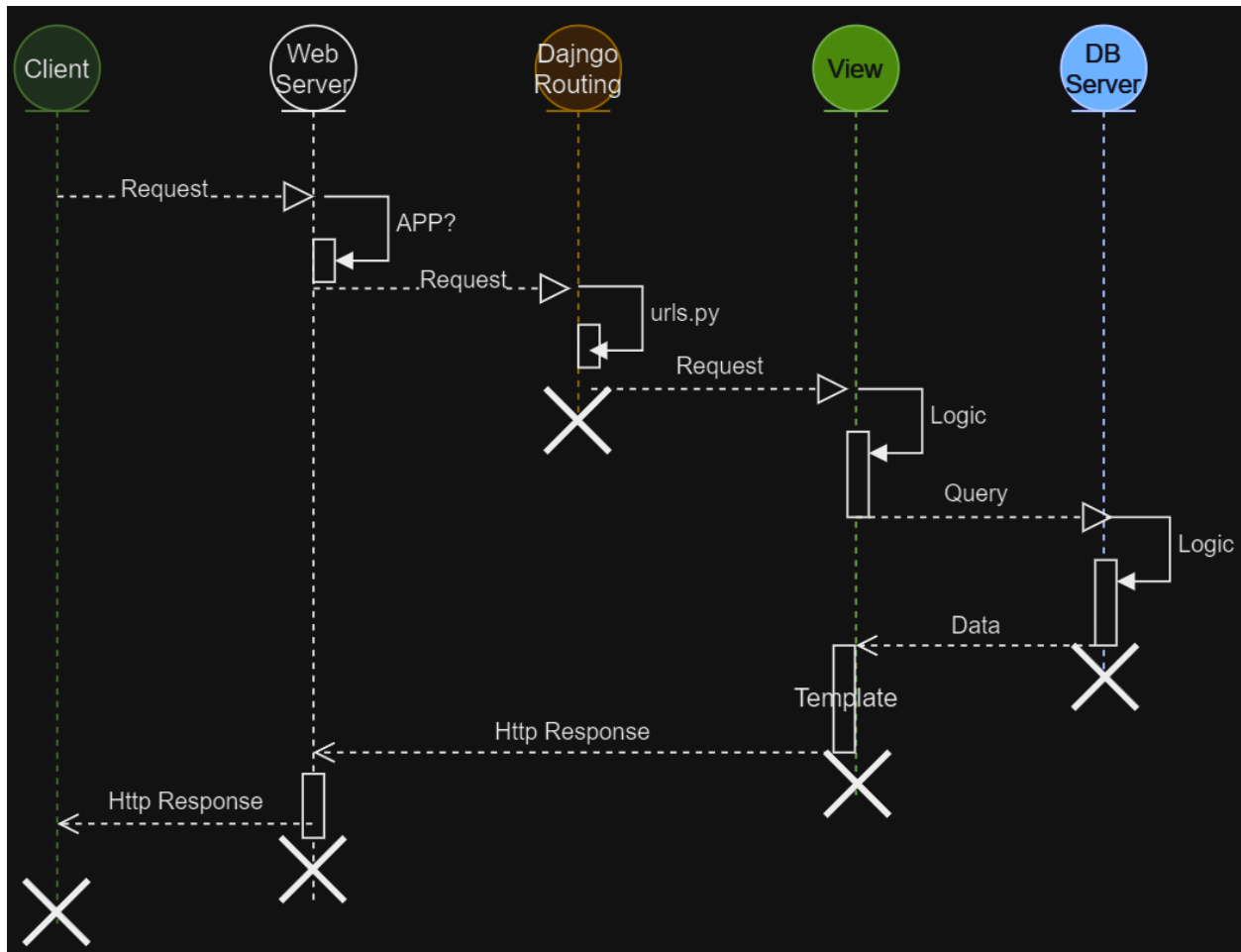


Contenido

<i>Repaso del flujo en Django</i>	<i>2</i>
<i>Corrección Links (request.path)</i>	<i>3</i>
<i>Realizar un cambio al modelo de la app projects.....</i>	<i>3</i>

Repaso del flujo en Django



1. El cliente realiza una solicitud HTTP (request)
 - a. Puede ser un navegador web, app móvil, script, llamada AJAX etc.
2. El servidor web, recibe la solicitud
 - a. El servidor web puede ser Apache, Nginx u otro junto con una interfaz de aplicación web como Gunicorn o uWSGI para servir aplicaciones Django
3. Ruteo de URL (Django routing)
 - a. El servidor web pasa la solicitud a la aplicación Django basándose en la configuración del servidor.
 - i. Puede ser mediante un proxy o mediante el uso de servidores de aplicaciones como Gunicorn o uWSGI
4. Middleware de Django
 - a. Antes de llegar a la vista, el request pasa a través del middleware de Django
 - i. Se realizan tareas tales como: autenticación, compresión, manipulación de encabezados, etc.
5. Django determina qué vista debe manejar la solicitud
 - a. Archivo urls.py

6. La vista realiza las operaciones correspondientes
 - a. Se puede conectar a la base de datos, mediante los modelos creados.
 - b. Realiza cálculos
7. Generación de respuesta HTTP
 - a. La vista crea una respuesta HTTP utilizando un objeto HttpResponse
8. Middleware de salida
 - a. La respuesta pasa nuevamente por el middleware de Django, por si se requiere realizar alguna tarea adicional antes de regresar la respuesta
9. Envío de la respuesta al servidor web
 - a. Django envía la respuesta al servidor web o al servidor de aplicaciones, dependiendo la configuración
10. Cliente recibe la respuesta
 - a. El cliente recibe la respuesta y la procesa
11. Cierre de la conexión
 - a. La conexión entre el cliente y el servidor se cierra después de que se haya completado la transmisión de datos

Corrección Links (request.path)

```
<li class="nav-item">
| <a class="nav-link {% if request.path == '/'%}active{% endif %}" href="{%
</li>
<li class="navbar-nav">
| <a class="nav-link {% if request.path == '/project/'%}active{% endif %}"
</li>
```

Realizar un cambio al modelo de la app projects

En el archivo models.py

Agregamos el atributo link, ya que está creada la base de datos, dicho campo debe permitir valores nulos y/o vacíos.

```
8 image = models.ImageField(verbose_name="Imagen", upload_to="projects")
9 link = models.URLField(null=True, blank=True, verbose_name="Liga")
```

Realizar las migraciones.

Ver los cambios en la carpeta "migrations"

Modificar el archivo project.html

```
{% if project.link %}  
<p>  
  <a href="{{project.link}}" target="_blank">  
    Más información</a>  
</p>  
{% endif %}
```