

Overview of Django and Angular

In this chapter we are going to discover the reasons why choosing the **Django** framework along with the **Python** programming language and the **Angular** framework along with the **Typescript** programming language make a perfect match for implementing a modern web application, which uses the common pattern that splits it in two different components a.k.a. the **frontend** and **backend**.

As mentioned, our web application is separated in two independent parts also known as the backend and frontend; this good practice offers several advantages for the development process:

- **Independent development**

With this approach it is possible to dedicate a specialized development team to develop the backend part and a different specialized team to the frontend so each team can focus on a different part.

Separate management of each development stays isolated from the other and less conflicts will appear when changes occur often, because usually the frontend and backend changes have no functional relationship.

It is possible to choose different technologies for the implementation of the backend and the frontend, using the one that suits best. As you already know, we have selected Django with Python as our candidates for the backend and Angular with Typescript for the frontend because each one shines at a different purpose.

- **Better solution**

As we are using different technologies, the combination of them results in a solution that is more powerful: more feature rich, more robust, and probably faster to develop than when using a single technology.

You may wonder why we use different technologies for the backend and frontend implementation. The reason is that each framework shines at a different purpose. And by combining them in the same project we get the best of both worlds.

The Django framework clearly shines for building robust backend applications with Python as the programming language. Python aims to provide clean and concise syntax while being simple, making it easier to read and write for developers.

The Angular framework is just used for building single-page applications (**SPAs**) or the frontend while using an object-oriented-programming approach **OOP** with Typescript as language. Typescript is based on JavaScript but enables writing code in a more productive way, making it easier to read and write for developers.

Angular also shines for its testing support compared to other JavaScript frameworks.

Both the Django and Angular frameworks follow the **batteries included** philosophy. This pragmatic trend basically means that they provide all the necessary tools for productive development out-of-the-box, and we do not need to install anything else to start coding. This way developers avoid spending precious time configuring additional components and can start working early on the application code. . Of course, this philosophy works best for standard projects that use the most commonly used functionality like ORM, REST API and security.

Both Django and Angular are mature frameworks, which are open source and have a huge community of users and support.

As we advance in the chapter describing the major features of these technologies, we will cover the following main topics:

- Understanding of the major advantages of Django framework
- Understanding of the major advantages of Angular framework

Understanding the major advantages of the Django framework

Let's begin by discovering some of the reasons why the Django framework is a perfect match for implementing a robust backend application using the **representational state transfer (REST) application programming interface (API)** pattern. The

REST API pattern is the most used for building distributed web applications and microservices nowadays, that's why we are going to focus on this approach in this book.

In the upcoming subsections, we have highlighted some of the key features that make Django shine.

Maturity and stability

Django has been in use for a long time, since 2003, and is tested by the framework's core development team and a large community of contributors.

This modern framework is known for its stability, which is why many developers choose it to build complex backend web applications.

Django benefits from the advantages of using Python as programming language, borrowing the simplicity and readability, great library and community support and easy integration with different technologies that Python offers.

Django owns a big community of users and developers where you can get active support in any case. The documentation is easy to understand and helps developers that are not familiar with Django get confident in a brief time with a low learning curve. The documentation is also an invaluable resource for solving doubts or troubles.

It is the developer's community that helps the framework stay stable.

The clean architecture and simple technology stack of Django make it especially easy for Python developers providing a small learning curve.

Clean architecture

The Django framework uses a variation of the popular **MVC architectural pattern (Model View Controller)**, the **MVT (Model View Template)** pattern..

Here is a brief explanation of the MVC/T pattern:

- **Model** or persistence layer
 - Manages the interaction with the database organized in a Domain model that matches the Business requirements. Communicates with the database with an object-relational mapping framework.

- Isolates the persistence layer from other layers of the application.
- **View** or user interface layer
 - Implements the presentation layer with a user interface or Api.
 - Handles the user interactions with the Domain model.
- In any case we are not going to use the View component in the form of a user interface in the application for this book because it is going to be implemented by the front-end application in Angular.
 For our application, the View will correspond to the REST Api. And Django can provide forms for interaction with the Api out-of-the-box. **Template/Controller** layer
 - Represents the design and look of the structure of how the data is shown to users (Template).
 - Represents a Layer for communication between the client or end user and the backend to handle requests and responses (Controller).

We are not going to use this part in the form of Template in our application because it is going to be implemented by the frontend. Only in the form of Controller will be used.

Full set of features and utilities

Various built-in packages in Django make it simple to develop web-based apps quickly; here are a few of them:

- A secure user authentication system
- An object relational mapping (ORM) system
- A built-in administration interface
- A web browsable API serialization

Rapid application development

The Django framework's pre-built features and tools help developers quickly create web applications with minimal code. This simplifies the start of a new project and allows quick prototyping of other features.

Scalability is critical when developing web applications. Ideally, they should handle substantial amounts of traffic and users without downtime. The Django framework supports load balancing, and developers can distribute incoming data across multiple servers.

The Django framework's reusable modules known as standalone "apps" hold a certain set of features, allowing for a smooth exchange of code across several project releases.

Django developers can create database-driven websites incorporating various technologies.

Integrating **Django** and **Angular** helps developers create dynamic UIs. Besides, its integration with HTML elements and CSS is key to creating visually appealing web pages. This makes the modular Django framework a double-edged sword!

Object-relational mapping

ORM is a well-known and common technique that automatically generates the source code for communication between the application code and the database based on a mapping of the Domain model.

The ORM implementation in Django provides a robust and user-friendly interface for interacting with databases. It helps the developer to interact with the database without the need to write the code in this layer, saving them all the effort caused by dealing with SQL queries. This simplifies scalable web application development without compromising performance.

Django's ORM offers compatibility with the majority of commonly used databases such as **PostgreSQL**, **MySQL**, **SQLite**, **Oracle and others**.

Django's ORM also allows developers to define their database models and relationships. This simplifies creating complex data models as developers do not need to deal with the underlying database schema. This results in much fewer lines of code to create and maintain that in turn are also easier to understand and enhances the overall application quality..

Django REST Framework

The REST API implementation allows developers to easily expose the functionality of the application through REST interaction.

The **Django REST framework (DRF)** is used to build a simple model-backed REST API. The framework has a good community for support in case of trouble, is well documented and is used by important companies in the world.

Admin tool

One cool feature is the **Django Admin** tool. Adding it to the project will make administrative work easier and simpler. This is a full web application provided out-of-the-box by Django for administration of the whole data model including security. We will talk about it in detail in **Chapter 2**.

Security

Django implements built-in protection against common web application vulnerabilities, and it is activated by default.

It promotes best practices for code concerning security that minimizes the chances for vulnerabilities and attacks. There is also a secure user authentication mechanism that simplifies the way developers use data and logins.

The security vulnerabilities the framework's built-in protections can deal with include:

- **Cross-site request forgery (CSRF)**: Prevention of unauthorized actions through forged requests.

This occurs when an attacker provides a fake interface that resembles the original one but performs malicious actions on it.

- **Cross-site scripting (XSS)**: Prevention of malicious script injection.

This occurs when an attacker exploits a vulnerability in a web application by executing a script from it in the browser with the purpose of stealing the session cookies.

- **SQL injection**: Prevention of an attacker interfering with the queries that an application makes to its database.

- **Clickjacking**: Prevention of an interface-based attack.

This occurs when a victim clicks on content on a visible website but executes some other actionable content in a hidden website.

Automated testing

Django preinstalled testing framework offers the possibility to create Unit and Integration Tests with little effort:

- With Django testing individual components in isolation (unit testing) is easy with the `django.test.TestCase` component.
- With Django testing the persistence layer and REST Api (integration testing) is easy with the `django.test.TestCase` and `rest_framework.test.APITestCase` components.
- Asynchronous code can also be tested in Django.

Understanding the major advantages of the Angular framework

In this section, you will discover some of the reasons why the Angular framework is a perfect match for implementing a robust frontend application using the SPA pattern. This is a commonly used pattern where the application uses a unique page that changes its content dynamically as response to the user interactions.

In the upcoming subsections, we have highlighted some of the key features that make Angular shine:

Maturity and stability

Angular was the first JavaScript based framework that appeared on the frontend development landscape; it was created by **Google** in 2010, so it's been in use for a long time and is tested by the framework's core development team as well as a large community of contributors.

This modern framework is known for its stability, which is why many developers choose it to build complex frontend web applications. It is the developer community that helps the framework stay stable.

Full set of features and utilities

The Angular framework's complete set of pre-built features allows a developer to implement the application and generate the new code with little coding effort:

- Instantly generate the initial code for web application.
- Create a web app with components having a loosely coupled interaction.
- Easily reuse components that are organized in a tree-like structure.
- Implement business logic in services that can be injected with fake implementations.
- Implement state management.
- Implement client-side navigation.
- Allow for clean separation between the UI and business logic.
- Design modern-looking UIs using Angular's own UI-component library.
- Modularize your app so modules are loaded on demand.

Rapid application development

The Angular framework's pre-built features and tools help developers quickly create web applications with minimal code. This simplifies the start of a new project and allows quick prototyping of other features. For this reason, replacement of developers is no big issue as long as the new members know Angular.

Scalability is critical when developing web applications. Angular encourages the best programming practices that will allow your application to grow in the future with no big issues. Angular will encourage and help to implement testable, and scalable user interfaces that perform well as they grow in the future.

Angular supports both TypeScript and JavaScript as a programming language. In mid- and large-size applications TypeScript increases the productivity of software developers

as it allows them to write code in an object-oriented way that avoids code duplication. TypeScript, being a statically typed language, has the advantage that errors are found at compile time instead of runtime, so your code is more reliable.

TypeScript has better support in IDEs for context-sensitive help and support for refactoring than Javascript.

Interfaces in Typescript offer a few advantages like avoiding compile time errors caused by objects that use a wrong type and allows declaring custom types .

Automated testing

The Angular framework's pre-built testing support is probably the best between JavaScript frameworks available today, helping developers quickly write code and automated tests together, following the good practice of test-driven development (**TDD**), which helps keep quality and productivity at its highest level throughout the whole life of the application.

Predictable releases

Angular's predictable release schedule and update tools help you keep your app up to date with the latest advancements in the framework and the web platform.

The Angular team predictably releases major updates twice a year. These regularly scheduled updates allow teams to be better informed and prepared for bug fixing and upgrading the application with new features.

Summary

In this chapter we discovered the main reasons why Django and Angular Frameworks together with Python and Typescript programming languages make a best match for writing a robust web application and why they are our best allies to keep the productivity at the highest level.

In the next chapter, we will start the practical side of this book by auto generating the code of a new Django Project with one command and implementing the persistence layer.

