PC 사용자 로그인/로그아웃 관제

PC의 작업스케쥴러 등록을 통한 로그인/로그아웃 관제 방법을 소개합니다.

PC의 작업스케쥴러 등록은 아래의 링크를 참고바랍니다.

https://soccerda.tistory.com/111

몇가지 주의사항

- 등록 시, 사용자의 로그온 여부와 관계없이 실행을 선택합니다.
- 암호를 저장하지 않습니다. 선택
- 동작은 아래와 같이 등록합니다.
    - 동작 : 프로그램 시작
    - 프로그램 : curl
    - 인수추가(옵션) : curl 을 통해 메신저 호출 파라메터 등록
        - 텔레그램 호출 예

        -
          ```
          -k -X POST -H “Content-type:application/json” https://api.telegram.org/bot{token}/sendmessage -d “{ \”chat_id\”: \”챗봇아이디\”, \”text\”: \”%computername%(home) logout : %username% remote access!!!\” }”
          ```

등록할 트리거 이벤트 예시

https://nas.nightfly.kr/sharing/R6K8LcM4y

---

harbor letsencrypt 인증서 갱신

Nginx Proxy Manager 를 통하여 let's encrypt ssl 인증서를 갱신한다.

해당 인증서의 디렉토리를 확인한다 (예 npm-43)

인증서를 복사한다.

```
cd [harbor설치홈]/data/secret
cp [proxyManager설치홈]/letsencrypt/live/npm-43/fullchain.pem cert/server.crt
cp [proxyManager설치홈]/letsencrypt/live/npm-43/privkey.pem cert/server.key
```

harbor 를 재시작한다.

```
docker-compose down
docker-compose up -d
```

Banyazavi T-Sharp 설치

https://www.clien.net/service/board/cm_nas/16430755

---

rocketchat mongodb migrate (mmap -> wiredTiger)

아래 절차는 major 버전 단계별로 1~5 를 수행한다. (버전 5에서 7로 변경 시, 버전 5 수행 > 버전 6 수행 > 버전 7 수행)

1. backup
    - docker-compose exec mongo mongodump –archive=/data/**mmapdump.gz** –gzip
    - container 안의 mmapdump.gz 를 외부로 가져온다.

2. container down
    - docker-compose down
    - db 디렉토리 rename (만약을 위해)

3. docker-compose 에서 몽고db 만 재시작 (변경할 버전으로 시작)
4. 복구
    - container 에 접근
    - docker-compose exec mongo bash
    - 기존 replica 셋으로 초기화 (rocket chat docker-compose 파일은 rs0 로 설정함)
      mongo 실행 : mongo 또는 mongosh
      // init/
      config = { _id : “rs0”, members: [{_id:0,host : “mongo:27017”}] }
      rs.initiate(config);

5. 데이터 복원
    - 컨테이너 안으로 백업한 데이터 전달
    - 데이터 복원
        - mongorestore –drop –archive=./**mmapdump.gz** –gzip –noIndexRestore

6. 컨테이너 재시작

---

docker 방화벽 설정

docker 컨테이너의 보안을 위해 iptables 을 설정하는 방안에 대해 기술해 놓고자 한다.

docker는 iptables 에 3개의 체인을 관리한다.

- 종류
    - DOCKER-USER
        - 관리자가 지정하는 룰
    - DOCKER
        - 외부설정 포트에서 내부 아이피의 포트 연동
    - DOCKER-ISOLATION-STAGE-#
        - 잘못된 인터페이스 경유 시, 삭제처리

패킷은 항상 DOCKER, DOCKER-USER 를 통해서 진입한다. (기본적으로 iptables을 수정하는 것을 비추천한다고 함. 자동 설정을 믿고 써라?)

---

docker-compose 프로젝트명 지정

docker-compose 로 실행 시, 기본적으로 디렉토리명이 프로젝트명으로 선언되어 컨테이너가 실행된다.

예를 들어, “디렉토리”_”컨테이너명”_1 과 같은 형식이다.

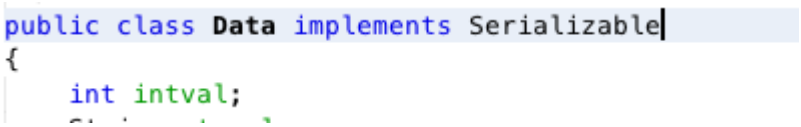
하지만, jenkins 등으로 컨테이너를 올릴 경우 다른 이름으로 프로젝트명이 지정되는 경우가 발생한다. 이럴 경우 명령창에서 수행한 컨테이너와 이름이 틀려 다시 올라가는데, 이때 포트가 충돌하여 안올라가는 경우가 발생한다.

그럼 프로젝트명을 고정할 수 없을까?

환경변수로 COMPOSE_PROJECT_NAME 을 지정하면 된다.

예를 들어, .env 파일을 만들어 COMPOSE_PROJECT_NAME=sample 이라고 기록해 놓으면 실행 시, 프로젝트명은 디렉토리명이 아닌 sample 이 된다.

.env 파일처럼 앞에 점(.)을 붙이기 어려우면, sample.env 등과 같이 별도 파일에 기록한 후, docker-compose.yml 파일에 환경설정파일을 지정하면 된다.



지정방법 : 각 컨테이너 하위에 아래와 같이 기록

env_file : sample.env

인식하는 순서 : docker-compose 파라메터 (-p) > .env > sample.env

Java HashMap 파일 입출력 (File IO)

[펌]
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=lsj30224&logNo=220586099250

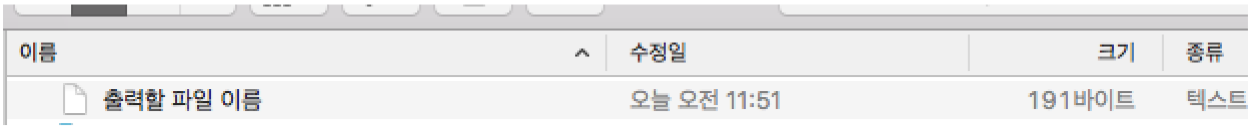
HashMap파일 입출력방법

출력 :

주의!!! 자료형처럼 쓰는 클래스는 Serializable 을 구현해야함

```
public class Data implements Serializable
{
    int intval;
```

```
    public static void main(String[] args) {
          HashMap<String, Data> hm = new HashMap<String, Data>();
          hm.put("Key1", new Data(1, "Key1 String Value", true));
        try {
                        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("출력할 파일 이름"));
            oos.writeObject(hm);
            oos.close();
        } catch (IOException ex) {
Logger.getLogger(Tester.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```



입력 (위에서 출력한 파일 받아옴)

받아올때는 자료형 변환 해줘야 합니다.

(이것도 좀 기네요 오른쪽으로 스크롤 해주세요)

```
    public static void main(String[] args) {
          HashMap<String, Data> hm = new HashMap<String,
Data>();
        try {
            ObjectInputStream ois = new ObjectInputStream(new
FileInputStream("출력할 파일 이름"));
             hm = (HashMap<String, Data>) ois.readObject();
//캐스팅 해줘야됨
        } catch (IOException ex) {
Logger.getLogger(Tester.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {
Logger.getLogger(Tester.class.getName()).log(Level.SEVERE,
null, ex);
        }
                            System.out.println("intval   :
"+hm.get("Key1").intval+", strval : "+hm.get("Key1").strval+",
boolval : "+hm.get("Key1").boolval);
    }
```

```
run:
intval : 1, strval : Key1 String Value, boolval : true
BUILD SUCCESSFUL (total time: 1 second)
```

---

mysql 파티션 추가

파티션 현황 조회

```
SELECT   TABLE_SCHEMA,   TABLE_NAME,   PARTITION_NAME,
```

```
PARTITION_ORDINAL_POSITION, TABLE_ROWS
FROM information_schema.PARTITIONS
WHERE TABLE_NAME = '테이블명';
```

파티션 생성

- MAXVALUE 를 사용하면 나중에 파티션을 추가할 수 없다. 추가가 필요한 경우, 파티션 생성을 새로 해야 된다. (데이터가 많으면 오래걸림)

```
ALTER TABLE 테이블명 PARTITION BY RANGE(TO_DAYS(필드명))(
PARTITION p20211201 VALUES LESS THAN (TO_DAYS('2022-01-01')),
PARTITION p20220101 VALUES LESS THAN (TO_DAYS('2022-02-01'))
);
```

파티션 추가

```
ALTER TABLE 테이블명 ADD PARTITION (
PARTITION P20220201 VALUES LESS THAN (TO_DAYS('2022-03-01')),
PARTITION P20220301 VALUES LESS THAN (TO_DAYS('2022-04-01'))
);
```

멀티사이트 워드프레스 도메인 변경하기

참조 : https://sheldon.co.kr/multisite_wordpress_domain_change/

워드프레스 를 로컬상태로 구축하고 웹호스팅으로 이전 방법시 참고 사항입니다.

워드프레스 멀티사이트를 사용하는 경우에는 데이터베이스를 직접 수동으로 수정해 주어야 합니다.

먼저 도메인을 변경하기에 앞서 혹시 모를 경우를 대비해 데이터를 백업해 둡니다. ftp에 접속하여 wp-content폴더를 복사해 내 컴퓨터에 저장해 두고, phpMyAdmin에 접속해 내보내기한 DB 역시 따로 저장해 둡니다.

1. phpMyAdmin에 접속한 후, 다음 **5**가지 테이블을 찾아 선택한 후 수정합니다.

- wp_options: siteurl과 home
- wp_site: 도메인 컬럼에 있는 예전 주소를 새 주소로 변경
- wp_sitemeta: siteurl

- wp_blogs: domain컬럼에 있는 예전 도메인 이름을 모두 새 도메인 이름으로 변경(멀티사이트 url)
- wp_#_options: 멀티사이트 개수만큼 생성된 wp_#_option의 예전 주소를 새 주소로 변경함.(ex. wp_2_options 테이블을 찾아 도메인 변경. 혹은 wp_3_options, wp_4_options 등등.. 멀티사이트 개수만큼 생성되어 있음)

2.ftp에 접속하여 **wp-config.php**파일을 찾아 에디터로 열어 준 후, 다음 코드를 추가합니다.

사이트 주소 및 DB 정보와 DB 패스워드를 변경합니다. (*멀티사이트가 있을시 해당 부분도 수정합니다.)

3.wp-config.php파일을 일부 수정합니다.

예전도메인을 define('DOMAIN_CURRENT_SITE', '예전도메인.com');

새로운 도메인으로 변경 define('DOMAIN_CURRENT_SITE', '신규도메인.com');

---

intelliJ gradle 버전 변경

참조 : https://namsick96.github.io/build%20tool/Gradle_version_change_at_Intellij/

프로젝트의 gradle 버전은 gradle-wrapper.properties에서 확인

터미널을 열어 아래와 같이 명령어를 통해 버전 변경

```
./gradlew wrapper --gradle-version 5.6.1
```

Settings > Build,Execution,Deployment > Build Tools > Gradle에서 하단의 Gradle 설정에서 Use Gradle from 설정을 'gradle-wrapper.properties' file로 설정