

在实际的工作中，绝大部分公司都有至少 3 个以上的环境，供测试与研发人员使用。测试人员不可能为每个环境都准备一个自动化测试脚步，因为这样的维护成本太过庞大。所以，要解决这个问题，就需要做到设计一套脚本，可以在各个环境上面运行。

## 实战演示

在之前的文章[《基于加密接口的测试用例设计》](#)中，提到过的字典结构体的基础上进行改造，将 URL 从一个写死的 IP 地址改为一个域名。

```
1 req_data={
2     "schema": "http",
3     "method": "get",
4     "url": "
http://docker.testing-studio.com:10000/demo64.txt
5     "headers":
6     {"Host":None}
7 }
8
```

然后，我们还需要另外一个字典结构体去存储环境的配置。

这里使用 `env` 变量存放一个字典结构体的环境配置。然后将请求结构体中的 URL 替换为 `env` 配置文件中个人选择的 URL。先用 `env["docker.testing-studio.com"]` 读到全部的环境配置选项，再使用改变 `env["default"]` 中的选择，去切换不同的环境。

```
1 #把host修改为ip，并附加host header
2 env={
3     "docker.testing-studio.com": {
4         "dev": "127.0.0.1",
5         "test": "1.1.1.2"
6     },
7     "default": "dev"
8 }
9 data["url"]=str(data["url"]).replace(
10     "docker.testing-studio.com",
11     env["docker.testing-studio.com"][env["default"]]
12 )
13 data["headers"]["Host"]="docker.testing-studio.com"
14
```

由上面两个步骤就可以将环境的切换作为一个可配置的选项，根据需求，很方便的去切换不同的使用环境。不过目前这样的解决办法依然是写在代码中，不够优雅，