**4.5多环境配置·Flask的配置参数详解**

Flask自带有一些配置参数，也就是我们的app.config，这里整个表格对配置参数进行描述说明。

|  |  |  |
| --- | --- | --- |
| 参数名 | 默认值 | 描述说明 |
| DEBUG | False | 是否开启Debug模式 |
| TESTING | False | 是否开启测试模式 |
| PROPAGATE\_EXCEPTIONS | None | 异常传播(是否在控制台打印LOG) 当Debug或者testing开启后,自动为True |
| PRESERVE\_CONTEXT\_ON\_EXCEPTION | None | 一两句话说不清楚,一般不用它 |
| PERMANENT\_SESSION\_LIFETIME | 31 | days , Session的生命周期(天)默认31天 |
| USE\_X\_SENDFILE | False | 是否弃用 x\_sendfile |
| LOGGER\_NAME | None | 日志记录器的名称 |
| LOGGER\_HANDLER\_POLICY | always | 日志处理策略 |
| SERVER\_NAME | None | 服务访问域名 |
| APPLICATION\_ROOT | None | 项目的完整路径 |
| SESSION\_COOKIE\_NAME | session | 在cookies中存放session加密字符串的名字 |
| SESSION\_COOKIE\_DOMAIN | None | 在哪个域名下会产生session记录在cookies中 |
| SESSION\_COOKIE\_PATH | None | cookies的路径 |
| SESSION\_COOKIE\_HTTPONLY | True | 控制 cookie 是否应被设置 httponly 的标志， |
| SESSION\_COOKIE\_SECURE | False | 控制 cookie 是否应被设置安全标志 |
| SESSION\_REFRESH\_EACH\_REQUEST | True | 这个标志控制永久会话如何刷新 |
| MAX\_CONTENT\_LENGTH | None | 如果设置为字节数， Flask 会拒绝内容长度大于此值的请求进入，并返回一个 413 状态码 |
| SEND\_FILE\_MAX\_AGE\_DEFAULT | 12 | hours 默认缓存控制的最大期限 |
| TRAP\_BAD\_REQUEST\_ERRORS | False | #如果这个值被设置为 True ，Flask不会执行 HTTP 异常的错误处理，而是像对待其它异常一样， #通过异常栈让它冒泡地抛出。这对于需要找出 HTTP 异常源头的可怕调试情形是有用的。 |
| TRAP\_HTTP\_EXCEPTIONS | False | # Werkzeug 处理请求中的特定数据的内部数据结构会抛出同样也是“错误的请求”异常的特殊的 key errors 。 # 同样地，为了保持一致，许多操作可以显式地抛出 BadRequest 异常。 # 因为在调试中，你希望准确地找出异常的原因，这个设置用于在这些情形下调试。 # 如果这个值被设置为 True ，你只会得到常规的回溯。 |
| EXPLAIN\_TEMPLATE\_LOADING | False |  |
| PREFERRED\_URL\_SCHEME | http | 生成URL的时候如果没有可用的 URL 模式话将使用这个值 |
| JSON\_AS\_ASCII | True | #默认情况下 Flask 使用 ascii 编码来序列化对象。如果这个值被设置为 False ， #Flask不会将其编码为 ASCII，并且按原样输出，返回它的 unicode 字符串。#比如 jsonfiy 会自动地采用 utf-8 来编码它然后才进行传输。 |
| JSON\_SORT\_KEYS | True | #默认情况下 Flask 按照 JSON 对象的键的顺序来序来序列化它。 #这样做是为了确保键的顺序不会受到字典的哈希种子的影响，从而返回的值每次都是一致的，不会造成无用的额外 HTTP 缓存。 #你可以通过修改这个配置的值来覆盖默认的操作。但这是不被推荐的做法因为这个默认的行为可能会给你在性能的代价上带来改善。 |
| JSONIFY\_PRETTYPRINT\_REGULAR | True |  |
| JSONIFY\_MIMETYPE | 'application/json' |  |
| TEMPLATES\_AUTO\_RELOAD | None |  |

注：以上配置都可以在config.py配置文件中的类中进行变量定义，其配置会覆盖flask的默认配置。