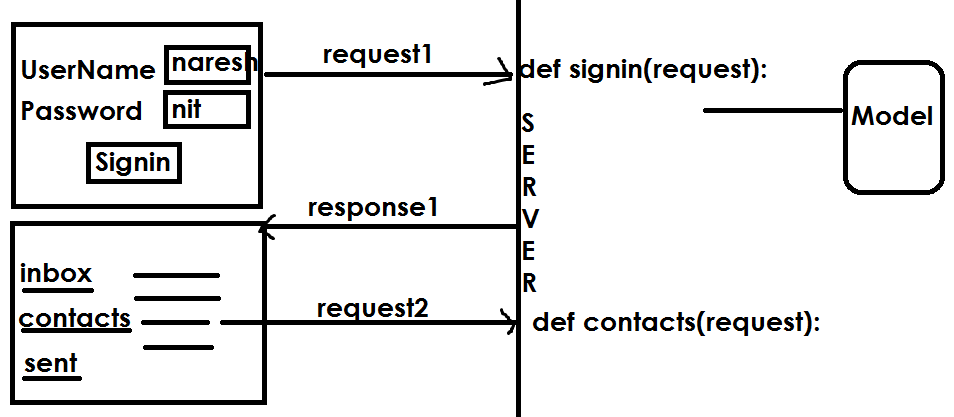
Session Management in Django

Http stands for Hyper Text Transfer Protocol. It is stateless protocol.

Every request send by client to server is a new request.

Server does not remember the request has come from which client.

The lifetime of request is until execution of view or until response is generated.



To maintain state between multiple requests of the same client we use session management or statement management

**Enabling sessions**

Sessions are implemented via a piece of [middleware](https://docs.djangoproject.com/en/4.1/ref/middleware/).

**To enable session functionality, do the following**:

Edit the [MIDDLEWARE](https://docs.djangoproject.com/en/4.1/ref/settings/#std-setting-MIDDLEWARE) setting and make sure it contains 'django.contrib.sessions.middleware.SessionMiddleware'. The default settings.py created by django-admin startproject has SessionMiddleware activated.

If you don’t want to use sessions, you might as well remove the SessionMiddleware line from [MIDDLEWARE](https://docs.djangoproject.com/en/4.1/ref/settings/#std-setting-MIDDLEWARE) and 'django.contrib.sessions' from your [INSTALLED\_APPS](https://docs.djangoproject.com/en/4.1/ref/settings/#std-setting-INSTALLED_APPS). It’ll save you a small bit of overhead.

**What is middleware?**

Middleware is a framework of hooks into Django's request/response processing. It's a light, low-level “plugin” system for globally altering Django's input or output. Each middleware component is responsible for doing some specific function.

## Configuring the session engine[¶](https://docs.djangoproject.com/en/4.1/topics/http/sessions/#configuring-the-session-engine)

By default, Django stores sessions in your database (using the model **django.contrib.sessions.models.Session**). Though this is convenient, in some setups it’s faster to store session data elsewhere, so Django can be configured to store session data on your filesystem or in your cache.

### Using database-backed sessions[¶](https://docs.djangoproject.com/en/4.1/topics/http/sessions/#using-database-backed-sessions)

If you want to use a database-backed session, you need to add **'django.contrib.sessions'** to your [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/4.1/ref/settings/#std-setting-INSTALLED_APPS) setting.

Once you have configured your installation, run **manage.py migrate** to install the single database table that stores session data.

## Using sessions in views[¶](https://docs.djangoproject.com/en/4.1/topics/http/sessions/#using-sessions-in-views)

When **SessionMiddleware** is activated, each **[HttpRequest](https://docs.djangoproject.com/en/4.1/ref/request-response/" \l "django.http.HttpRequest" \o "django.http.HttpRequest)** object – the first argument to any Django view function – will have a **session** attribute, which is a dictionary-like object.

You can read it and write to **request.session** at any point in your view. You can edit it multiple times.