

n

Contents of last session

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

GitHub



_ 1

Contents of this session

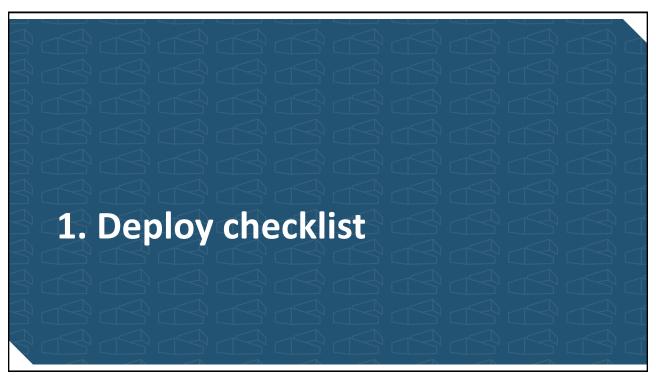
- 1. Deploy checklist
- 2. Performance
- 3. Sentry

2





2



Deploy checklist

- 1. Execute: docker-compose exec web python manage.py check --deploy
 - 1. A couple of issues will pop up:
 - 1. SECURE_HSTS_SECONDS
 - 2. SECURE_SSL_REDIRECT
 - 3. SECRET_KEY
 - 4. SESSION_COOKIE_SECURE
 - 5. CSRF_COOKIE_SECURE
 - 6. DEBUG

GitHub



Git checkpoint

1. Create new feature branch

GitHub <



Solve the issues

1. Update config/settings/base.py

6





config/settings/base.py

```
SECURE_SSL_REDIRECT = env.bool("DJANGO_SECURE_SSL_REDIRECT", default=True)
SECURE_HSTS_SECONDS = env.int("DJANGO_SECURE_HSTS_SECONDS", default=2592000)
SECURE_HSTS_INCLUDE_SUBDOMAINS = env.bool("DJANGO_SECURE_HSTS_INCLUDE_SUBDOMAINS", default=True)
SECURE_HSTS_PRELOAD = env.bool("DJANGO_SECURE_HSTS_PRELOAD", default=True)
SESSION_COOKIE_SECURE = env.bool("DJANGO_SESSION_COOKIE_SECURE", default=True)
CSRF_COOKIE_SECURE = env.bool("DJANGO_CSRF_COOKIE_SECURE", default=True)
```



Solve the issues

- 1. Update config/settings/base.py
- 2. Update docker-compose.yml

8





8

docker-compose.yml

depends on:

- db

environment:

- "DJANGO SECRET KEY=<value>"
- "DJANGO DEBUG=True"
- "DJANGO_SECURE_SSL_REDIRECT=False"
- "DJANGO SECURE HSTS SECONDS=0"
- "DJANGO_SECURE_HSTS_INCLUDE_SUBDOMAINS=False"
- "DJANGO_SECURE_HSTS_PRELOAD=False"
- "DJANGO SESSION COOKIE SECURE=False"
- "DJANGO_CSRF_COOKIE_SECURE=False"





Solve the issues

- 1. Update config/settings/base.py
- 2. Update docker-compose.yml
- 3. Update **DJANGO_SECRET_KEY** in **docker-compose-prod.yml**:
 - 1. Generate DJANGO_SECRET_KEY using:
 - 1. docker-compose exec web python -c "import secrets; print(secrets.token_urlsafe(38))"
- 4. Execute: docker-compose -f docker-compose-prod.yml up -d --build
- 5. Execute: docker-compose exec web python manage.py check --deploy
- 6. Also: pip install gunicorn & freeze

10





10

Git checkpoint

- 1. Commit changes
- 2. Create PR

GitHub



11

Questions?

GitHub



12

12



Git checkpoint

1. Create new feature branch

14





14

Django debug toolbar

- 1. Pip install django-debug-toolbar & freeze
- 2. Update config/settings/base.py





config/settings/base.py

16

Django debug toolbar

- 1. Pip install django-debug-toolbar & freeze
- 2. Update config/settings/base.py
- 3. Update config/urls.py

GitHub



config/urls.py

```
if settings.DEBUG:
    import debug_toolbar

urlpatterns = [
      path("__debug__/", include(debug_toolbar.urls))
] + urlpatterns
```

18





18

Django debug toolbar important features

- History: history of called endpoints
- Versions: versions of Django, Python and installed apps
- Time: timeline of page load
- Settings: Django settings
- Request: info about this request
- SQL: info on executed queries to load this page
 - Query
 - Query Plan
 - Execution Time
- Static files: info on used static files
- Templates: used templates + context data





Git checkpoint

- 1. Commit changes
- 2. Create PR

20



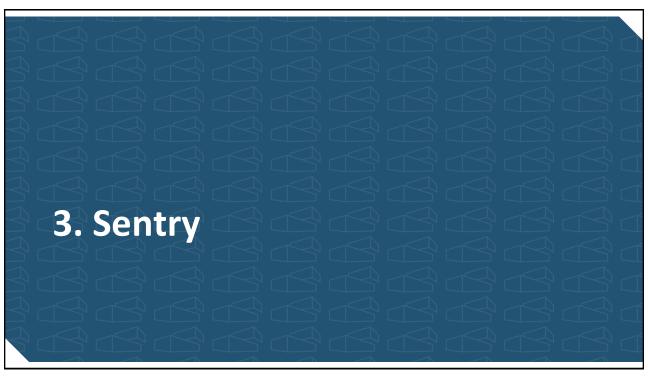


20

Questions?







22

Git checkpoint

1. Create new feature branch





Sentry setup

- 1. Go to: https://sentry.io/
- 2. Create account
- 3. Create project:
 - 1. Select Django
 - 2. Other settings you can tweak later
 - 3. Follow guide on Sentry (also the zero div)
 - 1. Freeze pip
 - 2. Create SENTRY_DSN environment variable
 - 3. Wrap sentry_sdk.init in a if checking if DSN is filled in
 - 1. To make sure tests aren't slowed down by useless init of Sentry

24





24

Sentry features

- Issues:
 - Shows info on user, client, cookies ... → useful side conditions for error resolving
 - Pinpoints at which exact code line the error occurred
- Performance:
 - Apdex:
 - · Open standard for measuring software performance based on expected user satisfaction
 - 0: No users satisfied; 1: All users satisfied
 - Based on configured goal time: settings > projects > [this project] > performance > RTT
 - Single event data:
 - Waterfall with SQL; sort of a remote Django Debug Toolbar





Git checkpoint

- 1. Commit changes
- 2. Create PR

26





26

Questions?



