexhaustible source of wisdom and inspiration. Its Second Canto,...