

INI Intern First Project - Wordpress & PHP

장예훈 (Video Delete)

18.07.02 Mon - 18.07.17 Tue

프로젝트 개요

목적

LAMP, Wordpress 를 이용하여 사용자 별 자신이 업로드한 비디오 파일을 관리할 수 있는 플랫폼 구축
비디오 업로드, 비디오 리스트, 비디오 삭제 기능을 제공

개발 서버 구축

- 팀원들이 맡은 각 기능을 구현 후 통합하고, 프로젝트의 요구사항 전체를 이행하기 위한 환경이 요구됨
- 각 기능들의 구현 환경과 개발 서버 환경의 버전 이슈를 방지하기 위해 개발 환경에 대한 컨벤션 설계

개발 서버 환경

- Ubuntu 16.04.5 LTS
- Wordpress
- Apache 2
- MySQL 5.7.23
- PHP 7.1
- API Star 0.5.40

Issue of 개발 서버 구축

우분투 버전 이슈

1. Issue 정의

- 처음 개발 서버에 AMP(Apache-MySQL-PHP) 설치 시 configuration 설정이 적용되지 않거나 명령어가 실행하지 않는 이슈 발생
- 여러 번의 에러 해결 시도 후 우분투가 18.04 버전인 것을 확인
- 우분투 18.04 버전에서 지원되지 않는 라이브러리 또는 패키지 내 모듈이 다소 존재

- 우분투에서는 버전 downgrade를 지원하지 않음
2. Issue 해결 방안
- 프로젝트 참여자들의 우분투 버전과 동일한 우분투 server 16.04.5 LTS 설치

비디오 삭제 및 비디오 파일 관리 프로세스

1. 개발 환경

- Ubuntu 16.04.5 LTS
- Wordpress
- Apache 2
- MySQL 5.7.23
- PHP 7.1
- Anaconda 4.3.11
- API Statr 0.5.40

2. Wordpress 설치 및 세팅

2.1. LAMP 설치

Issue of 2.

Wordpress를 지원하는 AMP 버전이 정해져 있음

1. Issue 정의
 - 초반에 AMP를 모두 설치하고 Wordpress를 설치하려고 할 때 에러가 나거나 configuration이 적용되지 않음
 - Wordpress를 지원 AMP 버전이 정해져 있음
2. Issue 해결 방안
 - Wordpress를 지원하는 AMP 버전 확인 후 설치

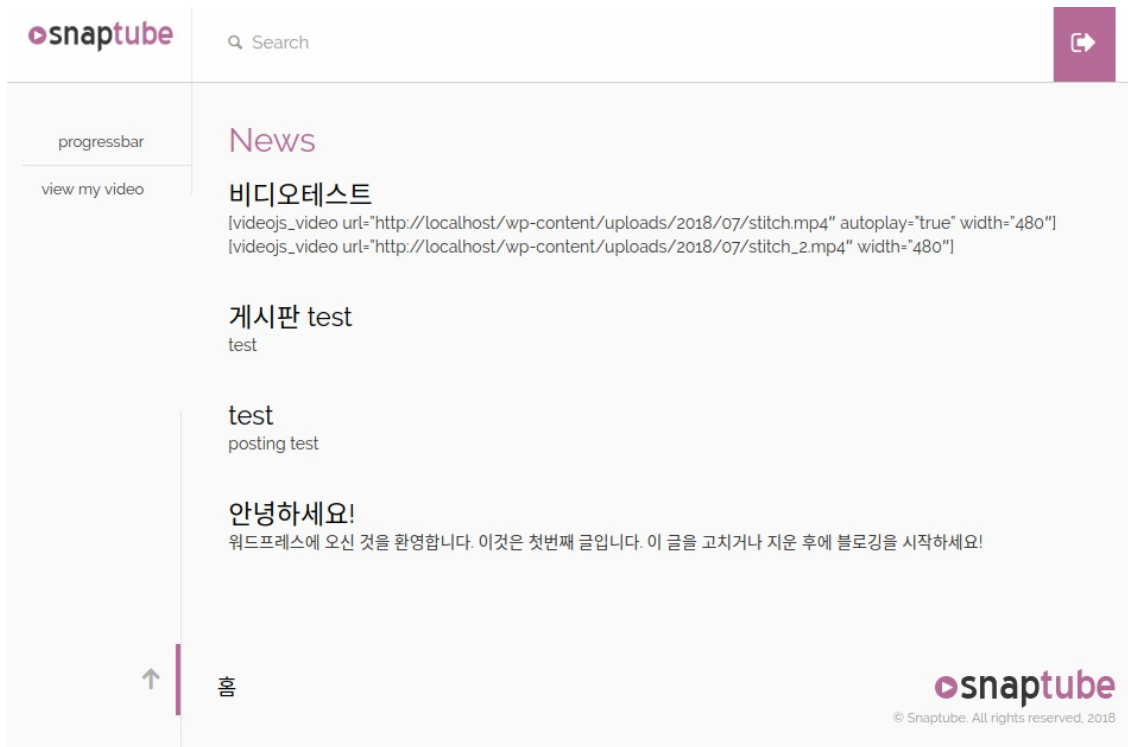
2.2. Wordpress 설치 및 세팅

2.2.1. Wordpress 설치

2.2.2. PlugIn 설치

Wordpress 환경에서 각 기능과 요구사항을 구현하기 위한 각종 플러그인 설치

- SnapTube
 - 프로젝트 진행 시 결과물이 실행 될 wordpress theme
 - SnapTube 화면



• PHP code snippets(Insert PHP)

- 워드프레스 페이지 내에서 동작 할 PHP 코드 삽입을 위한 플러그인
- `<? php , ?>` 태그 대신 `[insert_php]` , `[/insert_php]` 태그 안에 php 코드를 내장

Issue of 2.2. PHP code snippets(Insert PHP)

Wordpress 페이지 내 PHP 코드 삽입에 관련한 이슈

1. Issue 정의

- Wordpress 페이지 내 PHP 코드를 삽입하여 서버와 연동하는 것이 요구됨
- Wordpress templates 디렉터리에 템플릿 파일을 추가 후 PHP 코드를 작성하는 방법이 존재하였으나, 해당 템플릿 파일에 SnapTube 테마가 적용되지 않아 해당 방법을 적용 불가
- Wordpress 페이지 내 숏코드로 PHP 코드를 삽입하였으나 Wordpress에서 PHP 언어를 숏코드 지원하지 않음

2. Issue 해결 방안

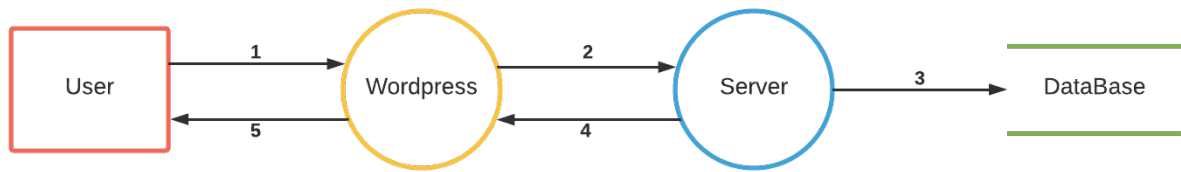
- Wordpress 페이지에 PHP 코드를 숏코드로 삽입 가능한 플러그인 설치
- `[insert_php]` , `[/insert_php]` 태그 내에 있는 PHP 코드가 숏코드로 페이지 내에서 동작 가능

• WP Mail SMTP

- Wordpress에서 회원 가입 시 가입 확인 메일을 보내기 위한 플러그인
- 사용자 별 비디오 관리가 프로젝트의 목적이기 때문에 회원 가입을 통해 user를 생성해야 함

3. 비디오 삭제 및 비디오 파일 관리

3.1. 비디오 삭제



3.1.1. 사용자 - 삭제 요청

- 사용자는 삭제하고 싶은 비디오의 `delete` 버튼을 클릭

3.1.2. Wordpress - 서버의 php 파일 호출

- 사용자의 링크 클릭 이벤트가 발생하면 Wordpress는 `delete` 버튼과 연결된 링크로 routing
- routing 정보

Method	URI	Code
GET	http://localhost/update_video_status.php?video_id=\$video_id	200

3.1.3. 서버 - 비디오 상태 변경

- routing 된 `update_video_status.php` 파일에서는 MySQL wordpress db에 접속하여 wp-uploaded_video 테이블에서 전달 받은 video id와 쿠키로 추출한 user name을 사용하여 사용자의 유효성을 검사
 - wp-uploaded_video 테이블에서 video id로 해당 비디오의 author를 쿼리
 - 쿠키로 추출한 username으로 wp-users 테이블에서 user id 쿼리 후 author와 동일하면 비디오의 status를 업데이트 하는 쿼리를 실행
- 사용자 유효성 검사가 성공적으로 수행되면 삭제 요청 받은 비디오의 status를 `delete` 로 변경

- `update_video_status.php` 호출 전 MySQL wp-uploaded_video 테이블

#	video_id	title	file_path	size	autho	uploaded_time	type	status
1	37	stitch.mp4	/var/www/html/wp-content/uploads/1/stitch.mp4	774258	1	2018-08-07 08:30:16	video/mp4	uploaded
2	39	long_stitch.mp4	/var/www/html/wp-content/uploads/1/long_stitch.mp4	5980324	1	2018-08-07 08:30:29	video/mp4	uploaded

- `update_video_status.php` 호출 후 MySQL wp-uploaded_video 테이블

#	video_id	title	file_path	size	autho	uploaded_time	type	status
17	37	stitch.mp4	/var/www/html/wp-content/uploads/1/stitch.mp4	774258	1	2018-08-07 08:34:39	video/mp4	delete
18	39	long_stitch.mp4	/var/www/html/wp-content/uploads/1/long_stitch.mp4	5980324	1	2018-08-07 08:30:29	video/mp4	uploaded

Issue of 1.3.

사용자 유효성 검사

1. Issue 정의

- 초기에는 Wordpress에서 GET Method로 변수를 전달 받을 때 video id와 username을 함께 받음
- 위와 같은 logic은 보안에 매우 취약함

2. Issue 해결 방안

- username 등의 user 정보는 쿠키로도 추출 가능
- routing 할 때는 video id만 전달
- 삭제를 요청한 user의 id는 서버에 있는 `update_video_status.php` 에서 쿠키 정보에서 추출

3.1.4. 서버 - Wordpress의 비디오 리스트 페이지 호출



- db 업데이트가 정상적으로 완료되면 다시 비디오 리스트 페이지를 호출함

3.1.5. Wordpress - 사용자에게 비디오 삭제 프로세스가 실행된 화면을 띄움


- Wordpress의 비디오 리스트 화면은 video의 status가 `delete` 가 아닌 비디오를 리스트에 출력하기 때문에 해당 프로세스를 거친 비디오는 리스트에서 보이지 않음

- `delete` 버튼 누르기 전 비디오 리스트 화면

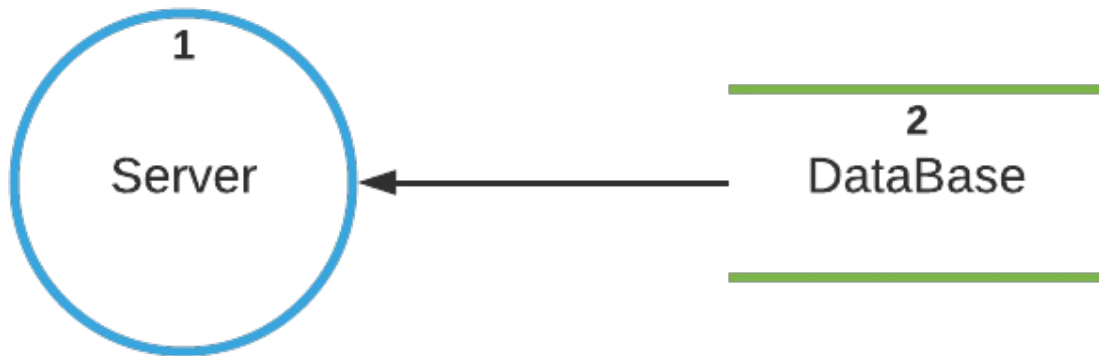
my_video_list
Home > my_video_list

thumbnail	title	size	uploaded_time	type	delete
	stitch.mp4	756.11 KB	2018-08-07 08:30:16	video/mp4	delete
	long_stitch.mp4	5.70 MB	2018-08-07 08:30:29	video/mp4	delete

- `delete` 버튼 누른 후 비디오 리스트 화면

my_video_list					
Home > my_video_list					
thumbnail	title	size	uploaded_time	type	delete
					
	long_stitch.mp4	5.70 MB	2018-08-07 08:30:29	video/mp4	delete

3.2. 비디오 파일 관리



- 서비스 관리 차원에서 wordpress에서 delete 버튼을 누른 즉시 서버와 DB에서 비디오 관련 정보를 삭제하지 않음
- 삭제된 비디오의 모든 정보를 저장하고 있으면 서버 메모리 이슈가 발생하기 때문에 서버 관리자 측의 별도 관리 프로세스가 필요함

3.2.1. 서버 - crontab을 등록하여 비디오 파일 삭제

- 매일 자정 실행되는 crontab을 등록하여 비디오 파일을 삭제하는 `delete_video_from_server.sh` 파일을 실행
- `delete_video_from_server.sh`
 - db 내에 status가 'delete' 인 비디오의 파일 경로에서 파일 포맷을 제외한 경로 쿼리 (비디오.mp4 파일과 해당 비디오의 썸네일인 비디오.png 파일 모두를 삭제하기 위함)
 - crontab


```
* * * * * sh /var/www/html/delete_video_from_server.sh
```
 - crontab이 실행되기 전

```
-rw-r--r-- 1 www-data www-data 5980324 8월 7 20:30 long_stitch.mp4
-rw-r--r-- 1 www-data www-data 444037 8월 7 20:30 long_stitch.png
-rw-r--r-- 1 www-data www-data 774258 8월 7 20:30 stitch.mp4
-rw-r--r-- 1 www-data www-data 96296 8월 7 20:30 stitch.png
```

- crontab이 실행된 후

```
-rw-r--r-- 1 www-data www-data 5980324 8월 7 20:30 long_stitch.mp4
-rw-r--r-- 1 www-data www-data 444037 8월 7 20:30 long_stitch.png
```

Issue of 2.1.

권한 이슈

1. Issue 정의

- delete_video_from_server.sh 파일을 실행하는 crontab이 동작하지 않음
- 우분투 내에서 삭제하고자 하는 비디오 파일의 user와 crontab을 등록한 user가 서로 달라 비디오 삭제 권한이 없음
- apache runuser를 root로 변경하여 시도 했으나 wordpress 웹 호스트 group과 user가 모두 root가 아니어서 apache가 실행되지 않음
- 비디오 파일이 업로드 되는 디렉터리의 권한을 777로 변경하고 진행하였으나 보안에 매우 취약한 구조
- sh 파일 내 sudo 명령어를 삽입하였으며 비밀번호 이슈가 추가 발생

2. Issue 해결방안

- 여러 방법을 시도하였으나 sh 파일 내에 sudo 를 삽입하여 진행하는 방법이 다른 디렉터리 혹은 서비스에 영향을 주지 않음
- crontab을 등록한 user에게 sudo 명령어 입력 시 비밀번호를 입력단계를 skip하도록 설정

3.2.2. DB - event scheduler를 등록하여 비디오 정보 삭제

- event scheduler를 등록하여 한 달에 한 번 비디오의 status가 'delete' 인 데이터들을 삭제
 - MySQL에 등록된 event scheduler

Name	Row Format	Time Zone	Type	Execute at	Interval value	Interval field	Starts	Ends	Status
 delete_video	root@localhost	SYSTEM	RECURRING		1	MONTH	2018-07-13 15:47:59		ENABLED

Issue of 2.2.

서버와 DB 간 비디오 삭제 실행 시점 이슈

1. Issue 정의

- 초기에 db에 등록된 event scheduler와 서버의 crontab 동작 시점을 같은 주기로 설정
- db에서 이미 정보가 삭제된 비디오는 서버에서 쿼리 해오지 않기 때문에 삭제하지 않음

2. Issue 해결 방안

- crontab과 event scheduler를 각각 다른 주기로 설정
- 서버의 데이터는 사용자가 같은 이름의 비디오 파일을 업로드 할 가능성이 높기 때문에 매일 삭제를 진행
- db 정보는 추후 발생 될 사용자 요구(ex. 사용자가 자신의 지우지 않은 비디오가 삭제되

었다는 컴플레인을 하거나, 실수로 삭제했을 때 복구 요청 시)에 대응하기 위해 한 달에 한번 동작하도록 설정

4. 후기

이번 프로젝트는 인턴 첫 프로젝트이며, 처음 접해보는 것들이 많아 난항을 겪었지만 그 과정에서 얻는 배움과 지식도 커서 프로젝트에 대한 애착이 많았다. wordpress라는 블로그 플랫폼도 처음 다뤄보았고, python 이외의 다른 언어로 직접 어떠한 기능을 구현하는 것도 처음이나 다름 없었다.

본 프로젝트를 진행하면서 새로운 툴과 언어에 대한 이해도 생겼지만, 원하고자 하는 해답을 위한 키워드로 에러를 검색하는 능력과 에러를 해결하는 여러 방법 중 나의 케이스에 잘 맞고 그 방법을 적용 시키는 능력 등 코딩 외의 스킬도 많이 향상된 것을 느꼈다. 또한 혼자 코딩을 하는 것이 아니라 각 파트 별 기능들이 연계되어 있고, 내가 구현하는 기능의 코드가 다른 사람이 이해할 수 있어야 했기 때문에 논리적이며 합리적인 코딩을 하기 위해 신경을 많이 쓴 점도 좋은 경험이었다.