

Session 3: *Various other things*

Xander Warszawski

GitHub



IEEE Student Branch
KU Leuven Campus Brugge

0

Contents of last session

1. Django Architecture
2. Environment variables
3. Pages app
4. Static assets
5. Common app
6. Albums app

1

GitHub



IEEE Student Branch
KU Leuven Campus Brugge

1

Contents of this session

1. Accounts app
2. Auto generated model diagram
3. Emails
4. Integrating payments using Mollie

2

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

2

1. Accounts app

3

Git checkpoint

1. Create new feature branch

4

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

4

Accounts app

1. Execute: ***docker-compose exec web python manage.py startapp accounts***
2. Update ***accounts/models.py***

5

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

5

accounts/models.py

```
from django.db import models
from django.contrib.auth.models import AbstractUser
from albums.models import Album

class CustomUser(AbstractUser):
    albums = models.ManyToManyField(Album)
```

6




IEEE Student Branch
KU Leuven Campus Brugge

6

Accounts app

1. Execute: **`docker-compose exec web python manage.py startapp accounts`**
2. Update **`accounts/models.py`**
3. Update **`accounts/admin.py`**

7




IEEE Student Branch
KU Leuven Campus Brugge

7

accounts/admin.py

```
from django.contrib import admin
from django.contrib.auth import get_user_model
from django.contrib.auth.admin import UserAdmin

class CustomUserAdmin(UserAdmin):
    model = get_user_model()
    list_display = [
        "email",
        "is_superuser"
    ]

admin.site.register(get_user_model(), CustomUserAdmin)
```

8




IEEE Student Branch
KU Leuven Campus Brugge

8

Accounts app

1. Execute: ***docker-compose exec web python manage.py startapp accounts***
2. Update ***accounts/models.py***
3. Update ***accounts/admin.py***
4. Update ***accounts/views.py***

9




IEEE Student Branch
KU Leuven Campus Brugge

9

accounts/views.py (1)

```
from django.views.generic import ListView
from django.db.models import Q

from albums.models import Album

class AlbumListView(ListView):
    paginate_by = 10
    model = Album
    context_object_name = "album_list"
    template_name = "albums/album_list.html"
```

10




IEEE Student Branch
KU Leuven Campus Brugge

10

accounts/views.py (2)

```
def get_queryset(self):
    query = self.request.GET.get("q")

    if query is None:
        return self.request.user.albums.objects.all()

    return self.request.user.albums.objects.filter(
        Q(title__icontains=query) | Q(artist__name__icontains=query)
    )
```

11




IEEE Student Branch
KU Leuven Campus Brugge

11

Accounts app

1. Execute: **`docker-compose exec web python manage.py startapp accounts`**
2. Update **`accounts/models.py`**
3. Update **`accounts/admin.py`**
4. Update **`accounts/views.py`**
5. Add **`accounts/urls.py`**

12



 IEEE Student Branch
 KU Leuven Campus Brugge

12

`accounts/urls.py`

```
from django.urls import path

from .views import AlbumListView

urlpatterns = [
    path("", AlbumListView.as_view(), name="user_album_list")
]
```

13



 IEEE Student Branch
 KU Leuven Campus Brugge

13

Accounts app

1. Execute: **`docker-compose exec web python manage.py startapp accounts`**
2. Update **`accounts/models.py`**
3. Update **`accounts/admin.py`**
4. Update **`accounts/views.py`**
5. Update **`accounts/urls.py`**
6. Update **`config/settings/base.py`**

14




IEEE Student Branch
KU Leuven Campus Brugge

14

`config/settings/base.py`

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'pages.apps.PagesConfig',
    'albums.apps.AlbumsConfig',
    'accounts.apps.AccountsConfig'
]
...
AUTH_USER_MODEL = "accounts.CustomUser"
```

15




IEEE Student Branch
KU Leuven Campus Brugge

15

Accounts app

1. Execute: **`docker-compose exec web python manage.py startapp accounts`**
2. Update **`accounts/models.py`**
3. Update **`accounts/admin.py`**
4. Update **`accounts/views.py`**
5. Update **`accounts/urls.py`**
6. Update **`config/settings/base.py`**
7. Update **`config/urls.py`**

16

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

16

config/urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('pages.urls')),
    path('albums/', include('albums.urls')),
    path('accounts/', include('accounts.urls'))
] + static(
    settings.MEDIA_URL, document_root=settings.MEDIA_ROOT
)
```

17

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

17

Accounts app

1. Execute: **`docker-compose exec web python manage.py startapp accounts`**
2. Update **`accounts/models.py`**
3. Update **`accounts/admin.py`**
4. Update **`accounts/views.py`**
5. Update **`accounts/urls.py`**
6. Update **`config/settings.py`**
7. Update **`config/urls.py`**
8. Execute: **`docker-compose exec web python manage.py makemigrations`**
9. Execute: **`docker-compose exec web python manage.py migrate`**




IEEE Student Branch
KU Leuven Campus Brugge

18

Exception on migration

1. Drop db and start over:
 1. Docker volume ls
 2. Docker volume rm <name from previous command>

19




IEEE Student Branch
KU Leuven Campus Brugge

19

Accounts app

1. Execute: **`docker-compose exec web python manage.py startapp accounts`**
2. Update **`accounts/models.py`**
3. Update **`accounts/admin.py`**
4. Update **`accounts/views.py`**
5. Update **`accounts/urls.py`**
6. Update **`config/settings.py`**
7. Update **`config/urls.py`**
8. Execute: **`docker-compose exec web python manage.py makemigrations`**
9. Execute: **`docker-compose exec web python manage.py migrate`**
10. Update **`templates/_base.html`**

20

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

20

templates/_base.html

```
<li><a href="{% url 'home' %}">Home</a></li>
<li><a href="{% url 'album_list' %}">Music</a></li>
<li><a href="{% url 'user_album_list' %}">My music</a></li>
<li><a href="">Log out</a></li>
<li><a href="">Log in</a></li>
<li><a href="">Sign up</a></li>
```

21

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

21

Add user registration

1. Pip install **django-allauth** & freeze
2. Update **config/settings/base.py**

22




IEEE Student Branch
KU Leuven Campus Brugge

22

config/settings/base.py (1)

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'allauth',
    'allauth.account',
    'allauth.socialaccount',
    'allauth.socialaccount.providers.github',
    'pages.apps.PagesConfig',
    'albums.apps.AlbumsConfig',
    'accounts.apps.AccountsConfig'
]
```

23




IEEE Student Branch
KU Leuven Campus Brugge

23

config/settings/base.py (2)

```
# django-allauth config
LOGIN_REDIRECT_URL = "home"
ACCOUNT_LOGOUT_REDIRECT = "home"
ACCOUNT_LOGOUT_ON_GET = True
SOCIALACCOUNT_LOGIN_ON_GET = True
SOCIALACCOUNT_AUTO_SIGNUP = True
SITE_ID = 1
AUTHENTICATION_BACKENDS = (
    "django.contrib.auth.backends.ModelBackend",
    "allauth.account.auth_backends.AuthenticationBackend"
)
EMAIL_BACKEND = "django.core.mail.backends.console.EmailBackend"
```

24




IEEE Student Branch
KU Leuven Campus Brugge

24

config/settings/base.py (3)

```
ACCOUNT_SESSION_REMEMBER = True
ACCOUNT_USERNAME_REQUIRED = False
ACCOUNT_AUTHENTICATION_METHOD = "email"
ACCOUNT_EMAIL_REQUIRED = True
ACCOUNT_UNIQUE_EMAIL = True

SOCIALACCOUNT_PROVIDERS = {
    "github": {
        "APP": {
            "client_id": env("GITHUB_CLIENT_ID", ""),
            "secret": env("GITHUB_CLIENT_SECRET", ""),
        }
    }
}
```

25




IEEE Student Branch
KU Leuven Campus Brugge

25

Add user registration

1. Pip install **django-allauth** & freeze
2. Update **config/settings/base.py**
3. Execute: **docker-compose exec web python manage.py migrate**
4. Update **config/urls.py**

26



 IEEE Student Branch
 KU Leuven Campus Brugge

26

config/urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('allauth.urls')),

    path('', include('pages.urls')),
    path('albums/', include('albums.urls')),
    path('accounts/', include('accounts.urls'))
] + static(
    settings.MEDIA_URL, document_root=settings.MEDIA_ROOT
)
```

27



 IEEE Student Branch
 KU Leuven Campus Brugge

27

Add user registration

1. Pip install **django-allauth** and freeze
2. Update **config/settings/base.py**
3. Execute: **docker-compose exec web python manage.py migrate**
4. Update **config/urls.py**
5. Add **templates/account/login.html**
6. Add **templates/account/signup.html**

28

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

28

templates/account/login.html

```
{% extends "_base.html" %}

{% load socialaccount %}

{% block title %}Log In{% endblock title %}

{% block content %}
    <h2>Log In</h2>
    <form method="post">
        {% csrf_token %} <!-- ALWAYS ADD CSRF protection on forms -->
        {{ form.as_p }}
        <button type="submit">Log In</button>
    </form>
    <p>Or</p>
    <a href="{% provider_login_url 'github' %}">Sign Up with GitHub</a>
{% endblock content %}
```

29

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

29

templates/account/signup.html

```
{% extends "_base.html" %}
{% load socialaccount %}
{% block title %}Sign Up{% endblock title %}
{% block content %}
    <h2>Sign Up</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Sign Up</button>
    </form>
    <p>Or</p>
    <a href="{% provider_login_url 'github' %}">Sign Up with GitHub</a>
{% endblock content %}
```

30

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

30

Add login via GitHub

1. Configure new OAuth app:
<https://github.com/settings/applications/new>
 1. Homepage URL: <http://localhost:8000/>
 2. Callback URL: <http://localhost:8000/accounts/github/login/callback>
2. Copy docker-compose.yml, rename to docker-compose-prod.yml and add to .gitignore
3. Update **docker-compose-prod.yml**

31

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

31

docker-compose-prod.yml

environment:

- "DJANGO_SECRET_KEY=<value>"
- "GITHUB_CLIENT_ID=<value>"
- "GITHUB_CLIENT_SECRET=<value>"

32




IEEE Student Branch
KU Leuven Campus Brugge

32

Add user registration

1. Pip install **django-allauth** and freeze
2. Update **config/settings.py**
3. Execute: **docker-compose exec web python manage.py migrate**
4. Update **config/urls.py**
5. Add **templates/account/login.html**
6. Add **templates/account/signup.html**
7. Update **templates/_base.html**

33




IEEE Student Branch
KU Leuven Campus Brugge

33

templates/_base.html

```
<li><a href="{% url 'home' %}">Home</a></li>
<li><a href="{% url 'album_list' %}">Music</a></li>
{% if user.is_authenticated %}
    <li><a href="{% url 'user_album_list' %}">My music</a></li>
    <li><a href="{% url 'account_logout' %}">Log out</a></li>
{% else %}
    <li><a href="{% url 'account_login' %}">Log in</a></li>
    <li><a href="{% url 'account_signup' %}">Sign up</a></li>
{% endif %}
```

34

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

34

Add user registration

1. Pip install **django-allauth** and freeze
2. Update **config/settings.py**
3. Execute: **docker-compose exec web python manage.py migrate**
4. Update **config/urls.py**
5. Add **templates/account/login.html**
6. Add **templates/account/signup.html**
7. Update **templates/_base.html**
8. Update **accounts/tests.py**

35

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

35

accounts/tests.py (1)

```
import pytest
from django.test import TestCase
from django.urls import reverse
from django.contrib.auth import get_user_model

from albums.models import Album, Artist

@pytest.mark.django_db
class AccountTests(TestCase):
```

36




IEEE Student Branch
KU Leuven Campus Brugge

36

accounts/tests.py (2)

```
@classmethod
def setUpTestData(cls):
    cls.artist = Artist.objects.create(
        name = "Beartooth"
    )

    cls.album = Album.objects.create(
        title = "Below",
        price = "15.99",
        record_label = "Red Bull Records"
    )

    cls.artist.albums.add(cls.album)
```

37




IEEE Student Branch
KU Leuven Campus Brugge

37

accounts/tests.py (3)

```
def setUp(self):
    self.user = get_user_model().objects.create_user(
        username="test user",
        email="test@example.com",
        password="testpass123"
    )

    self.user.albums.add(self.album)
```

38




IEEE Student Branch
KU Leuven Campus Brugge

38

accounts/tests.py (4)

```
def test_user_albums_list_view(self):
    self.client.login(email=self.user.email, password="testpass123")
    response = self.client.get(reverse("user_album_list"))
    assert response.status_code == 200
    assert "Below" in response
    assert "albums/album_list.html" in [x.name for x in response.templates]

def test_login_template(self):
    response = self.client.get(reverse("account_login"))
    assert response.status_code == 200
    assert "account/login.html" in [x.name for x in response.templates]

def test_signup_template(self):
    response = self.client.get(reverse("account_signup"))
    assert response.status_code == 200
    assert "account/signup.html" in [x.name for x in response.templates]
```

39




IEEE Student Branch
KU Leuven Campus Brugge

39

Git checkpoint

1. Commit changes
2. Create PR

40

GitHub**IEEE Student Branch**
KU Leuven Campus Brugge

40

Questions?

41

GitHub**IEEE Student Branch**
KU Leuven Campus Brugge

41

2. Auto generated model diagram

42

Git checkpoint

1. Create new feature branch

43

GitHub**IEEE Student Branch**
KU Leuven Campus Brugge

43

Add model diagrams

1. Download and install GraphViz: <https://graphviz.org/download/>
1. Add to System Path!
2. Pip install **django-extensions** and **pyparsing pydot & freeze** pip
3. Add **django_extensions** to **INSTALLED_APPS**
4. Add to base.py:


```
GRAPH_MODELS = {
    'all_applications': True,
    'group_models': True,
}
```
5. Execute: **python manage.py graph_models --pydot -a -g -o my_project_visualized.png**

44



 IEEE Student Branch
 KU Leuven Campus Brugge

44

Git checkpoint

1. Commit changes
2. Create PR

45



 IEEE Student Branch
 KU Leuven Campus Brugge

45

Questions?

46

GitHub**IEEE Student Branch**
KU Leuven Campus Brugge

46

3. *Emails*

47

Git checkpoint

1. Create new feature branch

48

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

48

Custom emails

1. Create folder **templates/account/email**
2. Add files:
 1. email_confirmation_subject.txt
 2. email_confirmation_message.txt

49

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

49

templates/account/email/email_confirmation_subject.txt

```
{% load i18n %}
{% autoescape off %}
{% blocktrans %}Confirm your sign up{% endblocktrans %}
{% endautoescape %}
```

50




IEEE Student Branch
KU Leuven Campus Brugge

50

templates/account/email/email_confirmation_message.txt

```
{% extends 'account/email/base_message.txt' %}
{% load account %}
{% load i18n %}

{% block content %}{% autoescape off %}{% user_display user as user_display %}
{% blocktrans with site_name=current_site.name site_domain=current_site.domain %}
You are receiving this e-mail because user {{ user_display }} has given your
e-mail address to register an account on {{ site_domain }}.

To confirm this is correct, go to {{ activate_url }}
{% endblocktrans %}{% endautoescape %}{% endblock content %}
```

51




IEEE Student Branch
KU Leuven Campus Brugge

51

Custom emails

1. Create folder **templates/account/email**
2. Add files:
 1. email_confirmation_subject.txt
 2. email_confirmation_message.txt
3. Create **templates/account/email_confirm.html**

52



 IEEE Student Branch
 KU Leuven Campus Brugge

52

templates/account/email_confirm.html (1)

```
{% extends '_base.html' %}
{% load i18n %}
{% load account %}

{% block head_title %}{% trans "Confirm E-mail Address" %}{% endblock head_title %}
```

53



 IEEE Student Branch
 KU Leuven Campus Brugge

53

templates/account/email_confirm.html (2)

```
{% block content %}
<h1>{% trans "Confirm E-mail Address" %}</h1>
{% if confirmation %}
    {% user_display confirmation.email_address.user as user_display %}
    <p>
        {% blocktrans with confirmation.email_address.email as email %}
            Please confirm that <a href="mailto:{{ email }}">{{ email }}</a> is an e-mail address for user
            {{ user_display }}.
        {% endblocktrans %}
    </p>
    <form method="post" action="{% url 'account_confirm_email' confirmation.key %}">
        {% csrf_token %}
        <button class="btn btn-primary" type="submit">{% trans 'Confirm' %}</button>
    </form>
```

54




IEEE Student Branch
KU Leuven Campus Brugge

54

templates/account/email_confirm.html (3)

```
{% else %}
    {% url 'account_email' as email_url %}
    <p>
        {% blocktrans %}
            This e-mail confirmation link expired or is invalid.
            Please <a href="{{ email_url }}">issue a new e-mail confirmation request</a>.
        {% endblocktrans %}
    </p>
{% endif %}
{% endblock content %}
```

55




IEEE Student Branch
KU Leuven Campus Brugge

55

Custom emails

1. Create folder **templates/account/email**
2. Add files:
 1. email_confirmation_subject.txt
 2. email_confirmation_message.txt
3. Create **templates/account/email_confirm.html**
4. Get SendGrid API Key

56

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

56

SendGrid

1. Register: <https://signup.sendgrid.com/>
2. Create a Single Sender:
 1. From Email: use your own email for testing purposes
3. Create API Key: https://app.sendgrid.com/settings/api_keys
 1. Restricted access
 1. Full access to mail send

57

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

57

Custom emails

1. Create folder **templates/account/email**
2. Add files:
 1. email_confirmation_subject.txt
 2. email_confirmation_message.txt
3. Create **templates/account/email_confirm.html**
4. Update **config/settings/base.py**

58

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

58

config/settings/base.py

```
SENDGRID_API_KEY = env('SENDGRID_API_KEY', '')
if len(SENDGRID_API_KEY) > 0:
    EMAIL_BACKEND = "django.core.mail.backends.smtp.EmailBackend"
    DEFAULT_FROM_EMAIL = env('SENDGRID_FROM_EMAIL', '')

    EMAIL_HOST = 'smtp.sendgrid.net'
    EMAIL_HOST_USER = 'apikey' # this is exactly the value 'apikey'
    EMAIL_HOST_PASSWORD = SENDGRID_API_KEY
    EMAIL_PORT = 587
    EMAIL_USE_TLS = True
else:
    EMAIL_BACKEND = "django.core.mail.backends.console.EmailBackend"
```

59

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

59

Custom emails

1. Create folder **templates/account/email**
2. Add files:
 1. email_confirmation_subject.txt
 2. email_confirmation_message.txt
3. Create **templates/account/email_confirm.html**
4. Get SendGrid API Key
5. Update **config/settings.py**
6. Update **docker-compose-prod.yml**

60



 IEEE Student Branch
 KU Leuven Campus Brugge

60

docker-compose-prod.yml

```
environment:
```

```
...
```

- "SENDGRID_API_KEY=<value>"
- "SENDGRID_FROM_EMAIL=<value>"

61



 IEEE Student Branch
 KU Leuven Campus Brugge

61

Custom emails

1. Create folder **templates/account/email**
2. Add files:
 1. email_confirmation_subject.txt
 2. email_confirmation_message.txt
3. Create **templates/account/email_confirm.html**
4. Get SendGrid API Key
5. Update **config/settings.py**
6. Update **docker-compose-prod.yml**
7. Execute: **docker-compose -f docker-compose-prod.yml up -d --build**

62



 IEEE Student Branch
 KU Leuven Campus Brugge

62

Git checkpoint

1. Commit changes
2. Create PR

63



 IEEE Student Branch
 KU Leuven Campus Brugge

63

Questions?

64

GitHub**IEEE Student Branch**
KU Leuven Campus Brugge

64

4. Integrating payments using Mollie

65

Mollie

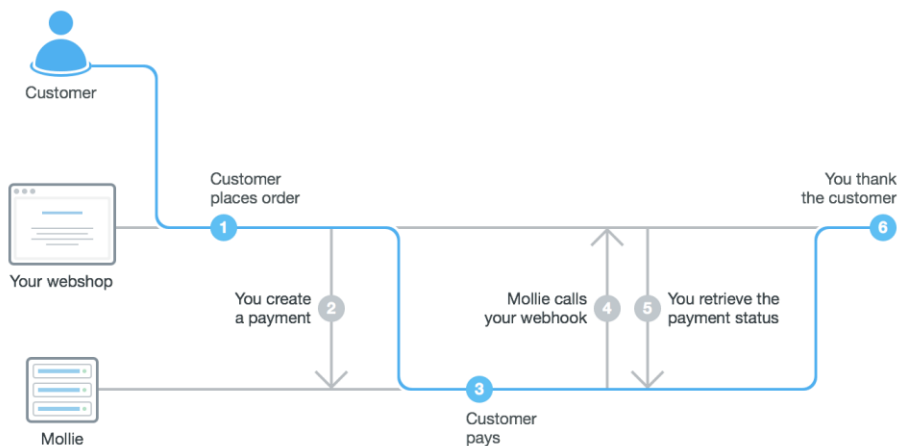
- Sign up: <https://my.mollie.com/dashboard/signup?lang=en>
- Developers > API Keys > Test API Key
- How do payments work?
 - source: <https://docs.mollie.com/payments/accepting-payments>

66


 IEEE Student Branch
 KU Leuven Campus Brugge

66

Mollie Payments API



67


 IEEE Student Branch
 KU Leuven Campus Brugge

67

Git checkpoint

1. Create new feature branch

68

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

68

Create webhook endpoint for Mollie

1. Pip install ***mollie-api-python*** and ***django-restframework*** & freeze
2. Execute: ***docker-compose exec web python manage.py startapp api***
3. Update ***api/views.py***

69

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

69

api/views.py (1)

```
from django.http import HttpResponse
from django.conf import settings

from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status

from mollie.api.client import Client

from accounts.models import CustomUser
from albums.models import Album
```

70




IEEE Student Branch
KU Leuven Campus Brugge

70

api/views.py (2)

```
class MolliePaymentView(APIView):
    def post(self, request, format=None):

        id = request.POST.get('id')
        if id is None:
            return Response(status=status.HTTP_400_BAD_REQUEST)

        mollie_client = Client()
        mollie_client.set_api_key(settings.MOLLIE_API_KEY)

        payment = mollie_client.payments.get(id)
        if payment.is_paid():
            user = CustomUser.objects.get(id=payment.metadata['userid'])
            album = Album.objects.get(id=payment.metadata['albumid'])
            user.albums.add(album)

        return HttpResponse(status=status.HTTP_200_OK)
```

71




IEEE Student Branch
KU Leuven Campus Brugge

71

Create webhook endpoint for Mollie

1. Pip install **mollie-api-python** and **django-rest-framework** and freeze
2. Execute: **docker-compose exec web python manage.py startapp api**
3. Update **api/views.py**
4. Add **api/urls.py**

72

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

72

api/urls.py

```
from django.urls import path

from .views import MolliePaymentView

urlpatterns = [
    path("mollie-payment", MolliePaymentView.as_view(), name="mollie_payment")
]
```

73

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

73

Create webhook endpoint for Mollie

1. Pip install **mollie-api-python** and **django-restframework** and freeze
2. Execute: **`docker-compose exec web python manage.py startapp api`**
3. Update **`api/views.py`**
4. Add **`api/urls.py`**
5. Update **`config/urls.py`**

74



 IEEE Student Branch
 KU Leuven Campus Brugge

74

`config/urls.py`

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('allauth.urls')),

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

    path('', include('pages.urls')),
    path('albums/', include('albums.urls')),
    path('accounts/', include('accounts.urls'))
] + static(
    settings.MEDIA_URL, document_root=settings.MEDIA_ROOT
)
```

75



 IEEE Student Branch
 KU Leuven Campus Brugge

75

Create webhook endpoint for Mollie

1. Pip install **mollie-api-python** and **django-rest-framework** and freeze
2. Execute: **docker-compose exec web python manage.py startapp api**
3. Update **api/views.py**
4. Add **api/urls.py**
5. Update **config/urls.py**
6. Update **config/settings/base.py**

76

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

76

config/settings/base.py

```
INSTALLED_APPS = [
    ...
    'rest_framework',
    ...
    'api.apps.ApiConfig'
]
...
MOLLIE_API_KEY = env('MOLLIE_API_KEY', '')
```

77

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

77

Create webhook endpoint for Mollie

1. Pip install **mollie-api-python** and **django-restframework** and freeze
2. Execute: **docker-compose exec web python manage.py startapp api**
3. Update **api/views.py**
4. Add **api/urls.py**
5. Update **config/urls.py**
6. Update **config/settings/base.py**
7. Add **MOLLIE_API_KEY** to **env vars** of **docker-compose-prod.yml**

78

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

78

Tests

1. Pip install **mock** & freeze
2. Update **config/settings/test.py**

79

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

79

config/settings/test.py

```
from .base import *

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'testdb'
    }
}

DEBUG = False

MOLLIE_API_KEY = "test_thisisanapikey"

SECURE_SSL_REDIRECT = False
SECURE_HSTS_SECONDS = 0
SECURE_HSTS_INCLUDE_SUBDOMAINS = False
SECURE_HSTS_PRELOAD = False
SESSION_COOKIE_SECURE = False
CSRF_COOKIE_SECURE = False

```

80

GitHub



IEEE Student Branch
KU Leuven Campus Brugge

80

Tests

1. Pip install **mock** and freeze
2. Update **config/settings/test.py**
3. Update **api/tests.py**

81

GitHub



IEEE Student Branch
KU Leuven Campus Brugge

81

api/tests.py (1)

```
import mock
import pytest
from django.test import TestCase
from django.urls import reverse
from django.contrib.auth import get_user_model

from mollie.api.resources.payments import Payments
from mollie.api.objects.payment import Payment

from albums.models import Album, Artist
```

82




IEEE Student Branch
KU Leuven Campus Brugge

82

api/tests.py (2)

```
@pytest.mark.django_db
class ApiTests(TestCase):
    @classmethod
    def setUpTestData(cls):
        cls.user = get_user_model().objects.create_user(
            username="test user",
            email="test@example.com",
            password="testpass123"
        )

        cls.artist = Artist.objects.create(
            name = "Beartooth"
        )
```

83




IEEE Student Branch
KU Leuven Campus Brugge

83

api/tests.py (3)

```
cls.album = Album.objects.create(
    title = "Below",
    price = "15.99",
    record_label = "Red Bull Records"
)

cls.artist.albums.add(cls.album)
```

84




IEEE Student Branch
KU Leuven Campus Brugge

84

api/tests.py (4)

```
@mock.patch.object(Payments, 'get', autospec=True)
def test_mollie_payment_webhook(self, mock_payments_get):
    payment_data = {
        'paidAt': 'now',
        'metadata': {
            'userid': self.user.id,
            'albumid': self.album.id
        }
    }
    mock_payments_get.return_value = Payment(payment_data)
```

85




IEEE Student Branch
KU Leuven Campus Brugge

85

api/tests.py (5)

```
data = {  
    'id': 'tr_thisisanid'  
}  
  
response = self.client.post(reverse("mollie_payment"), data)  
  
assert response.status_code == 200  
assert self.user.albums.all()[0].title == "Below"
```

86

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

86

Add payment button

1. Update **albums/models.py**

87

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

87

albums/models.py (1)

```
import uuid
from django.db import models
from django.conf import settings
from django.urls import reverse

from mollie.api.client import Client

from common.models import TimeStampedModel

class Album(TimeStampedModel):
    ...
```

88




IEEE Student Branch
KU Leuven Campus Brugge

88

albums/models.py (2)

```
def get_payment_url(self, request):
    if len(settings.MOLLIE_API_KEY) < 0:
        return ""

    client = Client()
    client.set_api_key(settings.MOLLIE_API_KEY)

    payment = client.payments.create({
        'amount': {
            'currency': 'EUR',
            'value': str(self.price)
        },
```

89




IEEE Student Branch
KU Leuven Campus Brugge

89

albums/models.py (3)

```
'metadata': {
    'user_id': request.user.id,
    'album_id': str(self.id)
},
'description': f"Payment of {self.title}",
'redirectUrl': request.build_absolute_uri(reverse("user_album_list")),
'webhookUrl': request.build_absolute_uri(reverse("mollie_payment"))
})

return payment.checkout_url
```

90




IEEE Student Branch
KU Leuven Campus Brugge

90

Add payment button

1. Update **albums/models.py**
2. Update **albums/views.py**

91




IEEE Student Branch
KU Leuven Campus Brugge

91

albums/views.py

```
class AlbumDetailView(DetailView):
    model = Album
    context_object_name = "album"
    template_name = "albums/album_detail.html"

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)

        if self.request.user.is_authenticated:
            context['payment_url'] = self.get_object().get_payment_url(self.request)
        return context
```

92




IEEE Student Branch
KU Leuven Campus Brugge

92

Add payment button

1. Update **albums/models.py**
2. Update **albums/views.py**
3. Update **templates/albums/album_detail.html**

93




IEEE Student Branch
KU Leuven Campus Brugge

93

templates/albums/album_detail.html

```
<p>Price: €{{ album.price }}</p>

{% if user.is_authenticated and not album in user.albums %}
    <button onclick="location.href='{{ payment_url }}'">Buy now</button>
{% endif %}
```

94




IEEE Student Branch
KU Leuven Campus Brugge

94

Git checkpoint

1. Commit changes
2. Create PR

95




IEEE Student Branch
KU Leuven Campus Brugge

95

Questions?

96

GitHub**IEEE Student Branch**
KU Leuven Campus Brugge