Building Web Applications with Flask

Flask is a powerful web framework that helps you build great projects using your favorite tools. Flask takes the flexible Python programming language and provides a simple template for web development. Once imported into Python, Flask can be used to save time building web applications. It goes against the flow with the microframework concept, leaving most of the architecture choices to the developer. Through its great API, extensions, and powerful patterns, Flask helps you create simple projects in minutes and complex ones as soon as possible.

From the beginning, Building Web Applications with Flask shows you how to utilize Flask's concepts, extensions, and components to create engaging, full-featured web projects. You'll learn how to properly handle forms using WTForms, design convenient templates with Jinja2 tags and macros, use NoSQL and SQL databases to store user data, test your projects with features and unit tests, create powerful authentication and user authorization, as well as administrative interfaces with ease, and more.

As Flask does not enforce an architectural recipe, neither do we! This book makes no coding assumptions on how you should code, leaving you free to experiment.

Who this book is written for

If you are a Python web developer who wants to learn more about developing applications in Flask and scaling them with industry-standard practices, this is the book for you.

What you will learn from this book

- Create single page applications with Flask
- Use macros, filters, tags, and control structures to render rich responses to user requests
- Receive forms securely through form-based classes and WTForms
- Utilize NoSQL or SQL databases to store user data seamlessly
- Generate simple, yet powerful, REST services from your data models
- Learn how to perform test-driven and behavior-driven developments in your Flask projects
- Use Flask components to create maintainable non-flat projects
- Harvest the power of extensions to build robust authorization and permission policies


Italo Maia

$ 34.99 US
£ 22.99 UK

Prices do not include local sales tax or VAT where applicable

Building Web Applications with Flask

Use Python and Flask to build amazing web applications, just the way you want them!
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Flask in a Flask, I Mean, Book'
- A synopsis of the book’s content
- More information on Building Web Applications with Flask
About the Author

Italo Maia is a full-stack developer with 10 years of experience in creating software for the mobile, Web, and desktop environments, having dedicated most of the last few years to development with Python and web technologies.

Author of Flask-Empty, a popular skeleton for Flask projects that aggregates good practices and recipes for quick prototyping, he is active in the Brazilian Python communities, having open source tools and libraries available in GitHub and Bitbucket.
One can hardly develop new applications in our "now world" without duct-taping a lot of technologies together, be it new trend databases, messaging systems, or languages of all kinds. When talking about web development, things might get slightly more complicated as not only do you have to mix a lot of technologies together, but they must also work well with the applications accessing them (also known as web browsers). They should also be compatible with your deployment server, which is another story in itself!

In the Python world, where people deliver amazing desktop software following great guidelines such as the Zen of Python and PEP8, we have, at our disposal, a wide range of libraries and frameworks for creating great web applications, each with its own philosophy. Django, for example, is a bundle solution; it makes choices for you on how your project should look, what should it have, and how things should be done. Web2py is another framework solution that goes beyond and bundles even the IDE with it. These are great concepts, but if you want to create something simple, I would suggest you to do it somewhere else. They're usually good choices, but sometimes they're just too much (the latest Django version seems decisive in changing that; let's keep an eye on further developments).

Flask positions itself, not as a full-power out-of-the-box solution like both Django and Web2py, but as a minimalistic solution where you're given the bare minimum to work with and choose all the other stuff. That's very helpful when you want granular control of your application, when you want to precisely pick your components, or when your solution is simple (not simplistic, okay?).

Preface
Preface

This book is a response to that scenario of beautiful code and many options in the Web world. It tries to walk through the main concerns regarding web development, from security to content delivery and from session management to REST services and CRUD. Important modern concepts such as overengineering, quality and the development process are covered, so as to achieve better results from day one. To make the learning process smooth, subjects are presented without rush and followed by commented examples. The book also sets out to give readers real-world advice on how to prevent common problems with code.

Come learn how to create great Flask applications, delivering value to your projects and customers!

What this book covers

Chapter 1, Flask in a Flask, I Mean, Book, introduces you to Flask, explaining what it is, what it is not, and how it positions itself in the web framework world.

Chapter 2, First App, How Hard Could it Be?, covers the very first step toward Flask development, including environment setup, your very own "Hello World" app, and how templates enter into this equation. A fluffy chapter it is!

Chapter 3, Man, Do I Like Templates!, deals with face tags and filters progresses through the Jinja2 template engine and how it integrates with Flask. Things start to get a little serious from here!

Chapter 4, Please Fill in This Form, Madam, discusses how to handle forms (as forms are a fact in the web development life) with all the care they need using WTForms in all its glory!

Chapter 5, Where Do You Store Your Stuff?, introduces you to the concepts of relational and non-relational databases, covering how to handle both cases, and also when to.

Chapter 6, But I Wanna REST Mom, Now!, is a chapter on creating REST services (as the REST hype must be satisfied), manually and using the amazing Flask-Restless.

Chapter 7, If Ain't Tested, It Ain't Game, Bro!, is our quality-centric chapter where you learn to deliver quality through proper testing, the TDD and BDD way!

Chapter 8, Tips and Tricks or Flask Wizardry 101, is a dense chapter covering good practices, architecture, blueprints, debugging, and session management.
Chapter 9, *Extensions, How I Love Thee*, covers all those great Flask extensions not covered so far that will help you achieve the productivity the real world requires from you.

Chapter 10, *What Now?*, ends our development trip covering all the basics for a healthy deployment, and points you toward your next steps in the Flask world.
Flask in a Flask, I Mean, Book

What is Flask? It’s a question that humanity has been pondering for millennia... well, actually, since 2010, when Armin Ronacher first committed to the project. Flask is a Web framework and is quite different from what most people are used to working with. It is less presumptuous about how your application should look or what you should use to make it available. The BSD licensed package has all this!

An introduction to Flask and its features

The Flask framework is actually a glue, a very nice one, that sticks together the amazing Werkzeug and Jinja2 frameworks, responsible for answering requests and presenting the output (HTML, maybe). In the MVC architecture, also known as Model-View-Controller, Flask covers C and V. But where is M? Flask does not provide you with an integrated model layer out-of-the-box as that is not actually needed for a web application. If you do need to work with a database, just pick your database solution from the many available and create your own model layer, which is not hard, and be happy! The concept of a micro-framework, with good intentions and made just for Flask, is all about giving you the the smallest (but also the most useful) feature set you need, and one that won't get in the way.

Just what are the features that must be in the framework?

• A development server and debugger (sanity-friendly)
• Unicode support (Latin language-friendly)
• WSGI compliance (uWsgi-friendly)
• A unit-test client client (code with quality)
• URL routing (it brings tears to my eyes, it's so beautiful!)
• Request dispatching
• Secure cookies
• Sessions
• Jinja2 templates (tags, filters, macros, and more)

With that much, you can handle Ajax requests, browser requests, and user sessions between requests; route HTTP requests to your controllers; evaluate form data; respond to HTML and JSON; and so on.

That is nice, but is Flask not an MVC framework? Well, that's arguable. If a web framework does not implement an MVC antipattern, such as handling requests in the view or mixing models and controllers, it could potentially facilitate an MVC, which, in my opinion, is as good as it gets because it does not enforce your application structure.

Flask is not an MVC framework as it does not implement the model layer, although it does not restrict you in any way if you wish to create your own.

If you need a simple, single-file web application that receives a form and gives back an answer, HTML or not, Flask will help you with that, easily. If you need a multilayer, high-depth modularized Facebook clone, Flask can also be there for you.

So, what did we learn so far?

• Flask was born in 2010
• Flask is a minimalistic web framework based on Jinja2 and Werkzeug
• Flask does not enforce a specific project architecture

Refer to Flask license details at http://flask.pocoo.org/docs/0.10/license/.

Right now, you might be wondering which of your neat project ideas could be done with Flask. That's the spirit! What about thinking together on this one?

Flask does not come with bundled functionality in terms of database integration, a forms library, administration interface, or migration tools. You can have these through extensions, which will be discussed soon enough, but they are all external to Flask. If you need these extensions right at the beginning of your project and you don't want to set it up (or can't spare the time to), you might do better with a full-fledged MVC all-in one, low-cohesive, and high-coupling framework such as Django.
Now, imagine you need to build a website with a single form, such as a http://cashcash.cc/ clone, which receives a form and returns the current currency trade values; Flask could help conclude your project really fast.

Let's think further. What if you need a specific set of libraries to work together in your project and you don't want the web framework getting in the way; that's another very good scenario for Flask as it gives you the bare minimum and lets you put together everything else you may need. Some frameworks have such a high level of coupling (read dependency) on their own components that you may have a serious problem if you want to use a specific alternative.

For example, you may want to use a NoSQL database in your project; however, if you do so, some components of your project may stop working (for example: an administrative component).

Basically, if you have the time to spare, if you're doing something simple, if you want to implement your own architecture solution, or if you need granular control of the components used in your project, Flask is the web framework for you.

**Summary**

Now, let's talk about awesomeness, your awesomeness after reading this book—you will be capable of handling HTTP and Ajax requests; creating fully featured web applications with database integration (SQL and NoSQL) and REST services; using Flask extensions (forms, caching, logging, debugging, auth, permissions, and so on); and modularizing and unit- and feature-testing your applications.

I hope you enjoy this book and build great things with what you learn here.
Where to buy this book

You can buy Building Web Applications with Flask from the Packt Publishing website. Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.