

# urls.py

Below is the documentation for `urls.py` file, explaining each route, its purpose, and how it works.

## 1. 3D Model Services [↗](#)

These routes handle operations related to 3D models, such as uploading, listing, and viewing models.

Route	Purpose	View/Service Function	HTTP Method	Example URL
<code>upload_model/</code>	Upload a 3D model.	<code>upload_model_service</code>	POST	<code>http://127.0.0.1:8000/upload_model/</code>
<code>model_view/</code>	List all available 3D models.	<code>model_list_service</code>	GET	<code>http://127.0.0.1:8000/model_view/</code>
<code>models/&lt;int:model_id&gt;/</code>	View details of a specific 3D model by its ID.	<code>view_model_service</code>	GET	<code>http://127.0.0.1:8000/models/1/</code>

## 2. Case Management [↗](#)

These routes handle operations related to cases, such as creating, fetching, and searching for cases.

Route	Purpose	View/Service Function	HTTP Method	Example URL
<code>cases/</code>	Render the UI for managing cases.	<code>cases</code>	GET	<code>http://127.0.0.1:8000/cases/</code>
<code>cases/api/</code>	Fetch all cases (API endpoint).	<code>fetch_cases_service</code>	GET	<code>http://127.0.0.1:8000/cases/api/</code>
<code>cases/api/create/</code>	Create a new case (API endpoint).	<code>create_case_service</code>	POST	<code>http://127.0.0.1:8000/cases/api/create/</code>
<code>cases/api/search/</code>	Search for cases (API endpoint).	<code>search_cases_service</code>	GET	<code>http://127.0.0.1:8000/cases/api/search/</code>

## 3. Calls [↗](#)

These routes handle operations related to calls.

Route	Purpose	View/Service Function	HTTP Method	Example URL
-------	---------	-----------------------	-------------	-------------

calls/	Render the UI for managing calls.	calls	GET	http://127.0.0.1:8000/calls/
calls/	Process calls (API endpoint).	process_call_service	POST	http://127.0.0.1:8000/calls/

#### 4. Accounting [↗](#)

These routes handle operations related to accounting and transactions.

Route	Purpose	View/Service Function	HTTP Method	Example URL
accounting/	Render the UI for managing accounting.	accounting	GET	http://127.0.0.1:8000/accounting/
accounting/api/	Process accounting transactions (API endpoint).	process_transaction_service	POST	http://127.0.0.1:8000/accounting/api/

#### 5. Sales [↗](#)

These routes handle operations related to sales.

Route	Purpose	View/Service Function	HTTP Method	Example URL
sales/	Render the UI for managing sales.	sales	GET	http://127.0.0.1:8000/sales/
sales/	Fetch sales summary (API endpoint).	get_sales_summary	GET	http://127.0.0.1:8000/sales/

#### 6. Lab Products [↗](#)

These routes handle operations related to lab products.

Route	Purpose	View/Service Function	HTTP Method	Example URL
lab_product/	Render the UI for managing lab products.	lab_product	GET	http://127.0.0.1:8000/lab_product/
lab_product/	Fetch lab products (API endpoint).	fetch_lab_products	GET	http://127.0.0.1:8000/lab_product/

## 7. Settings [↗](#)

These routes handle operations related to lab settings.

Route	Purpose	View/Service Function	HTTP Method	Example URL
<code>settings/</code>	Render the UI for managing lab settings.	<code>settings</code>	GET	<code>http://127.0.0.1:8000/settings/</code>
<code>settings/</code>	Fetch lab settings (API endpoint).	<code>get_lab_settings</code>	GET	<code>http://127.0.0.1:8000/settings/</code>

---

## 8. Accounts [↗](#)

These routes handle operations related to user accounts.

Route	Purpose	View/Service Function	HTTP Method	Example URL
<code>accounts/</code>	Render the UI for managing user accounts.	<code>accounts</code>	GET	<code>http://127.0.0.1:8000/accounts/</code>
<code>accounts/api/</code>	Fetch user accounts (API endpoint).	<code>get_accounts</code>	GET	<code>http://127.0.0.1:8000/accounts/api/</code>

---

## 9. Designer Dashboard [↗](#)

These routes handle operations related to the designer dashboard.

Route	Purpose	View/Service Function	HTTP Method	Example URL
<code>designer_dashboard/</code>	Render the UI for the designer dashboard.	<code>designer_dashboard</code>	GET	<code>http://127.0.0.1:8000/designer_dashboard/</code>
<code>designer_dashboard/</code>	Fetch designer dashboard data (API endpoint).	<code>get_designer_dashboa rd</code>	GET	<code>http://127.0.0.1:8000/designer_dashboard/</code>
<code>designer_dashboard/a pi/</code>	Fetch designer dashboard data (API endpoint).	<code>get_designer_dashboa rd</code>	GET	<code>http://127.0.0.1:8000/designer_dashboard/api/</code>

---

## 10. Outsourcing [↗](#)

These routes handle operations related to outsourcing.

Route	Purpose	View/Service Function	HTTP Method	Example URL
<code>outsource/</code>	Render the UI for managing outsourcing.	<code>outsource</code>	GET	<code>http://127.0.0.1:8000/outsource/</code>
<code>outsource/</code>	Send a case to outsource (API endpoint).	<code>send_to_outsource_service</code>	POST	<code>http://127.0.0.1:8000/outsource/</code>
<code>outsource/api/&lt;int:case_id&gt;/</code>	Send a specific case to outsource (API endpoint).	<code>send_to_outsource_service</code>	POST	<code>http://127.0.0.1:8000/outsource/api/1/</code>

## How to Use the Documentation

### 1. For Developers:

- Use the table to understand the purpose of each route and its corresponding view or service function.
- Test the routes using the provided example URLs.

### 2. For API Consumers:

- Use the API endpoints to interact with the backend (e.g., fetching data, creating records).
- Refer to the **HTTP Method** column to know which method to use (GET, POST, etc.).

### 3. For UI Developers:

- Use the UI routes to render the corresponding HTML templates.
- Ensure the view functions ( `cases` , `calls` , etc.) are implemented in `views.py` .