

< 서버 환경에서 DEBUG 모드 끄기 >

- 장고의 환경 설정 파일에는 DEBUG를 운영 환경에서는 반드시 False로 해야 한다고 되어 있다.
파라미터를 DEBUG=True 상태로 운영하면 오류 발생시 서버정도가 높을되기 때문이다.

[실습] DEBUG 설정하기

1) DEBUG 항목을 False로 설정하기. (Project/mysite/config/settings/prod.py)

DEBUG = False

↳ 그리고 이를 이용하여 변경 내역을 서버에 적용한다.

2) DEBUG 항목의 차이점 확인하기.

DEBUG = True

- 404 오류에서 url.py 파일에 설정한 내용이 노출된다

DEBUG = False

- Not Found 만 보인다

[실습] 오류 페이지를 서비스하게 만들기.

- 서버 환경의 404 오류 페이지는 멋있어서 서비스업지 못하다.
404, 500 오류 페이지를 '서비스하게' 표시하자.

1) handler 404 변수 설정하기 (config/url.py)

handler404 = 'common.views.page_not_found'

↳ handler 404 변수를 설정하면 404 오류 발생시 사용자가 정의한 뷰 함수를 호출한다.

2) Page_not_found 함수 추가하기. (Common/Views.py)

```
def Page_not_found(request, exception):
```

```
    # You page not found
```

```
    return render(request, 'Common/404.html', {})
```

3) 404 템플릿 작성하기 (templates/Common/404.html)

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="container">
```

```
<div class="row justify-content-center">
```

```
<div class="col-12 text-center">
```

```
<span class="display-1 d-block">404</span>
```

```
<div class="mb-4 lead">페이지를 찾을 수 없습니다.</div>
```

```
<a href="/" class="btn btn-link">홈으로 돌아가기</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
{% endblock %}
```

↳ 글자 이용해서 서버에 적용하기..Gunicorn 재설정

<이슈 노트>

1. Gunicorn 과 Nginx 설정, 실행을 완료하고 IP로 접속하니, 기능들이 작동하지 않고 500 Error 발생, 부트스트랩도 불러오지 못하는 것으로 보임.

↳ 예상 원인

DEBUG = True 로 바꾸어 원인 자세히 찾아보기.

↳ Google 로그인 설정이 admin에서 설정되지 않아서 오류 발생

↳ 해결 방법

↳ Admin에서 구글 로그인 설정 완료

2. 정적 파일을 nginx가 불러오지 못해서 부트스트랩이 적용되지 않음.

↳ NGINX 프로세스는 일반적으로 www-data 사용자로 실행 되어서 권한을 1115로 바꿔야함.

3. 명령어 정리

- systemctl daemon-reload

↳ /etc/systemd/system/ 의 서비스 파일을 갱신한 뒤, systemd 데몬에게 변경된 서비스 정보를 다시 읽어들이도록 재시작.

- systemctl restart msite.service

↳ msite.service 완전히 중단했다가 다시 가동함, 변경된 서비스 설정이 적용되어 재실행됨

- systemctl reload msite.service

↳ 애플리케이션을 완전 재시작하지 않고 설정 파일만 다시 읽도록 요청.

Gunicorn은 reload 신호를 처리하지 않거나 restart 사용

- systemctl status msite.service

↳ 서비스 실행 상태, 마지막 실행, 오류 코드 등을 확인.

- nginx t

↳ NGINX 설정 파일 문법을 테스트함.

- systemctl reload nginx

↳ 구동중인 nginx 프로세스들에게 재설정 신호를 보내, 설정 파일을 다시 읽어들이게 한다.

- systemctl restart nginx

↳ nginx 재시작, 모든 워커 프로세스가 새 설정으로 재실행 된다.

4. 수정 내용이 서버 환경에서 반영되는 과정

1) mysite.service 수정 : 서비스 파일 내 경로, 환경 변수 등을 변경

2) sudo systemctl daemon-reload : systemd에게 변경된 서비스 정보를 알림.

3) sudo systemctl restart mysite.service : systemd을 새로운 설정으로 재실행

4) Django 코드, 설정 파일 수정

5) nginx 설정 수정

↳ nginx -t 로 테스트

6) sudo systemctl reload nginx