Task 1: API Creation

To create an API based on the provided documentation, you'll need to determine the specific requirements, such as the endpoints, request/response formats, and any additional functionalities. However, without the actual documentation, it's challenging to provide a comprehensive solution. Nevertheless, I can provide you with a basic template for creating an API using Django Rest Framework (DRF).

Here's an example of how you can create a simple API using Django:

1. Install Django and Django Rest Framework:

pip install django djangorestframework

2. Create a new Django project:

django-admin startproject myproject

cd myproject

3. Create a new Django app:

python manage.py startapp myapp

4. In the `myproject/settings.py` file, add `'rest\_framework'` to the `INSTALLED\_APPS` list.

5. In the `myproject/urls.py` file, add the following code:

from django.urls import include, path

from rest\_framework import routers

from myapp.views import ExampleViewSet

router = routers.DefaultRouter()

router.register(r'example', ExampleViewSet)

urlpatterns = [

path('api/', include(router.urls)),

]

6. In the `myapp/views.py` file, add the following code:

from rest\_framework import viewsets

from rest\_framework.response import Response

class ExampleViewSet(viewsets.ViewSet):

def list(self, request):

response\_data = {

'message': 'This is an example endpoint',

'data': {

'key': 'value'

}

}

return Response(response\_data)

7. Run the development server:

python manage.py runserver

This will start the Django development server, and you can access the API at `http://localhost:8000/api/example` in your browser or using an API testing tool like Postman.

Please provide the actual API documentation or more details about the functionality for a more accurate solution.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Task 2: API Documentation in Django

To write API documentation in Django, you can use the Django Rest Framework's built-in support for generating documentation using the OpenAPI Specification (OAS) and Swagger UI.

Here's an example of how you can document an API using Django Rest Framework:

1. Install `django-rest-swagger` package:

pip install django-rest-swagger

2. In the `myproject/settings.py` file, add `'rest\_framework\_swagger'` to the `INSTALLED\_APPS` list.

3. In the `myproject/urls.py` file, modify the code as follows:

from django.urls import include, path

from rest\_framework import routers

from myapp.views import ExampleViewSet

from rest\_framework\_swagger.views import get\_swagger\_view

router = routers.DefaultRouter()

router.register(r'example', ExampleViewSet)

urlpatterns = [

path('api/', include(router.urls)),

path('docs/', get\_swagger\_view(title='API Documentation')),

]

4. Run the development server:

python manage.py runserver

Now, you can access the API documentation at `http://localhost:8000/docs/`. It will provide an interactive UI where you can explore the API endpoints and their documentation.

Remember to replace `'example'` with the appropriate endpoint and viewset names in the code provided based on your functionality.

Note: Django Rest Framework also supports other documentation tools like drf-yasg and drf-spectacular. You can choose the one that best fits your requirements.

Please provide the actual functionality or more details about the API for a more accurate documentation solution.