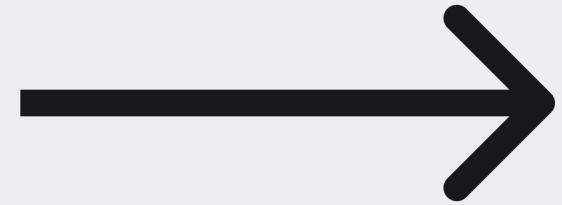


A Complete Overview of

Message

Framework

In Django



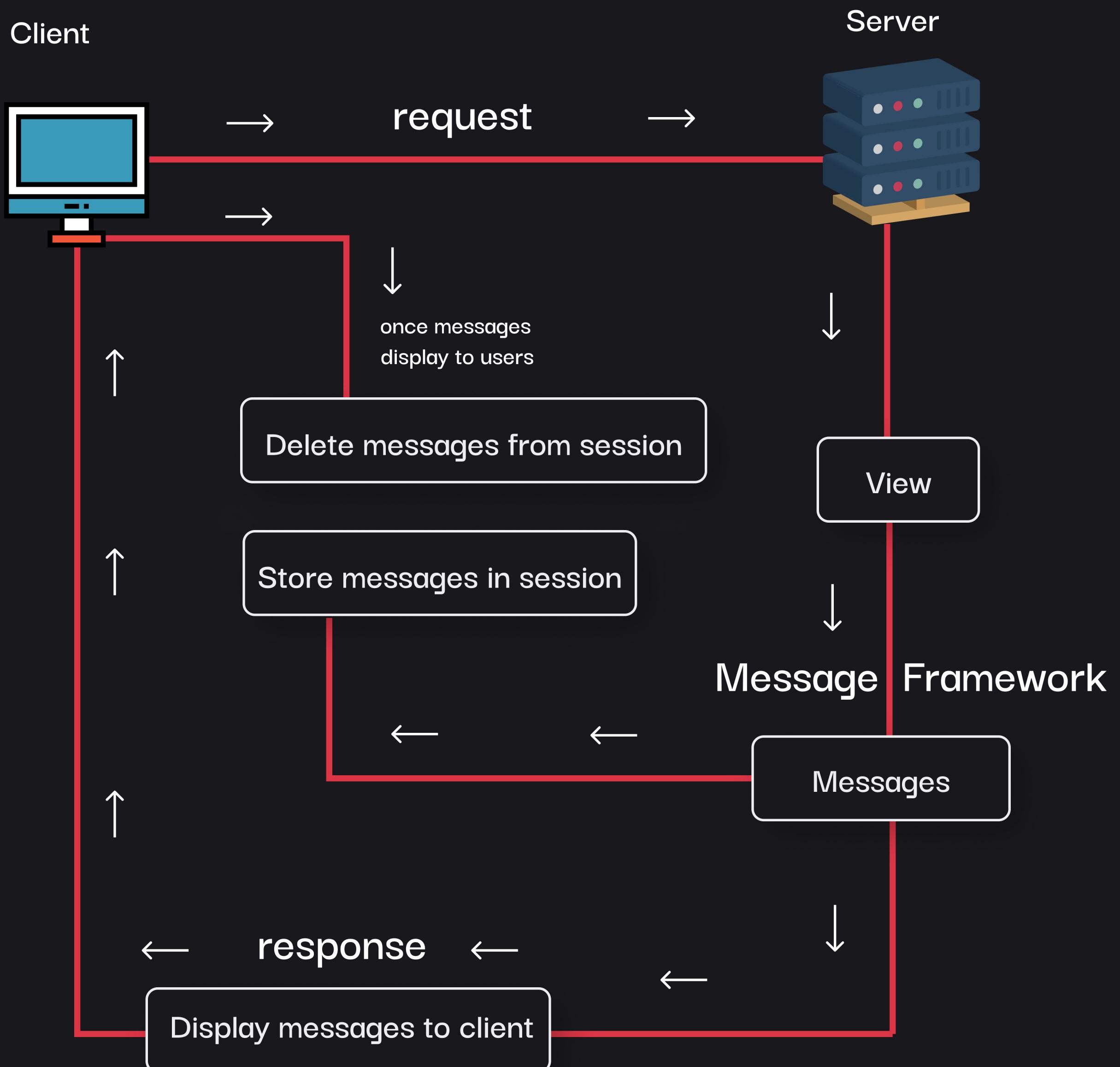
#python



Introduction

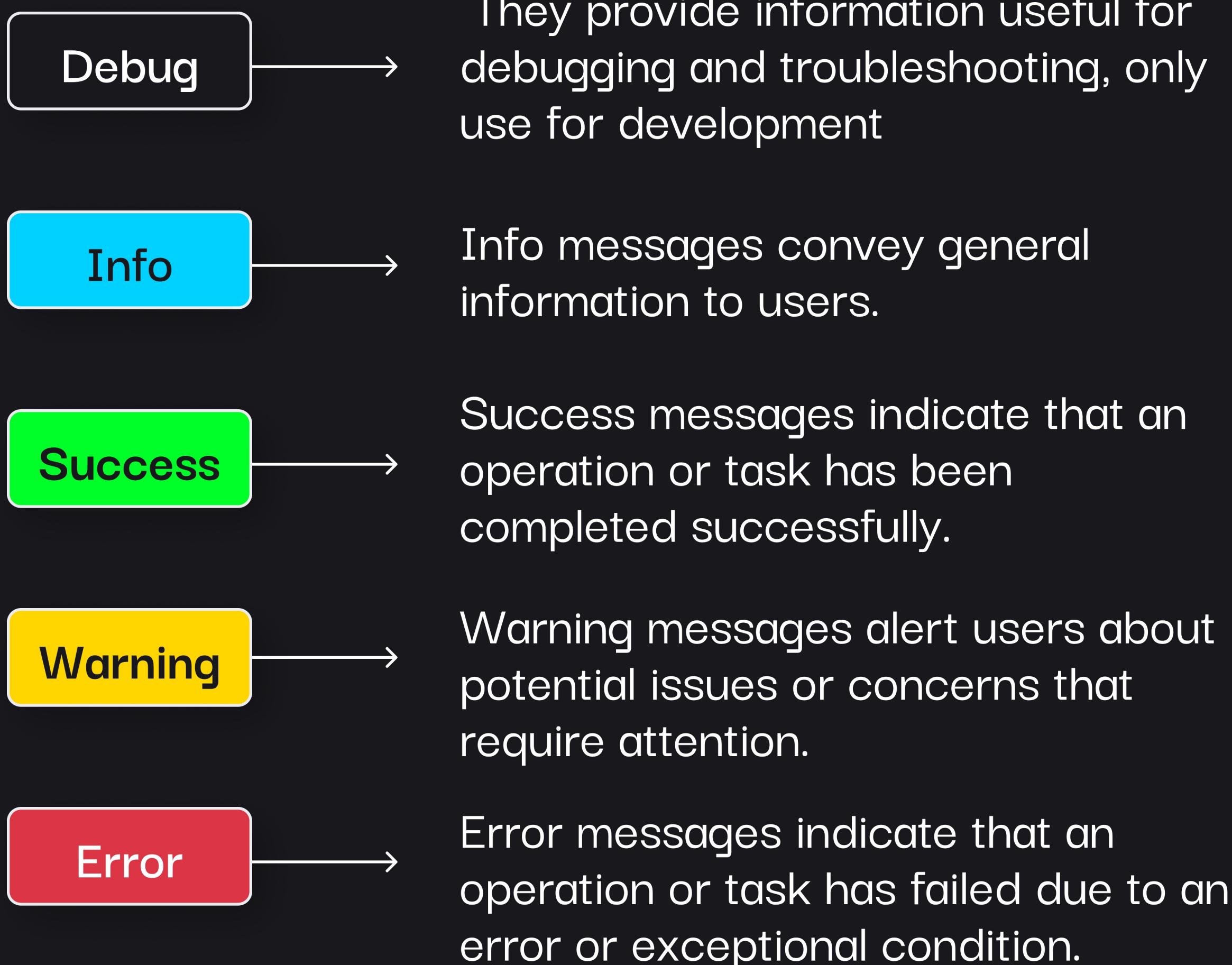
In Django, the message framework allows you to send messages from views to templates and display them to users. This is useful for providing feedback or notifications to users after certain actions, such as successful form submissions, errors, or other important events.

How Message Framework Works in Django



Different Levels of messages in Message Framework

Message Framework provides different levels of messages based on their importance or severity.



Let's use Message Framework in Django

To use message framework you need to first import messages from django.contrib package



```
from django.contrib import messages
from django.shortcuts import render

def Login(request):
    # Your login logic here

    →messages.success(request, "Login Successfully")

    return render(request, "login.html")
```

You can simply use the messages and set it's levels (debug, info, success, warning, error) and pass the request and the message you want to display to client.

Let's display our messages in templates :

To display the messages to client, just add the code in template file

checking if we have message or not

next we loop the messages, because we can have multiple messages

```
<html>
  <head>
    <title>Login</title>
  </head>
  <body>

    <!-- Display messages -->

    {% if messages %}
      {% for message in messages %}
        <div class="alert alert-{{message.tags}}">{{message}}</div>
      {% endfor %}
    {% endif %}

  </body>
</html>
```

You can also display the message tags or levels to give a suitable colours to messages

Here we simply display the message

Examples of different messages levels

DEBUG

```
def debug_view(request):  
    messages.debug(request, "This is debug level message")
```

INFO

```
def info_view(request):  
    messages.info(request, "This is info level message")
```

SUCCESS

```
def success_view(request):  
    messages.success(request, "This is success level message")
```

WARNING

```
def warning_view(request):  
    messages.warning(request, "This is warning level message")
```

ERROR

```
def error_view(request):  
    messages.error(request, "This is error level message")
```

Note some important points :

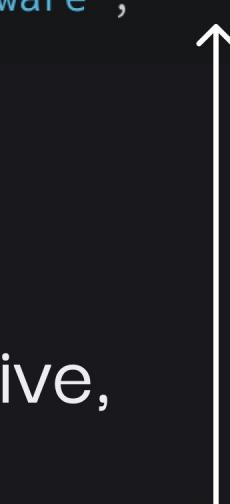
Make sure before use message framework in django these settings should be enabled.

```
TEMPLATES = [
    {
        "OPTIONS": {
            "context_processors": [
                "django.contrib.messages.context_processors.messages",
            ],
        },
    },
]
```



```
MIDDLEWARE = [
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
]
```

Make sure SessionMiddleware is active, this middleware helps to store messages in session

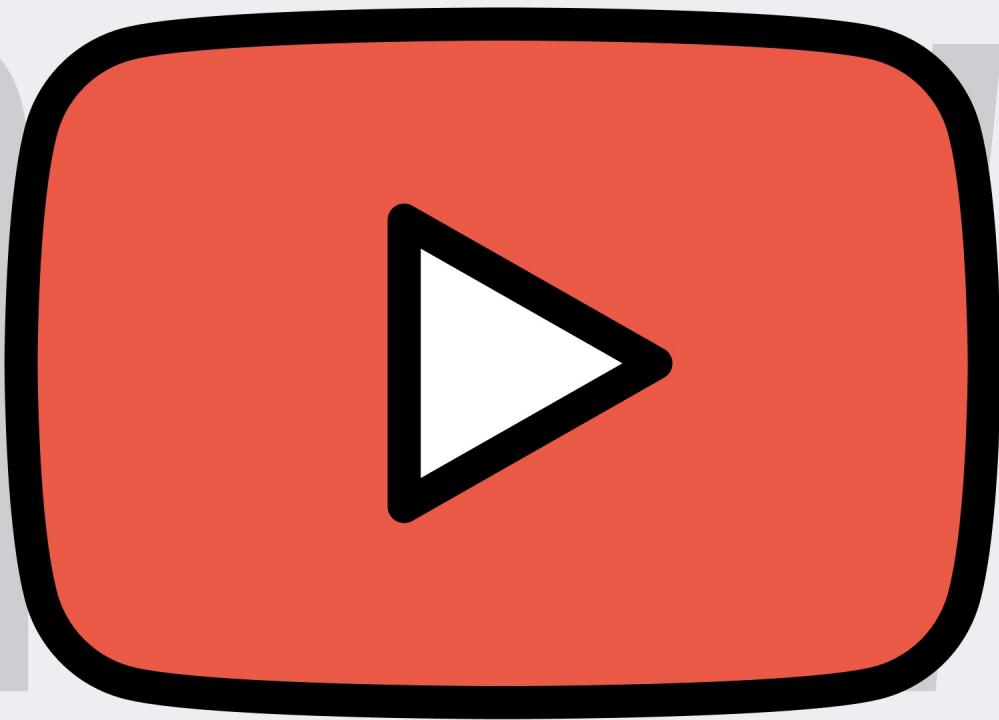


Make sure this messages context processors must be added under TEMPLATES, because of this we are able to use messages variable in templates and display messages

By default these settings are already present in settings, you don't have to worried about this

For more django related content

Subscribe To My Youtube Channel



Link in bio

DID YOU FIND THIS HELPFUL



Follow for more

#pythonworld