The New and Improved Flask Mega-Tutorial

(2024 Edition)

Miguel Grinberg

The New and Improved Flask Mega-Tutorial (2024 Edition)

Miguel Grinberg

Jan 09, 2025

Contents

Preface

Back in 2012, I decided to start a software development blog. Because I am a do-it-yourselfer at heart, instead of using Blogger or WordPress, I sat down and wrote my own blog engine, using a then little known web framework called Flask. I knew I wanted to code it in Python, and I first tried Django, which was the most popular Python web framework at the time. But unfortunately Django seemed too big and too structured for my needs. I've found that Flask gave me as much power, while being small, unopinionated and unobtrusive.

Writing my own blog engine was an awesome experience that left me with a lot of ideas for topics I wanted to blog about. Instead of writing individual articles about all these topics, I decided to write a long, overarching tutorial that Python beginners can use to learn web development. And just like that, the Flask Mega-Tutorial was born!

The book that you have in your hands has been revised several times since 2012 to keep up with changes in the Python and Flask ecosystems. I released a first major revision, update and expansion of the original tutorial in 2018 thanks to the support of almost 600 Kickstarter backers. In 2021 I refreshed the content again after the release of Flask 2.0. This last revision, which I'm calling the "2024 Edition", includes support for Flask 3.0.

Who This Book Is For

This book will take you on a journey through a realistic web development project, from start to end. If you have just a little bit of experience coding in Python and understand how the web works at a high-level, you should have no trouble using this book to learn how to develop your own web applications using Python and Flask.

The tutorial assumes that you are familiar with the command line in your operating system. If you aren't, then I recommend that you learn how to execute programs, create directories, copy files, etc. using the command line before you begin.

If you have learned Flask with my original Mega-Tutorial, this new edition will introduce you to new features in Flask that did not exist when I wrote the original articles, as well as give you an updated look at important topics such as authentication, full-text search and internationalization. In addition

to the revised content, this version of the tutorial includes new chapters that cover topics that have become relevant in recent times, such as APIs, background jobs and containers.

Requirements

The example code that accompanies this book can be used on any platform on which Python runs, so macOS, Linux and Microsoft Windows are all valid choices. I have tested all the code extensively on Python versions as far back as 3.5 and continue to do so as new versions come out, so any recent version of Python should work fine.

If you are using a Microsoft Windows computer, you probably know that the world of web development is dominated by UNIX-based workflows, and you may rightly feel that you are at a disadvantage. That should not be a major concern when you work with this book, because when necessary, specific instructions that apply to Windows users are noted. My assumption is that if you are working on Windows you will be using the command prompt to work with your application. If you prefer to use PowerShell, you will need to translate commands to the appropriate syntax for that shell.

This may be hard to accept if you work on Windows, but I think you will have a better experience if you force yourself to learn UNIX, which can be done right on your Windows computer without making any drastic configuration changes. My recommendation is that you install UNIX tools on your Windows system and adopt the UNIX workflow. If you are interested in doing this, one option is the Windows Subsystem for Linux (WSL)[1], an officially supported feature of Windows 10 and 11 that adds a Linux system that runs in parallel with your Windows operating system and includes a UNIX version of Python. If your system is not compatible with WSL, then another very good option is Cygwin[2], an open-source POSIX emulation layer that includes Windows ports of a large number of UNIX tools, including Python. I have worked with Python under both WSL and Cygwin and find them perfectly adequate for web development work.

About The Example Application

The application that I'm going to develop as part of this tutorial is a nicely featured microblogging server that I decided to call *Microblog*. Pretty creative, I know.

Just so that you have some idea of what you will learn if you follow this tutorial, these are some of the topics that I will cover:

- User management, including secure password handling, logins, user profiles and avatars.
- Database management and database migration support
- Handling of user input via web forms

[1] https://msdn.microsoft.com/en-us/commandline/wsl/about
[2] https://cygwin.org

- Pagination of long lists of items
- Full-text search
- Email notifications to users
- HTML templates
- Working with dates and times
- Internationalization and localization
- Installation on a production server
- Working with Docker containers
- Application Programming Interfaces
- Push notifications
- Background jobs

I hope this application will serve as a template that you can use for writing your own web applications.

How To Work With The Example Code

I have released the complete source code for this project on the following GitHub repository: https://github.com/miguelgrinberg/microblog. There is a commit in this repository for each chapter.

The way I envision you will work through this tutorial is by writing the application on your own, based on the instructions provided in the text, at least for the first few chapters. You can certainly copy and paste portions of code from the text or from GitHub to save some typing, but I think it is important that you familiarize yourself with the task of coding a Flask application by writing the code yourself, instead of just downloading the files from GitHub (unless explicitly instructed to do so).

The GitHub repository can serve as a reference if you get lost and can't get the application to work. You can compare your files against the code in the repository link provided with each chapter if you get stuck with a problem you can't solve.

Conventions Used In This Book

This book frequently includes commands that you need to type in a terminal session. For these commands, a $ will be shown as a command prompt. This is a standard prompt for many Linux shells, but may look unfamiliar to Microsoft Windows users. For example:

```
$ python hello.py
hello
```

In a lot of the terminal examples, you are going to be required to have an activated *virtual environment* (do not worry if you don't know what this is yet, you will find out very soon!). For those examples, the prompt will appear as `(venv) $`:

```
(venv) $ python hello.py
hello
```

You will also need to interact with the Python interactive interpreter. Examples that show statements that need to be entered in a Python interpreter session will use a >>> prompt, as in the following example:

```
>>> print('hello!')
hello
```

In all cases, lines that are not prefixed with a $ or >>> prompt, are output printed by the command, and should not be typed.

Acknowledgements

This project would not have been possible without the amazing support of my Kickstarter backers. My deepest thanks go to Dhritiman Sagar, Alex Anderson, Bahrom Matyakubov, Dave Finnegan, John Gann, John W. O'Brien, Kojo Idrissa, Mark Anders, Raph, Fredrik Dahlgren, Jorge García García, Todd Twiggs, Pietro P Peterlongo, Chris Davis, Alexandre Harano, Bob Jordan, Chris Dent, Chris Jones, CptJason, Daniel Abeles, Daniel Plas Rivera, Dipanjan Sarkar, Eric Chou, Eric Ho, Graham Williamson, jiho Bak, John Sobanski, Kai Mies, Len Sumnler, Marc P. Rostock, Michael Sim, Nick Brandaleone, Nnamdi E. Anyanwu, R. Da Costa Faro, Reimund Klain, Scott Strattner, SNC Cloud Dev (twitter.com/snc_clouddev), T81, Tobias Siebenlist, Viet Le, Ed Wachtel, Shivas Jayaram, JVA, GenLots.com, Martin Thorsen Ranang, DFW Python, Allan Swanepoel, Andrej Stabenow, Anthony Bourguignon, Aron Filbert, Auke Bakker, Bryson Tyrrell, Chuck Woodraska, Colin R. Crossman, Dario Varotto, Dax Morrow, Eric G. Barron, Everett Toews, Fisherworks, flasky mcflaskface, Iain Hunter, Jeremy Barisch Rooney, Jesse Liles, Jindrich K. Smitka, Jing Sheng Pang, Karthik Ramakrishnan, Kevin Porterfield (KP), Leonel Decunta, Martynas Budvytis, Mathew Divine, Matt Makai (Full Stack Python), Matt Trentini, Michael from Talk Python, Nana B Okyere, Nathan Sanders, Nduka Obinna Azubuike, Neal Duncan, Philip Penquitt, Rémi Debette, Romer Ibo, Ryan Hagan,

Scott Andrew Underwood, Stephan Simon, Steve Bartell, Timothy DAuria, Vitaly Popovich, Yi Luo and the remaining 484 backers.