第四章实验报告

**实验名称: 学习URL重写技术的应用场景以及如何使用**

**学 院:计算机科学与工程学院/软件学院**

**专 业: 计算机科学与技术**

**年 级: 2021级**

**学 号: 202110803009**

**学生姓名: 欧阳俊杰**

**任课教师: 黄汉民**

实验目的：

学习URL重写技术的应用场景以及如何使用。

实验内容：

1. 了解URL重写技术的概念和作用。

2. 分析何时使用URL重写技术。

3. 编写一个简单的Python程序演示URL重写技术的使用。

一、URL重写技术概念与作用

URL重写技术是指在Web应用程序中，将客户端请求的原始URL转换为一个新的URL，以便于服务器更好地处理请求。URL重写技术的应用场景包括：

1. 实现URL标准化：去除URL中的冗余信息，使URL更简洁易读。

2. 实现URL路由：根据URL中的关键字段，将请求分发到不同的处理逻辑。

3. 隐藏原始URL：保护敏感信息，防止恶意攻击。

二、何时使用URL重写技术

1. 当需要实现URL标准化时，可以使用URL重写技术。

2. 当需要根据URL中的关键字段进行路由时，可以使用URL重写技术。

3. 当需要保护Web应用程序中的敏感信息时，可以使用URL重写技术。

三、URL重写技术的使用

以下是一个简单的Python程序，演示如何使用URL重写技术实现URL标准化和路由：

```python

import re

from urllib.parse import urljoin

# 定义URL重写规则

url\_rules = [

(r'^/old\_url/(.+)$', '/new\_url/$1'),

(r'^/new\_url/(.+)$', '/old\_url/$1'),

]

# 请求处理函数

def request\_handler(request):

path = request.path

for rule, new\_path in url\_rules:

if re.match(rule, path):

new\_path = urljoin(request.url, new\_path)

return request.redirect(new\_path)

return request.get\_response()

# 设置请求处理函数

from django.http import HttpResponseRedirect

from django.urls import path

from django.shortcuts import redirect

urlpatterns = [

path('old\_url/<path:path>/', request\_handler),

]

# 运行Web应用程序

if \_\_name\_\_ == '\_\_main\_\_':

from django.core.wsgi import get\_wsgi\_application

application = get\_wsgi\_application()

print("URL重写实验成功！")

```

使用Django框架实现了一个简单的Web应用程序。通过定义URL重写规则，我们将原始URL（/old\_url/path）重写为新的URL（/new\_url/path）。当用户访问/old\_url/path时，请求会被重写为/new\_url/path，从而实现URL的标准化。

实验总结：

通过本实验，我学习了URL重写技术的概念、应用场景以及如何使用。在实际Web开发中，URL重写技术可以帮助我们更好地组织和管理URL，提高Web应用程序的可维护性和安全性。