**Overview**

* We have defined our Models (aka database tables) and created some initial records to work with, it's time to write the code that presents that information to users.
* The first thing we need to do is determine what information we want to display in our website pages and define the URLs to use for returning those pages’ resources.
* Then we'll create a URL mapper, views, and templates to display the pages.
* The following diagram describes the main data flow, and the components required when handling **HTTP requests and responses**.
* As we already implemented the model, the main components we'll create are:
  + **URL mappers** to forward the supported URLs (and any information encoded in the URLs) to the appropriate view functions.
  + **View functions** to get the requested data from the models, create HTML pages that display the data, and return the pages to the user to view in the browser.
  + **Templates** to use when rendering data in the views.

Diagram

Description automatically generated

**URL dispatcher**

* If, at this point, you ran python manage.py runserver again, you’d still see the “Welcome to Django” message, with no trace of our “Hello world” view anywhere. That’s because our mysite project doesn’t yet know about the hello view; we need to tell Django explicitly that we’re activating this view at a particular URL. (Continuing our previous analogy of publishing static HTML files, at this point we’ve created the HTML file but haven’t uploaded it to a directory on the server yet.) To hook a view function to a particular URL with Django, use a URLconf.

A *URLconf* is like a table of contents for your Django-powered Web site. Basically, it’s a mapping between URLs and the view functions that should be called for those URLs. It’s how you tell Django, “For this URL, call this code, and for that URL, call that code.” For example, “When somebody visits the URL /foo/, call the view function foo\_view(), which lives in the Python module views.py.”

When you executed django-admin.py startproject in the previous chapter, the script created a URLconf for you automatically: the file urls.py. By default, it looks something like this:

**Overview**

To design URLs for an app, you create a **Python module** informally called a **URLconf** (URL configuration).

This module is pure Python code and is a mapping between URL path expressions to Python functions (your views).

This mapping can be as short or as long as needed. It can reference other mappings. And, because it’s pure Python code, it can be constructed dynamically.

Django also provides a way to translate URLs according to the active language. See the internationalization documentation for more information.