# Foreword for Experienced Programmers

## Thread-Locals in Flask

One of the design decisions in Flask was that simple tasks should be simple; they should not take a lot of code and yet they should not limit you. Because of that, Flask has a few design choices that some people might find surprising or unorthodox. For example, Flask uses thread-local objects internally so that you don't have to pass objects around from function to function within a request in order to stay threadsafe. This approach is convenient, but requires a valid request context for dependency injection or when attempting to reuse code which uses a value pegged to the request. The Flask project is honest about thread-locals, does not hide them, and calls out in the code and documentation where they are used.

## Develop for the Web with Caution

Always keep security in mind when building web applications.

If you write a web application, you are probably allowing users to register and leave their data on your server. The users are entrusting you with data. And even if you are the only user that might leave data in your application, you still want that data to be stored securely.

Unfortunately, there are many ways the security of a web application can be compromised. Flask protects you against one of the most common security problems of modern web applications: cross-site scripting (XSS). Unless you deliberately mark insecure HTML as secure, Flask and the underlying Jinja2 template engine have you covered. But there are many more ways to cause security problems.

The documentation will warn you about aspects of web development that require attention to security. Some of these security concerns are far more complex than one might think, and we all sometimes underestimate the likelihood that a vulnerability will be exploited - until a clever attacker figures out a way to exploit our applications. And don't think that your application is not important enough to attract an attacker. Depending on the kind of attack, chances are that automated bots are probing for ways to fill your database with spam, links to malicious software, and the like.

Flask is no different from any other framework in that you the developer must build with caution, watching for exploits when building to your requirements.