

Flask命令行接口

目录

- [Flask命令行接口](#)
 - [运行Shell](#)
 - [自定义命令](#)
 - [应用上下文](#)

Flask命令行接口

在之前的章节[Flask框架简介](#)中我们已经使用过Flask的命令行接口。

运行Shell

使用下面的shell命令来开启一个交互式的Python shell：

```
(flask_env3) $ flask shell
```

这将开启一个交互式的Python shell，并且在其中设置好了正确的应用上下文和本地变量（local variables）。这是通过调用应用的 `Flask.make_shell_context()` 方法做到的。默认地你将可访问到 `app` 和 `g`。

自定义命令

Flask 使用[click](#)来实现命令行接口，这使得添加自定义命令非常容易。例如，如果你想要一个shell 命令来初始化数据库，你可以这样做：

```
import click
from flask import Flask

app = Flask(__name__)

@app.cli.command()
def initdb():
    """Initialize the database."""
    click.echo('Init the db')
```

在命令行中执行这个命令：

```
(flask_env3) $ flask initdb
Init the db
```

应用上下文

大多数命令需要对应用对象执行操作，所以为它们设置好应用上下文非常有必要。正因为如此，如果你在 `app.cli` 上通过 `command()` 注册了一个回调，这个回调将自动被 `cli.with_appcontext()` 包装（wrapped）来通知命令行系统确保设置好一个应用上下文。如果一个命令是通过 `add_command` 或其他方法稍后添加的，那这种行为将不可用。

这种行为也可以通过向装饰器传递 `with_appcontext=False` 来禁用：

```
@app.cli.command(with_appcontext=False)
def example():
```

pass

日期: 2017-07-14

作者: 田宇伟