**蓝图的基本使用**

**蓝图：**

之前我们写的url和视图函数都是处在同一个文件，如果项目比较大的话，这显然不是一个合理的结构，而蓝图可以优雅的帮我们实现这种需求。

**1.** 蓝图的作用就是让我们的Flask项目更加模块化，结构更加清晰,为了更好的管理项目 让项目达到分层解耦 而产生的。可以将相同模块的视图函数放在同一个蓝图下，同一个文件中，方便管理。

**2.** 基本语法：

\* 在蓝图文件中导入Blueprint：

**from** flask **import** Blueprint

user\_bp = Blueprint(**'user'**,\_\_name\_\_)

\* 在主app文件中注册蓝图：

**from** flask **import** Flask

**from** blueprints.user **import** user\_bp

app = Flask(\_\_name\_\_)

app.register\_blueprint(user\_bp)

**3.** 如果想要某个蓝图下的所有url都有一个url前缀，那么可以在定义蓝图的时候，指定url\_prefix参数：

user\_bp = Blueprint(**'user'**,\_\_name\_\_,url\_prefix=**'/user'**)

*#个人中心的 url与视图函数*

@user\_bp.route(**'/profile/'**)

**def** profile():

**return '个人中心页面'**

*#个人设置中心的 url与视图函数*

@user\_bp.route(**'/settings/'**)

**def** settings():

**return '个人设置页面'**

**注意：** 在定义url\_prefix的时候，要注意后面的斜杠，如果给了，那么以后在定义url与视图函数的时候，就不要再在url前面加斜杠了。

**访问结果如：**

