**Tutorial Part-1:**

**The Local Library website**

The module is the first part of our tutorial to explain what you'll learn, and provides an overview of the "local library" example website we'll be working through and evolving in subsequent modules.

|  |  |
| --- | --- |
| **Prerequisites:** | Read the [Django Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction). For the following modules you'll also need to have set up a Django development environment.  |
| **Objective:** | To introduce the example application used in this tutorial, and allow readers to understand what topics will be covered.  |

**Overview**

Welcome to the MDN "Local Library" Django tutorial, in which we develop a website that might be used to manage the catalog for a local library.

In this series of tutorial modules you will:

* Use Django's tools to create a skeleton website and application.
* Start and stop the development server.
* Create models to represent your application's data.
* Use the Django admin site to populate your site's data.
* Create views to retrieve specific data in response to different requests, and templates to render the data as HTML to be displayed in the browser.
* Create mappers to associate different URL patterns with specific views.
* Add user authorization and sessions to control site behaviour and access.
* Work with forms.
* Write test code for your app.
* Use Django's security effectively.
* Deploy your application to production.

You have learned about some of these topics already, and touched briefly on others. By the end of the tutorial series you should know enough to develop simple Django apps by yourself.

**The LocalLibrary website**

*LocalLibrary* is the name of the website that we'll create and evolve over the course of this series of tutorials. As you'd expect, the purpose of the website is to provide an online catalog for a small local library, where users can browse available books and manage their accounts.

This example has been carefully chosen because it can scale to show as much or as little detail as we need, and can be used to show off almost any Django feature. More importantly, it allows us to provide a *guided* path through the most important functionality in the Django web framework:

* In the first few tutorial modules we will define a simple *browse-only*library that library members can use to find out what books are available. This allows us to explore the operations that are common to almost every website: reading and displaying content from a database.
* As we progress, the library example naturally extends to demonstrate more advanced Django features. For example we can extend the library to allow users to reserve books, and use this to demonstrate how to use forms, and support user authentication.

Even though this is a very extensible example, it's called ***Local****Library* for a reason — we're hoping to show the minimum information that will help you get up and running with Django quickly. As a result, we'll store information about books, copies of books, authors and other key information. We won't however be storing information about other items a library might store, or provide the infrastructure needed to support multiple library sites or other "big library" features.