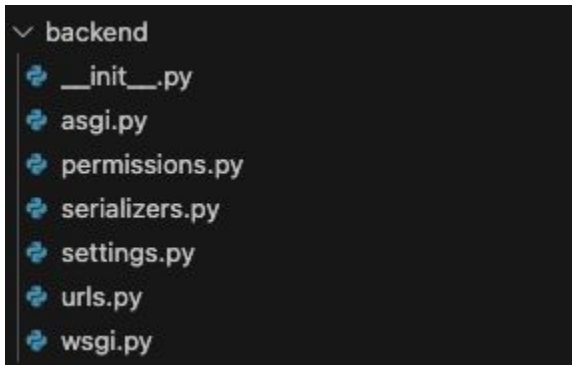


# Project: Backend



# asgi.py



CI Python Linter

```
1 import os
2
3 from django.core.asgi import get_asgi_application
4
5 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'backend.settings')
6
7 application = get_asgi_application()
8 |
```

Settings:



Results:

All clear, no errors found

# permissions.py



CI Python Linter

```
1 from rest_framework import permissions
2
3
4 class IsOwnerOrReadOnly(permissions.BasePermission):
5     def has_object_permission(self, request, view, obj):
6         if request.method in permissions.SAFE_METHODS:
7             return True
8         return obj.owner == request.user
9
10
11 class IsInviteOwnerOrReadOnly(permissions.BasePermission):
12     def has_object_permission(self, request, view, obj):
13         if request.method in permissions.SAFE_METHODS:
14             return True
15         return obj.user == request.user and obj.inbound
16
17
18 class IsAuthorOrReadOnly(permissions.BasePermission):
19     def has_object_permission(self, request, view, obj):
20         if request.method in permissions.SAFE_METHODS:
21             return True
22         return obj.author == request.user
23
24
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1 from dj_rest_auth.serializers import UserDetailsSerializer
2 from rest_framework import serializers
3
4
5 class CurrentUserSerializer(UserDetailsSerializer):
6     profile_id = serializers.ReadOnlyField(source='profile.id')
7     profile_avatar = serializers.ReadOnlyField(source='profile.avatar.url')
8     default_workstream = serializers.ReadOnlyField(
9         source='profile.default_workstream.name'
10    )
11     default_workstream_id = serializers.ReadOnlyField(
12         source='profile.default_workstream.id'
13    )
14
15 class Meta(UserDetailsSerializer.Meta):
16     fields = UserDetailsSerializer.Meta.fields + (
17         'profile_id', 'profile_avatar', 'default_workstream',
18         'default_workstream_id'
19     )
20
```

Settings:



Results:

All clear, no errors found

# settings.py



CI Python Linter

```
1 """
2 Django settings for backend project.
3
4 Generated by 'django-admin startproject' using Django 3.2.4.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.2/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.2/ref/settings/
11 """
12
13 from pathlib import Path
14 import os
15 import dj_database_url
16
17 if os.path.exists('env.py'):
18     import env
19
20 CLOUDINARY_STORAGE = {
21     'CLOUDINARY_URL': os.environ.get('CLOUDINARY_URL')
22 }
23 MEDIA_URL = '/media/'
24 DEFAULT_FILE_STORAGE = 'cloudinary_storage.storage.MediaCloudinaryStorage'
```

Settings:



Results:

[168](#): E501 line too long (91 > 79 characters)  
[171](#): E501 line too long (81 > 79 characters)  
[174](#): E501 line too long (82 > 79 characters)  
[177](#): E501 line too long (83 > 79 characters)

Some lines of standard Django code are too long and impossible to reduce.

# urls.py



CI Python Linter

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('frontend.urls')),
7     path('api-auth/', include('rest_framework.urls')),
8     path('dj-rest-auth/', include('dj_rest_auth.urls')),
9     path('dj-rest-auth/registration/', include(
10         'dj_rest_auth.registration.urls'
11     )),
12     path('api/', include('profiles.urls')),
13     path('api/', include('workstream.urls')),
14     path('api/', include('category.urls')),
15     path('api/', include('task.urls')),
16     path('api/', include('project.urls')),
17     path('api/', include('invite.urls')),
18 ]
19
```

Settings:



Results:

All clear, no errors found

# wsgi.py



CI Python Linter

```
1 import os
2
3 from django.core.wsgi import get_wsgi_application
4
5 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'backend.settings')
6
7 application = get_wsgi_application()
8 |
```

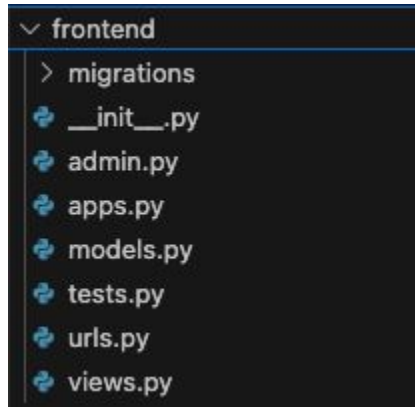
Settings:



Results:

All clear, no errors found

# App: Frontend





# apps.py



## CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class FrontendConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'frontend'
7
```

Settings:



Results:

All clear, no errors found

# urls.py



CI Python Linter

```
1 from django.urls import path, re_path
2 from .views import index
3
4 urlpatterns = [
5     path('', index, name='index'),
6     re_path(
7         r"^(?!api/lapildj-rest-auth/ldj-rest-auth).*",
8         index,
9     ),
10 ]
11
```

Settings:



Results:

All clear, no errors found

# views.py



CI Python Linter

```
1 from django.shortcuts import render
2
3
4 # Create your views here.
5 def index(request):
6     return render(request, "build/index.html")
7 |
```

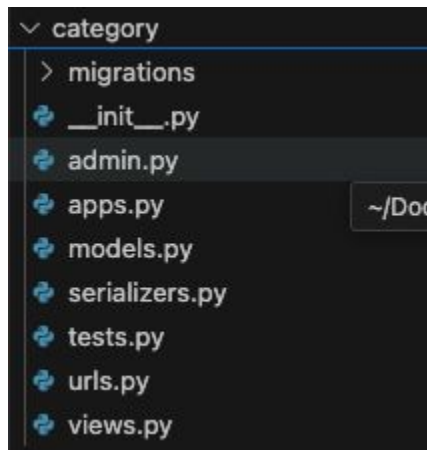
Settings:



Results:

All clear, no errors found

# App: Category



# apps.py



CI Python Linter

```
1 '''apps for category app'''
2 from django.apps import AppConfig
3
4
5 class CategoryConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'category'
8 |
```

Settings:



Results:

All clear, no errors found

# models.py



CI Python Linter

```
1 '''models for category app'''
2 from django.db import models
3 from django.contrib.auth.models import User
4 from workstream.models import Workstream
5
6 |
7 class Category(models.Model):
8     '''Category Model'''
9     owner = models.ForeignKey(User, on_delete=models.CASCADE)
10    created_at = models.DateTimeField(auto_now_add=True)
11    updated_at = models.DateTimeField(auto_now=True)
12    name = models.CharField(max_length=255, blank=True)
13    workstream = models.ForeignKey(
14        Workstream, on_delete=models.CASCADE,
15        related_name='category_workstream'
16    )
17
18 class Meta:
19     ordering = ['-created_at']
20
21 def __str__(self):
22     return self.name
23
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1 '''serializer for category app'''
2 from rest_framework import serializers
3 from .models import Category
4
5
6 class CategorySerializer(serializers.ModelSerializer):
7     owner = serializers.ReadOnlyField(source='owner.username')
8     workstream = serializers.ReadOnlyField(source='workstream.name')
9     is_owner = serializers.SerializerMethodField()
10
11     def get_is_owner(self, obj):
12         request = self.context['request']
13         return request.user == obj.owner
14
15     class Meta:
16         model = Category
17         fields = [
18             'id', 'owner', 'name', 'workstream', 'is_owner',
19         ]
20
```

Settings:



Results:

All clear, no errors found

# urls.py



CI Python Linter

```
1 '''urls for category app'''
2 from django.urls import path
3 from category import views
4
5 urlpatterns = [
6     path('category/', views.CategoryList.as_view()),
7     path('category/<int:pk>/', views.CategoryDetail.as_view()),
8 ]
9
```

Settings:



Results:

All clear, no errors found



# views.py



CI Python Linter

```
1  '''view for category app'''
2  from rest_framework import generics
3  from backend.permissions import IsStaffOrReadOnly
4  from .models import Category
5  from .serializers import CategorySerializer
6
7
8  class CategoryList(generics.ListCreateAPIView):
9      """
10     List Category view
11     """
12     serializer_class = CategorySerializer
13     permission_classes = [IsStaffOrReadOnly]
14
15     def get_queryset(self):
16         """
17         This view should return a list of all the categories in the
18         current users default workstream
19         """
20         user = self.request.user
21         return Category.objects.filter(
22             workstream=user.profile.default_workstream
23         )
24
```

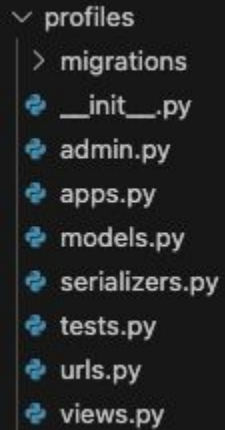
Settings:



Results:

All clear, no errors found

# App: Profiles



A screenshot of a file explorer interface with a dark background. The 'profiles' directory is expanded, showing a list of files and subdirectories. The 'migrations' subdirectory is also expanded. Each file is preceded by a small blue icon representing a Python file.

- profiles
  - migrations
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - models.py
  - serializers.py
  - tests.py
  - urls.py
  - views.py

# admin.py



## CI Python Linter

```
1 from django.contrib import admin
2 from .models import Profile
3
4 admin.site.register(Profile)
5
```

### Settings:



### Results:

All clear, no errors found

# apps.py



## CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class ProfilesConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'profiles'
7
```

Settings:



Results:

All clear, no errors found

# models.py



CI Python Linter

```
1  '''Models for profiles app'''
2  from django.db import models
3  from django.db.models.signals import post_save
4  from django.contrib.auth.models import User
5  from workstream.models import Workstream
6
7
8  class Profile(models.Model):
9      '''Profile model'''
10     owner = models.OneToOneField(User, on_delete=models.CASCADE)
11     created_at = models.DateTimeField(auto_now_add=True)
12     updated_at = models.DateTimeField(auto_now=True)
13     first_name = models.CharField(max_length=255, blank=True)
14     last_name = models.CharField(max_length=255, blank=True)
15     avatar = models.ImageField(
16         upload_to='images/avatar/', default='../default_profile_dkfqqb'
17     )
18     default_workstream = models.ForeignKey(
19         Workstream, null=True, on_delete=models.SET_NULL)
20
21     class Meta:
22         ordering = ['-created_at']
23
24     def __str__(self):
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1 from rest_framework import serializers
2 from .models import Profile
3 from workstream.models import Workstream
4
5
6 class UserWorkstreamForeignKey(serializers.PrimaryKeyRelatedField):
7     '''class for returning user specific workstream'''
8     def get_queryset(self):
9         user = self.context['request'].user
10        return Workstream.objects.filter(ws_participants__owner=user)
11
12
13 class ProfileSerializer(serializers.ModelSerializer):
14     '''serializer for get profile api'''
15     owner = serializers.ReadOnlyField(source='owner.username')
16     is_owner = serializers.SerializerMethodField()
17     default_workstream = UserWorkstreamForeignKey()
18
19     def get_is_owner(self, obj):
20         request = self.context['request']
21         return request.user == obj.owner
22
23 class Meta:
24     model = Profile
```

Settings:



Results:

All clear, no errors found

# urls.py



## CI Python Linter

```
1  '''urls for profiles app'''
2  from django.urls import path
3  from profiles import views
4
5  urlpatterns = [
6      path('profiles/', views.ProfileList.as_view()),
7      path('profiles/invite/', views.ProfileInviteList.as_view()),
8      path('profiles/<int:pk>', views.ProfileDetail.as_view()),
9      path('profiles/switch/<int:pk>', views.ProfileSwitchWorkstream.as_view()),
10 ]
11 |
```

Settings:



Results:

All clear, no errors found

# views.py



CI Python Linter

```
1  '''view for profiles app'''
2  from django.db.models import Count
3  from django.db.models import Q
4  from rest_framework import generics
5  from backend.permissions import IsOwnerOrReadOnly
6  from .models import Profile
7  from .serializers import *
8
9
10 class ProfileList(generics.ListAPIView):
11     """
12     List all profiles.
13     No create view as profile creation is handled by django signals.
14     """
15     queryset = Profile.objects.all().order_by('-created_at')
16     serializer_class = ProfileSerializer
17
18
19 class ProfileInviteList(generics.ListAPIView):
20     """
21     List all profiles for workstream invites
22     excludes already participating users and pending invite
23     """
24     serializer_class = ProfileSerializer
```

Settings:

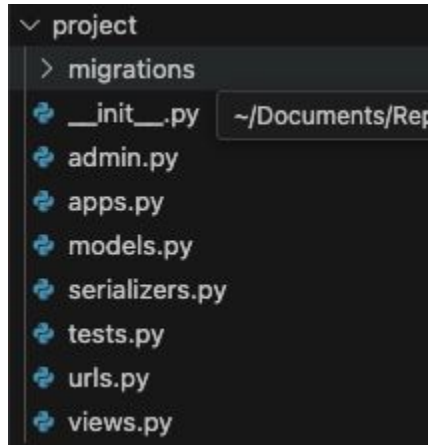


Results:

All clear, no errors found



# App: Project



# apps.py



## CI Python Linter

```
1 '''apps for project app'''
2 from django.apps import AppConfig
3
4
5 class ProjectConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'project'
8 |
```

Settings:



Results:

All clear, no errors found

# models.py



CI Python Linter

```
1 '''models for project model'''
2 from django.db import models
3 from django.contrib.auth.models import User
4 from workstream.models import Workstream
5
6
7 class Project(models.Model):
8     '''
9     Project model
10    '''
11    title = models.CharField(max_length=255)
12    workstream = models.ForeignKey(Workstream, on_delete=models.CASCADE)
13    created_at = models.DateTimeField(auto_now_add=True)
14    updated_at = models.DateTimeField(auto_now=True)
15    owner = models.ForeignKey(User, on_delete=models.CASCADE)
16
17 class Meta:
18     ordering = ['-created_at']
19
20 def __str__(self):
21     return self.title
22
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1  '''serialiser for project app'''
2  from rest_framework import serializers
3  from .models import Project
4
5
6  class ProjectSerializer(serializers.ModelSerializer):
7      '''
8      Project serializer
9      '''
10     owner = serializers.ReadOnlyField(source='owner.username')
11     workstream = serializers.ReadOnlyField(source='workstream.name')
12     is_owner = serializers.SerializerMethodField()
13
14     def get_is_owner(self, obj):
15         request = self.context['request']
16         return request.user == obj.owner
17
18     class Meta:
19         model = Project
20         fields = [
21             'id', 'owner', 'title', 'workstream', 'is_owner',
22         ]
23
```

Settings:



Results:

All clear, no errors found

# urls.py



CI Python Linter

```
1  '''urls for project app'''
2  from django.urls import path
3  from project import views
4
5  urlpatterns = [
6      path('project/', views.ProjectList.as_view()),
7      path('project/<int:pk>/', views.ProjectDetail.as_view()),
8  ]
9
```

Settings:



Results:

All clear, no errors found

# views.py



CI Python Linter

```
1  '''views for project app'''
2  from rest_framework import generics
3  from backend.permissions import IsStaffOrReadOnly, IsOwnerOrReadOnly
4  from .models import Project
5  from .serializers import ProjectSerializer
6
7
8  class ProjectList(generics.ListCreateAPIView):
9      """
10     List project view api
11     """
12     serializer_class = ProjectSerializer
13     permission_classes = [IsStaffOrReadOnly]
14
15     def get_queryset(self):
16         """
17         filter list to display active users default workstream categories
18         """
19         user = self.request.user
20         return Project.objects.filter(
21             workstream=user.profile.default_workstream)
22
23     def perform_create(self, serializer):
24         """
```

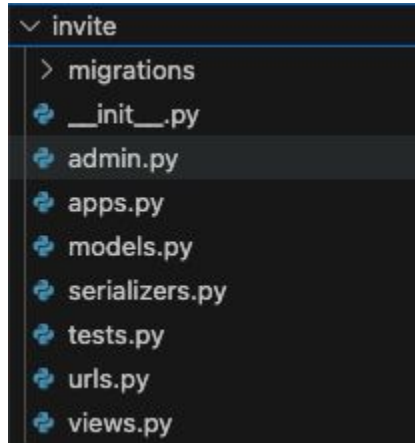
Settings:



Results:

All clear, no errors found

# App: Invites



# apps.py



## CI Python Linter

```
1 '''apps for invite app'''
2 from django.apps import AppConfig
3
4
5 class InviteConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'invite'
8
```

Settings:



Results:

All clear, no errors found



# models.py



CI Python Linter

```
1 '''models for invite model'''
2 from django.db import models
3 from django.db.models.signals import post_save
4 from django.contrib.auth.models import User
5 from workstream.models import Workstream
6
7
8 class Invite(models.Model):
9     '''
10     Invite model
11     '''
12     workstream = models.ForeignKey(
13         Workstream, on_delete=models.CASCADE, related_name='ws_invite')
14     created_at = models.DateTimeField(auto_now_add=True)
15     updated_at = models.DateTimeField(auto_now=True)
16     user = models.ForeignKey(
17         User, on_delete=models.CASCADE, related_name='user_invite')
18     accepted = models.BooleanField(default=False)
19     inbound = models.BooleanField(default=True)
20
21 class Meta:
22     ordering = ['-created_at']
23
24 def __str__(self):
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1  '''serializer for invite app'''
2  from rest_framework import serializers
3  from .models import Invite
4  from workstream.models import Workstream
5
6
7  class InviteSerializer(serializers.ModelSerializer):
8      '''
9      serializer for inviting other users to your workstream
10     '''
11     accepted = serializers.ReadOnlyField()
12     inbound = serializers.ReadOnlyField()
13     workstream = serializers.ReadOnlyField(source='workstream.name')
14     is_owner = serializers.SerializerMethodField()
15     display_name = serializers.ReadOnlyField(source='user.username')
16     avatar = serializers.ReadOnlyField(source='user.profile.avatar.url')
17
18     def get_is_owner(self, obj):
19         request = self.context['request']
20         return obj.workstream.owner == request.user
21
22     class Meta:
23         model = Invite
24         fields = [
```

Settings:



Results:

All clear, no errors found

# urls.py



CI Python Linter

```
1  '''urls for invites app'''
2  from django.urls import path
3  from invite import views
4
5  urlpatterns = [
6      path('invite/', views.InviteList.as_view()),
7      path('invite/<int:pk>/', views.InviteDetail.as_view()),
8      path('join/', views.JoinList.as_view()),
9      path('join/<int:pk>/', views.JoinDetail.as_view()),
10 ]
11
```

Settings:



Results:

All clear, no errors found

# views.py



## CI Python Linter

```
1  '''views for invites app'''
2  from rest_framework import generics
3  from backend.permissions import (IsWorkstreamOwnerOrReadOnly,
4                                  IsInviteOwnerOrReadOnly,
5                                  InvitePermissions
6                                  )
7  from .models import Invite
8  from .serializers import InviteSerializer, JoinSerializer, AcceptSerializer
9
10
11 class InviteList(generics.ListCreateAPIView):
12     """
13     List invites or create new
14     """
15     serializer_class = InviteSerializer
16     permission_classes = [IsWorkstreamOwnerOrReadOnly]
17
18     def get_queryset(self):
19         """
20         filter list to display users default workstream invites
21         """
22         user = self.request.user
23         return Invite.objects.filter(
24             workstream=user.profile.default_workstream)
```

Settings:



Results:

All clear, no errors found

# App: Task

```
▼ task
  > migrations
  🌀 __init__.py
  🌀 admin.py
  🌀 apps.py
  🌀 models.py
  🌀 serializers.py
  🌀 tests.py
  🌀 urls.py
  🌀 views.py
```

# apps.py



## CI Python Linter

```
1 '''apps for task app'''
2 from django.apps import AppConfig
3
4
5 class TaskConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'task'
8
```

Settings:



Results:

All clear, no errors found

# models.py



CI Python Linter

```
1  '''models for task app'''
2  from django.db import models
3  from django.contrib.auth.models import User
4  from category.models import Category
5  from project.models import Project
6
7
8  class Task(models.Model):
9      '''
10     Task model
11     '''
12     PRIORITY_LEVELS = [
13         (1, 'No-priority'),
14         (2, 'Low-priority'),
15         (3, 'Medium-priority'),
16         (4, 'High-priority'),
17     ]
18     author = models.ForeignKey(
19         User, on_delete=models.CASCADE, related_name='author')
20     created_at = models.DateTimeField(auto_now_add=True)
21     updated_at = models.DateTimeField(auto_now=True)
22     name = models.CharField(max_length=255)
23     detail = models.TextField(blank=True)
24     priority = models.IntegerField(choices=PRIORITY_LEVELS, default=1)
25     is_completed = models.BooleanField(default=False)
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1  '''serializer for task app'''
2  from rest_framework import serializers
3  from .models import Task
4  from category.models import Category
5  from category.serializers import CategorySerializer
6  from project.serializers import ProjectSerializer
7  from backend.serializers import CurrentUserSerializer
8
9
10 class CategoryForeignKey(serializers.PrimaryKeyRelatedField):
11     def get_queryset(self):
12         user = self.context['request'].user
13         return Category.objects.filter(
14             workflow=user.profile.default_workflow)
15
16
17 class TaskCreateSerializer(serializers.ModelSerializer):
18     '''
19     serializer for creating task
20     '''
21     author = serializers.ReadOnlyField(source='user.username')
22     is_owner = serializers.SerializerMethodField()
23     category = CategoryForeignKey()
24
```

Settings:



Results:

All clear, no errors found



# urls.py



CI Python Linter

```
1 '''urls for task app'''
2 from django.urls import path
3 from task import views
4
5 urlpatterns = [
6     path('task/open/', views.TaskListOpen.as_view()),
7     path('task/assigned/', views.TaskListAssigned.as_view()),
8     path('task/create/', views.TaskCreate.as_view()),
9     path('tasklist/', views.UserTaskList.as_view()),
10    path('task/<int:pk>/', views.TaskDetail.as_view()),|
11    path('task/<int:pk>/complete/', views.CompleteTask.as_view()),
12    path('task/<int:pk>/admin/', views.TaskAssignAdmin.as_view()),
13    path('task/<int:pk>/assign/', views.TaskAssignSelf.as_view()),
14    path('task/<int:pk>/leave/', views.TaskUnassignSelf.as_view()),
15 ]
16
```

Settings:



Results:

All clear, no errors found

# views.py



CI Python Linter

```
1 '''views for task app'''
2 from rest_framework import generics
3 from backend.permissions import (IsOwnerOrReadOnly,
4                                 IsAuthorOrReadOnly,
5                                 IsParticipantOrReadOnly)
6 from .models import Task
7 from .serializers import *
8
9
10 class TaskCreate(generics.CreateAPIView):
11     """
12     Task create view
13     """
14     serializer_class = TaskCreateSerializer
15
16     def perform_create(self, serializer):
17         serializer.save(author=self.request.user)
18
19
20 class TaskListOpen(generics.ListAPIView):
21     """
22     List all unassigned task in current users default workstream.
23     """
24     serializer_class = TaskSerializer
```

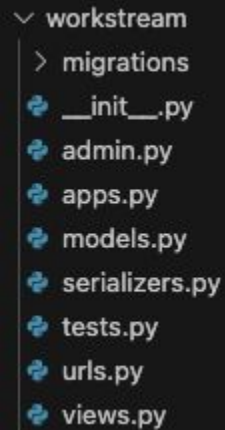
Settings:



Results:

All clear, no errors found

# App: Workstream



```
workstream
├── migrations
├── __init__.py
├── admin.py
├── apps.py
├── models.py
├── serializers.py
├── tests.py
├── urls.py
└── views.py
```

A screenshot of a file explorer interface with a dark background. It shows a directory named 'workstream' which is expanded. Inside 'workstream', there is a subdirectory 'migrations' and several Python files: '\_\_init\_\_.py', 'admin.py', 'apps.py', 'models.py', 'serializers.py', 'tests.py', 'urls.py', and 'views.py'. Each file is preceded by a small blue icon representing a file.

# apps.py



## CI Python Linter

```
1 '''apps for workstream app'''
2 from django.apps import AppConfig
3
4
5 class WorkstreamConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'workstream'
8
```

Settings:



Results:

All clear, no errors found

# models.py



CI Python Linter

```
1  '''models for workstream app'''
2  from django.db import models
3  from django.db.models.signals import post_save
4  from django.contrib.auth.models import User
5
6
7  class Workstream(models.Model):
8      '''Workstream Model'''
9      owner = models.ForeignKey(User, on_delete=models.CASCADE)
10     created_at = models.DateTimeField(auto_now_add=True)
11     updated_at = models.DateTimeField(auto_now=True)
12     name = models.CharField(max_length=255, blank=True)
13     users = models.ManyToManyField(
14         User, through='Participant', related_name='workstream_users')
15
16     class Meta:
17         ordering = ['-created_at']
18
19     def __str__(self):
20         return self.name
21
22
23     class Participant(models.Model):
24         '''Participant model'''
```

Settings:



Results:

All clear, no errors found

# serializers.py



CI Python Linter

```
1  '''serializer for workstream app'''
2  from rest_framework import serializers
3  from .models import Workstream, Participant
4  from backend.serializers import CurrentUserSerializer
5
6
7  class WorkstreamSerializer(serializers.ModelSerializer):
8      '''
9      workstream serializer
10     '''
11     owner = CurrentUserSerializer(read_only=True)
12     users = CurrentUserSerializer(many=True, read_only=True)
13     is_owner = serializers.SerializerMethodField()
14
15     def get_is_owner(self, obj):
16         request = self.context['request']
17         return request.user == obj.owner
18
19     class Meta:
20         model = Workstream
21         fields = [
22             'id', 'owner', 'created_at', 'updated_at', 'name',
23             'users', 'is_owner',
24         ]
```

Settings:



Results:

All clear, no errors found

# urls.py



CI Python Linter

```
1 '''urls for workstream app'''
2 from django.urls import path
3 from workstream import views
4
5 urlpatterns = [
6     path('workstream/', views.WorkstreamUserList.as_view()),
7     path('workstream/all/', views.WorkstreamList.as_view()),
8     path('workstream/leave/', views.GetLeaveWorkstream.as_view()),
9     path('workstream/join/', views.WorkstreamInvitelist.as_view()),
10    path('workstream/leave/<int:pk>/', views.LeaveWorkstream.as_view()),
11    path('workstream/active/', views.ParticipantListActive.as_view()),
12    path('workstream/participant/', views.ParticipantList.as_view()),
13    path('workstream/<int:pk>/', views.WorkstreamDetail.as_view()),
14    path('workstream/participant/<int:pk>/',
15         views.ParticipantDetail.as_view()),
16 ]
17
```

Settings:



Results:

All clear, no errors found

# views.py



CI Python Linter

```
1  '''views for workstream app'''
2  from rest_framework import generics
3  from backend.permissions import IsOwnerOrReadOnly, IsWorkstreamOwnerOrReadOnly
4  from .models import Workstream, Participant
5  from .serializers import WorkstreamSerializer, ParticipantSerializer
6
7
8  class WorkstreamList(generics.ListAPIView):
9      """
10     List all workstreams.
11     """
12     queryset = Workstream.objects.all().order_by('-created_at')
13     serializer_class = WorkstreamSerializer
14
15
16  class WorkstreamUserList(generics.ListCreateAPIView):
17      """
18     List all workstreams for current user
19     """
20     serializer_class = WorkstreamSerializer
21
22     def perform_create(self, serializer):
23         """
24         create workstream with default value for owner
```

Settings:



Results:

All clear, no errors found