

<https://www.halvorsen.blog>

django

Django Sessions

Hans-Petter Halvorsen



Contents

- [Introduction](#)
- [Start using Sessions in Django](#)
- [Session Example](#)

<https://www.halvorsen.blog>

django

Introduction



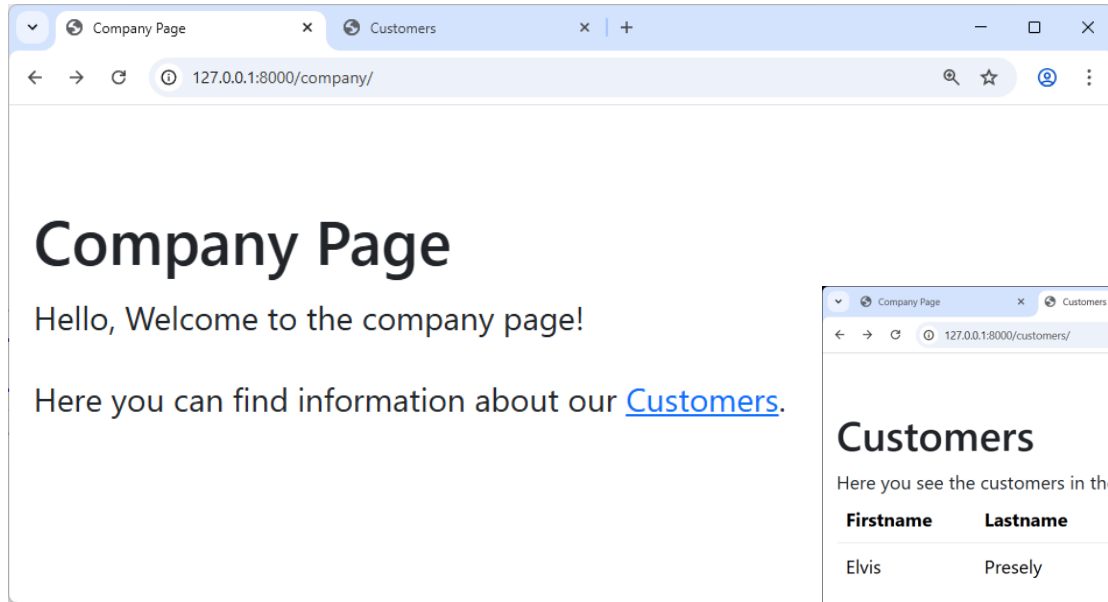
Hans-Petter Halvorsen

[Table of Contents](#)

Sessions

- **HTTP is a stateless protocol.** Meaning it does not remember anything from one web page to another
- We will see how we can use something called **Session variables** for sharing information between different web pages.

Company App

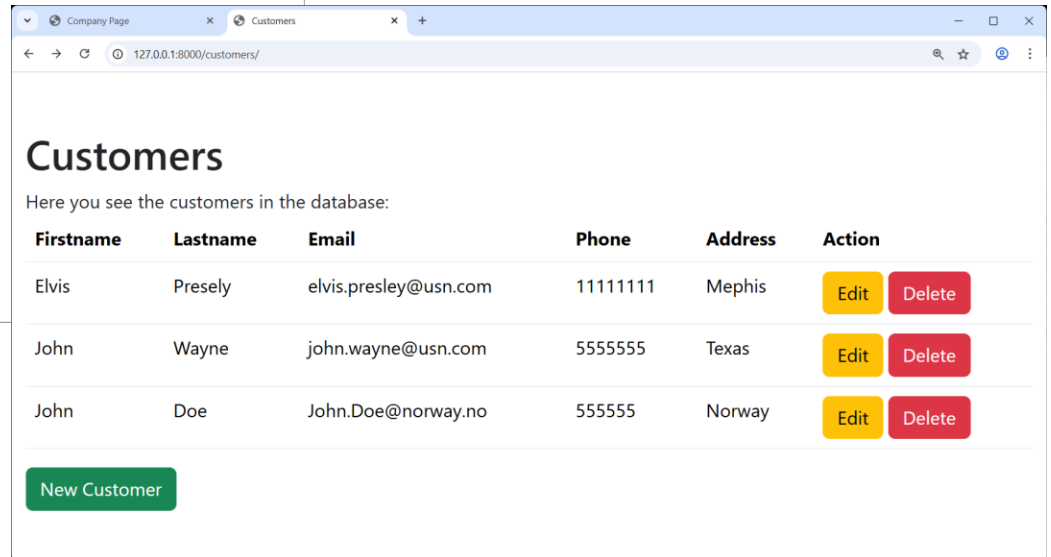


The screenshot shows a web browser window with two tabs: 'Company Page' and 'Customers'. The address bar displays '127.0.0.1:8000/company/'. The main content area features a large heading 'Company Page', a welcome message 'Hello, Welcome to the company page!', and a link 'Customers'.

Company Page

Hello, Welcome to the company page!

Here you can find information about our [Customers](#).



The screenshot shows a web browser window with two tabs: 'Company Page' and 'Customers'. The address bar displays '127.0.0.1:8000/customers/'. The main content area features a heading 'Customers', a message 'Here you see the customers in the database:', a table with customer data, and a 'New Customer' button.

Customers

Here you see the customers in the database:

Firstname	Lastname	Email	Phone	Address	Action
Elvis	Presley	elvis.presley@usn.com	11111111	Mephis	Edit Delete
John	Wayne	john.wayne@usn.com	5555555	Texas	Edit Delete
John	Doe	John.Doe@norway.no	555555	Norway	Edit Delete

[New Customer](#)

Resources

- Django Tutorial Part 7: Sessions framework:
https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Django/Sessions
- Sessions framework using django – Python:
<https://www.geeksforgeeks.org/python/python-sessions-framework-using-django/>

<https://www.halvorsen.blog>

django

Start using Sessions in Django



Hans-Petter Halvorsen

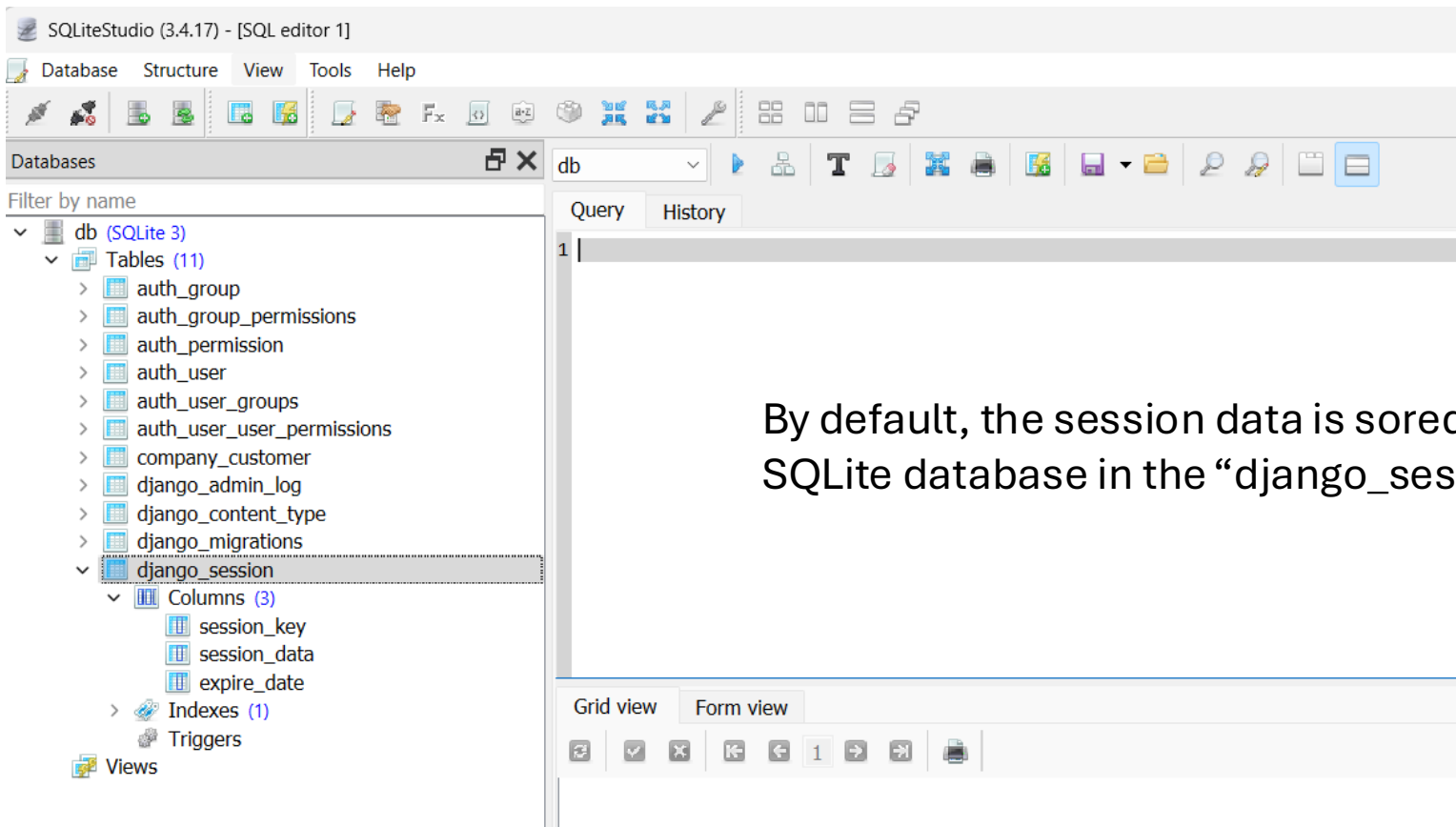
[Table of Contents](#)

Enable Sessions

```
settings.py ×
djangoproject1 > settings.py > ...
31 # Application definition
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     'company'
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
```

Make sure Sessions is enabled in “settings.py”. These should be included when creating the Django project, if not make sure to include these lines as seen in the image.

Database



By default, the session data is stored in the SQLite database in the “django_session” table.

<https://www.halvorsen.blog>

django

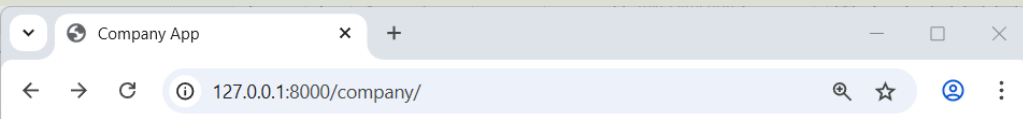
Session Example



Hans-Petter Halvorsen

[Table of Contents](#)

New Company App



Company Home Page

Hello, Welcome to **USN**.

Here you see our [customers](#).

A screenshot of a web browser window. The title bar says "Customers". The address bar shows "127.0.0.1:8000/customers/". The page content includes a heading "Customers", a red box highlighting the text "Here you see a list of customers for company USN:", a table with customer data, and a "New Customer" button at the bottom.

Firstname	Lastname	Email	Phone	Address	Action
Elvis	Presley	elvis.presley@usn.com	111111	USA	Edit Delete
Donald	Trump	donald.trump@usn.com	222222	USA	Edit Delete
John	Doe	john.doe@usn.com	333333	Norway	Edit Delete
Hans-Petter	Halvorsen	hans.p.halvorsen@usn.no	11111111	Norway	Edit Delete

[New Customer](#)

“company” View

```
def company(request):  
    companyname = "USN"  
    request.session["companyname"] = companyname  
    context = {'companyname': companyname,}  
    return render(request, "company.html", context)
```

“company.html” Template

```
<!DOCTYPE html>
<html>
  <head>
    <title>Company App</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>
    <div class="container-fluid pt-5">
      <h1>Company Home Page</h1>
      <p>Hello, Welcome to <b>{{ companyname }}</b>.</p>
      <p>Here you see our <a href="/customers">customers</a>.</p>
    </div>
  </body>
</html>
```

“customers” View

```
def customers(request):
    customers = Customer.objects.all()
    template = loader.get_template('customers.html')
    companyname = request.session["companyname"]
    context = {
        'customers': customers,
        'companyname': companyname,
    }
    return HttpResponse(template.render(context, request))
```

“customers.html” Template

```
<!DOCTYPE html>
<html>
<head>
  <title>Customers</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</head>

<body>
  <div class="container-fluid pt-5">
    <h1>Customers</h1>
    <p>Here you see a list of customers for company <b>{{ companyname }}</b></p>
    <div class="table-responsive">
      <table class="table">
        <thead>
          <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Email</th>
            <th>Phone</th>
            <th>Address</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody>
          <% for customer in customers %>
          <tr>
            <td>{{ customer.first_name }}</td>
            <td>{{ customer.last_name }}</td>
            <td>{{ customer.email }}</td>
            <td>{{ customer.phone }}</td>
            <td>{{ customer.address }}</td>
            <td>
              <a href="/customers/update/{{ customer.pk }}" class="btn btn-warning">Edit</a>
              <a href="/customers/delete/{{ customer.pk }}" class="btn btn-danger">Delete</a>
            </td>
          </tr>
          <% endfor %>
        </tbody>
      </table>
      <a href="/customers/new" class="btn btn-success">New Customer</a>
    </div>
  </body>
</html>
```

Get CompanyName from Database

We create a new “Company” Model

```
class Company(models.Model):  
    companyname = models.CharField(max_length=100)  
    email = models.CharField(max_length=100)  
    website = models.CharField(max_length=100)  
  
    def __str__(self):  
        return self.companyname
```

We need to use “Migrate” to update the Database and make the necessary Database Tables in the Database:

```
..>python manage.py makemigrations
```

```
..>python manage.py migrate
```

Django Admin

We can insert Company data directly in the database using, e.g., SQLiteStudio, or we can use the Django Admin interface.

```
admin.py ×
company > admin.py
1 from django.contrib import admin
2 from .models import Company
3 from .models import Customer
4
5 # Register your models here.
6 admin.site.register(Company)
7 admin.site.register(Customer)
```

Django Admin

A screenshot of the Django Admin interface. The browser address bar shows '127.0.0.1:8000/admin/'. The page header is a dark blue bar with the text 'Django administration' in yellow, followed by 'WELCOME, HANSP. VIEW SITE / CHANGE PASSWORD / LOG OUT' and a moon icon. Below the header, the text 'Site administration' is displayed. A dark blue bar contains the text 'AUTHENTICATION AND AUTHORIZATION'. Underneath, there are two rows: 'Groups' and 'Users', each with a green '+ Add' button and a yellow pencil 'Change' button. Another dark blue bar contains the text 'COMPANY'. Below it, there are two rows: 'Companies' and 'Customers', each with a green '+ Add' button and a yellow pencil 'Change' button.

A screenshot of the Django Admin 'Add company' form. The browser address bar shows '127.0.0.1:8000/admin/company/company/add/'. The page header is a dark blue bar with the text 'Django administration' in yellow, followed by 'WELCOME, HANSP. VIEW SITE / CHANGE PASSWORD / LOG OUT' and a moon icon. Below the header, a dark blue bar contains the breadcrumb 'Home > Company > Companys > Add company'. The main content area is white and contains the text 'Add company'. Below this, there are three form fields: 'Companyname:' with the value 'USN', 'Email:' with the value 'usn@usn.no', and 'Website:' with the value 'http://www.usn.no'. At the bottom of the form, there is a dark blue button with the text 'SAVE'.

“company” View

```
def company(request):  
    company = Company.objects.first()  
    request.session["companyname"] = company.companyname  
    context = {'company': company,}  
    return render(request, "company.html", context)
```

“company.html” Template

```
<!DOCTYPE html>
<html>
  <head>
    <title>Company App</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>
    <div class="container-fluid pt-5">
      <h1>Company Home Page</h1>
      <p>Hello, Welcome to <b>{{ company.companyname }}</b>.</p>
      <p>Email: {{ company.email }}</p>
      <p>Web: {{ company.website }}</p>
      <p>Here you see our <a href="/customers">customers</a>.</p>
    </div>
  </body>
</html>
```

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

