

Django

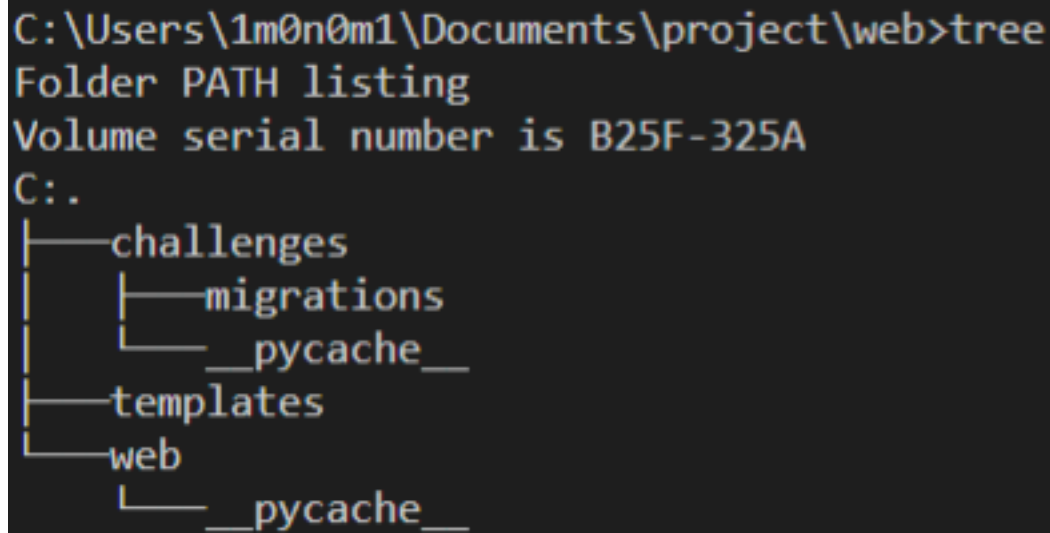
```
pytho
mkdir project
cd project
python -m pip install django
python -m django --version
django-admin startproject web
cd web
python manage.py runserver and http://localhost:8000
```

```
python manage.py startapp challenges
Modificación de vistas en challenges/views.py
```

Estructura clave para el entendimiento.

```
web/
  challenges/ <- App
  web/        <- Configuración del proyecto
```

Creación de carpeta con templates en la raíz de web/



```
C:\Users\1m0n0m1\Documents\project\web>tree
Folder PATH listing
Volume serial number is B25F-325A
C:..
|---challenges
|   |---migrations
|   |---__pycache__
|---templates
|---web
|   |---__pycache__
```

Asignación de ruta a templates en la palabra , alterando web/settings.py

```
settings.py × views.py login.html
web > settings.py > ...
52 ROOT_URLCONF = 'web.urls'
53
54 ∨ TEMPLATES = [
55 ∨ {
56     'BACKEND': 'django.template.backends.django.DjangoTemplates',
57 ∨     'DIRS': [
58         os.path.join(os.path.join(BASE_DIR, 'templates')),
59     ],
60     'APP_DIRS': True,
61 ∨     'OPTIONS': {
62 ∨         'context_processors': [
63             'django.template.context_processors.debug',
```

Configuración de app para archivos estaticos, alterando web/settings.py
django.contrib.staticfiles

```
settings.py × views.py login.html
web > settings.py > ...
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40 ]
41
```

Para servir scripts personalizados, alterando web/settings.py

```
settings.py × JS index.js views.py login.html urls.py
web > settings.py > ...
121
122 # Static files (CSS, JavaScript, Images)
123 # https://docs.djangoproject.com/en/3.0/howto/static-files
124 STATIC_ROOT = '/'
125 STATIC_URL = '/static/'
126
127 STATICFILES_DIRS = [
128     os.path.join(os.path.join(BASE_DIR, "static")),
129 ]
```

Alterando challenges/urls.py

```
settings.py JS index.js views.py login.html urls.py ●
challenges > urls.py > ...
1 from . import views
2 from django.urls import path
3 from django.conf import settings
4 from django.conf.urls.static import static
5
6 urlpatterns = [
7     path('', views.index, name='index'),
8 ] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
9
10
```

Todas

Todas las peticiones son por defecto aseguradas contra csrf, por motivo de velocidad y de facilidad lo vamos a deshabilitar en cada funcion que implemente el methodo post

```
settings.py  JS index.js  views.py  X  login.html  urls.py
challenges > views.py > ...
4  from django.template.loader import render_to_string
5  from django.views.decorators.csrf import csrf_exempt
6  # Create your views here.
7  @csrf_exempt
8  def index(request):
9      context = {}
10     #return render(request, 'challenges/index.html', context)
11     if request.method == 'GET':
12         return render(request, "login.html", context)
13     else:
14         return "WO"
```

Se configura el login para que envíe a través del método POST, y además se agregan los nombres para poder recibir los keys con django

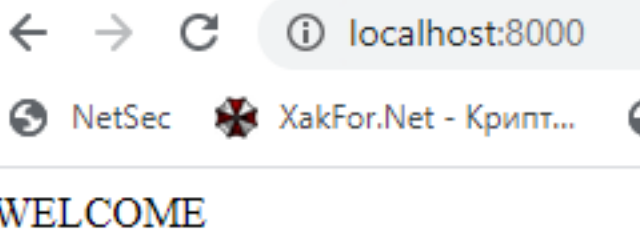
```
settings.py  JS index.js  views.py  login.html X  urls.py
templates > login.html > html > body.text-center > form.form-signin > input#inputPassword.form-cont
78  <!-- Custom styles for this template -->
79  <!-- <script type="text/javascript" src="index.js"></script> -->
80  head>
81  ody class="text-center">
82  <form class="form-signin" method="post">
83  mg class="mb-4" src="https://static.thenounproject.com/png/17611-
84  1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
85  abel for="inputUser" class="sr-only">Username</label>
86  nput type="text" id="inputUser" class="form-control" name="username"
87  abel for="inputPassword" class="sr-only">Password</label>
88  nput type="password" id="inputPassword" name="password" class="fo
89  iv class="checkbox mb-3">
90  <label>
91  |   <input type="checkbox" value="remember-me"> Remember me
92  </label>
```

El login se debería ver algo así:

```
def index(request):
    context = {
        'message': ""
    }
    #return render(request, 'challenges/index.html', context)
    if request.method == 'GET':
        return render(request, "login.html", context)
    else:
        username = request.POST.get('username')
        password = request.POST.get('password')

        if not username or not password:
            context['message'] = 'Usuario y clave obligatorios!!'
            return render(request, "login.html", context)
        elif username == "admin" and password == "123456":
            return HttpResponse("WELCOME")
        else:
            context['message'] = 'Usuario y/o clave incorrectos!!'
            return render(request, "login.html", context)
```

Asi se ve cuando las credenciales son validas



Asi cuando no son validas.

Please sign in

☐ Remember me

Usuario y/o clave incorrectos!!

Sign in

© 2017-2020

Configurando el componente de urls

```
settings.py  JS index.js  views.py  <> login.html  urls.py  ×  ▶
challenges > urls.py > ...
1  from . import views
2  from django.urls import path
3  from django.conf import settings
4  from django.conf.urls.static import static
5
6  urlpatterns = [
7      path('', views.index, name='index'),
8      path('index', views.index, name='index'),
9      path('dashboard', views.dashboard, name='dashboard'),
10 ] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
11
12
```

para autenticarse

```

elif username == "admin" and password == "123456":
    request.session['logged'] = 1
    return redirect('dashboard')
else:

```

para controlar la autenticación

```

def dashboard(request):
    sess = request.session.get('logged')
    if sess == 1:
        return HttpResponse("WELCOME")
    else:
        return redirect('index')

```

```

[25/Jul/2020:12:15:55] "POST / HTTP/1.1" 300 135261
PS C:\Users\lm0n0m1\Documents\project\web> python .\manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
PS C:\Users\lm0n0m1\Documents\project\web> python .\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

```

Prueba despues de eso.

WELCOME

Modificando un poco la vista y el template

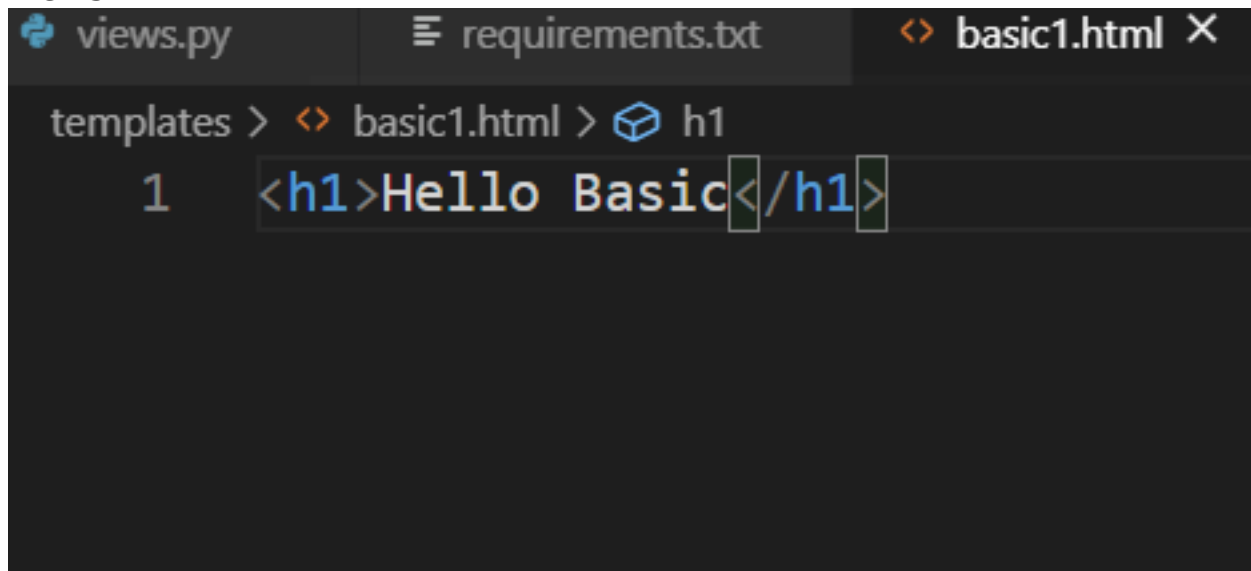
urls:

```
JS index.js  views.py  basic1.html  dashboard.html  login.html  urls.py  ×  ▶
challenges > urls.py > ...
1  from . import views
2  from django.urls import path
3  from django.conf import settings
4  from django.conf.urls.static import static
5
6  urlpatterns = [
7      path('', views.index, name='index'),
8      path('index', views.index, name='index'),
9      path('dashboard', views.dashboard, name='dashboard'),
10     path('basic1', views.basic1, name='basic1'),
11 ] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
12
13
```

vista:

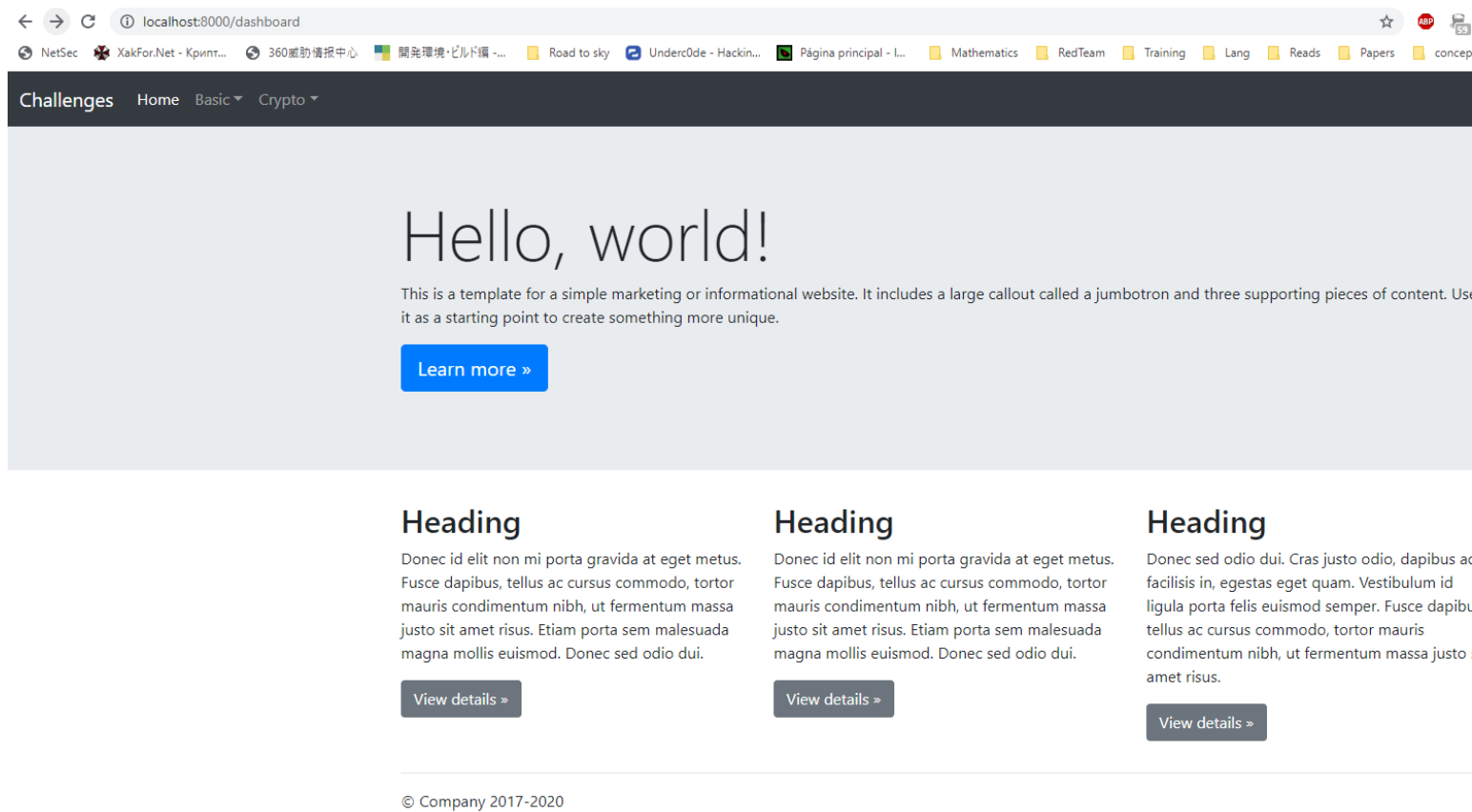

```
31 def dashboard(request):
32     sess = request.session.get('logged')
33     if sess != 1:
34         return redirect('index')
35     context = {}
36     return render(request, "dashboard.html", context)
37
38
39 def basic1(request):
40     sess = request.session.get('logged')
41     if sess != 1:
42         return redirect('index')
43     context = {}
44     return render(request, "basic1.html", context)
```

views

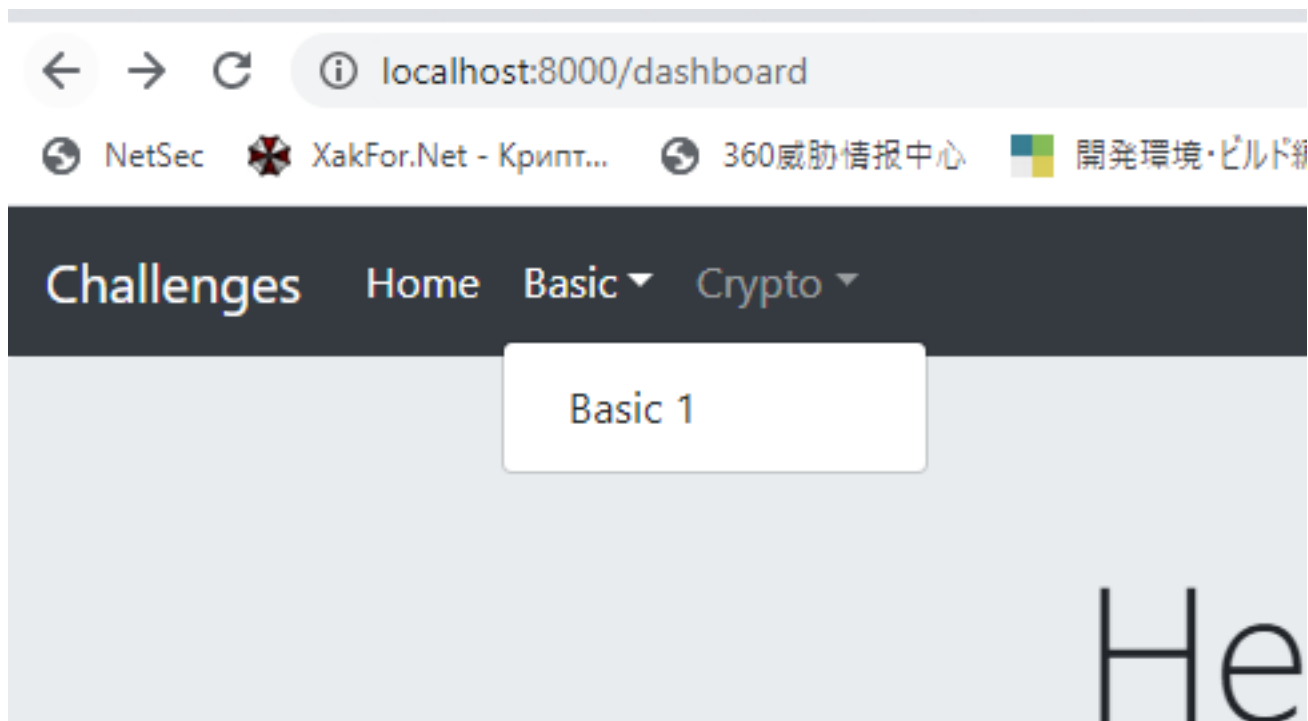


```
views.py requirements.txt basic1.html X
templates > basic1.html > h1
1 <h1>Hello Basic</h1>
```

template:



challenge 1



Hello Basic

asasd
asd
asd