

Django Tutorial: The Local Library website

Created by 이윤준 (yoonyoon.lee@gmail.com)

May, 2019

We learn what you'll learn, and provide an overview of the "local library" example website we'll be working through and evolving in subsequent articles.

Overview

We try to develop a website that might be used to manage the catalog for a local library.

- 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.

The LocalLibrary website

LocalLibrary is the name of the website that we'll create and evolve over the course. 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.

It allows us to provide a *guided* path through the most important functionality in the Django web framework:

In the first few tutorial articles 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.

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.

I'm stuck, where can I get the source?

If you get stuck, you can find the fully developed version of the website on [Github here](#).