**< 동작환경 >**

웹페이지는 기본적으로 Chrome에서 동작합니다.

**< 전체 프로젝트 실행방법 >**

자바 프로젝트의 Main Class는 MainClass.main()을 run하게 되면 실행되어지는데, 이 때 main실행의 argument로 총 2개가 들어가지는데 그 중 첫번째로 필요한 argument는 20151211과 같은 form의 날짜, 두번째로는 수행번호입니다. 이 때, 수행번호 1~6번을 통해 각각의 클래스를 실행시킬 수도 있고 0번을 주어 1~6에서 수행되어지는 모든 클래스를 한번에 동작하게 할 수도 있습니다.

**< 크롤러 수행 방법 >**

1. 리눅스 환경에서 jar파일을 활용해서 수행하기.

java -jar jar파일이름(dailyTrendAnalist-1.0-SNAPSHOT-jar-with-dependencies.jar) 수행할날짜 수행번호

jar파일은 소스파일 - httpClient - target 폴더에 있습니다.

최상위 폴더에 dailyTrendAnalist-1.0-SNAPSHOT-jar-with-dependencies.jar 파일을 사용하시면 됩니다.

2. gui에서 바로 수행.

먼저 아규먼트를 설정해준다. 아규먼트는 리눅스와 마찬가지로 첫 번째는 수행할 날짜, 두 번째는 수행번호를 주고 수행한다.

만약 데이터베이스에 접근하지 못할 경우, 또는 새로운 데이터 베이스에서 수행할 경우, 로컬로 수행할 경우 등 데이터베이스의 주소를 바꾸는 법은 MainClass의 con = DriverManager.getConnection("jdbc:mysql://연결주소/디비이름", "사용자명", "비밀번호") 부분을 수정한다.

3. jar파일 만들기.

본 코드는 인텔리제이와 메이븐을 기반으로 작성되었습니다. 메이븐의 Lifecycle에서 install을 수행시켜서 jar파일을 만듭니다. 다른 방법을 통해서 수행하면 dependency들을 포함하지 않아 제대로 수행되지 않을 수 있습니다. 이 방법을 수행할 경우 프로젝트의 target폴더에 dailyTrendAnalist-1.0-SNAPSHOT-jar-with-dependencies.jar과 dailyTrendAnalist-1.0-SNAPSHOT.jar의 두 개의 파일이 생성되는데, 후자의 경우 dependency를 포함하지 않은 jar파일이기 때문에 수행하는 곳에서 dependency설정을 따로 해줘야 하는 불편함이 있습니다. 그래서 전자를 통해 수행하시기 바랍니다.

- 날짜의 경우

날짜를 yesterday로 줄 경우, 바로 어제자의 뉴스들을 수집하며 다로 날짜를 줄 경우 (ex. 20151111과 같은 4자리의 년, 2자리의 월, 2자리의 일) 해당 날짜의 뉴스를 수집합니다.

- 수행번호의 경우

0 :

(NewsCrawler 클래스 수행) -> 형태소 분석(ContentTokenize 클래스 수행) -> tf-idf와 연관어 찾기(AnalysisClass 클래스 수행) -> 디비 데이터 엑셀로 추출(ExtractIntoExcelClass 수행) -> 토픽 랭킹화, 연관어 랭킹화(TopicClass, AssociatedWordClass 클래스 수행) -> 랭킹의 이미지 추출(GetImage 클래스 수행)

1 : 크롤링만 수행

2 : 형태소 분석만 수행

3 : tf-idf와 연관어 찾기만 수행

4 : 디비 데이터를 엑셀로 추출만 수행

(데이터 추출의 경우 데이터베이스에 들어간 데이터를 엑셀로 확인하기 위해 엑셀로 추출함.)

5 : 토픽 랭킹화, 연관어 랭킹화만 수행

6 : 랭킹의 이미지 추출만 수행

**< 서버 수행 방법 >**

서버의 실행 파일은 소스파일 - DTA - app.js 입니다.

먼저 node js를 서버를 수행할 컴퓨터에 설치해 놓고 윈도우의 경우 cmd창, linux일 경우엔 console창에서 'node 서버폴더의 app.js'로 수행한다. 예를 들어 서버 폴더일 경우 'node app.js'로 수행. 수행 폴더가 아닐 경우 폴더까지의 경로도 잡아준다.

**< 데이터 베이스 쿼리문 >**

--------------------------------------------------------------------------------

-- MySQL Script generated by MySQL Workbench

-- 12/07/15 13:03:55

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`NEWS`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`NEWS` (

`NEWS\_NO` INT NOT NULL AUTO\_INCREMENT COMMENT '',

`NEWS\_TITLE` VARCHAR(100) NULL COMMENT '',

`NEWS\_URL` VARCHAR(200) NULL COMMENT '',

`NEWS\_TIME` VARCHAR(50) NULL COMMENT '',

`NEWS\_CONTENT` VARCHAR(10000) NULL COMMENT '',

`NEWS\_CO` VARCHAR(20) NULL COMMENT '',

`NEWS\_CONTENT\_MOD` VARCHAR(10000) NULL COMMENT '',

`NEWS\_YYYYDDMM` VARCHAR(10) NULL COMMENT '',

PRIMARY KEY (`NEWS\_NO`) COMMENT '')

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`NEWS\_NOUN`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`NEWS\_NOUN` (

`NEWS\_NO` INT NOT NULL COMMENT '',

`NEWS\_NOUN\_CONTENT` VARCHAR(10000) NULL COMMENT '',

INDEX `fk\_NEWS\_NOUN\_NEWS\_idx` (`NEWS\_NO` ASC) COMMENT '',

PRIMARY KEY (`NEWS\_NO`) COMMENT '',

CONSTRAINT `fk\_NEWS\_NOUN\_NEWS`

FOREIGN KEY (`NEWS\_NO`)

REFERENCES `mydb`.`NEWS` (`NEWS\_NO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`NEWS\_ANALYSIS`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`NEWS\_ANALYSIS` (

`NEWS\_NO` INT NOT NULL COMMENT '',

`NEWS\_TOPIC` VARCHAR(45) NULL COMMENT '',

`NEWS\_TOPIC\_TFIDF` DOUBLE NULL COMMENT '',

INDEX `fk\_NEWS\_ANALYSIS\_NEWS1\_idx` (`NEWS\_NO` ASC) COMMENT '',

PRIMARY KEY (`NEWS\_NO`) COMMENT '',

CONSTRAINT `fk\_NEWS\_ANALYSIS\_NEWS1`

FOREIGN KEY (`NEWS\_NO`)

REFERENCES `mydb`.`NEWS` (`NEWS\_NO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`TOPIC`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`TOPIC` (

`TOPIC` VARCHAR(45) NOT NULL COMMENT '',

`TOPIC\_RANK` INT NULL COMMENT '',

`TOPIC\_YYYYDDMM` VARCHAR(10) NULL COMMENT '',

`TOPIC\_NO` INT NOT NULL AUTO\_INCREMENT COMMENT '',

PRIMARY KEY (`TOPIC\_NO`) COMMENT '')

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`TOPIC\_ASSOCIATED`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`TOPIC\_ASSOCIATED` (

`TOPIC\_ASSOCIATED` VARCHAR(45) NOT NULL COMMENT '',

`TOPIC\_NO` INT NOT NULL COMMENT '',

`ASSOCIATED\_RANK` INT NULL COMMENT '',

`TOPIC` VARCHAR(45) NULL COMMENT '',

PRIMARY KEY (`TOPIC\_NO`, `TOPIC\_ASSOCIATED`) COMMENT '',

INDEX `fk\_table1\_TOPIC1\_idx` (`TOPIC\_NO` ASC) COMMENT '',

CONSTRAINT `fk\_table1\_TOPIC1`

FOREIGN KEY (`TOPIC\_NO`)

REFERENCES `mydb`.`TOPIC` (`TOPIC\_NO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

--------------------------------------------------------------------------------

**< 웹페이지 이용방법 >**

웹페이지에서 제공하는 페이지는 크게 두 가지의 페이지로, main 페이지에서는 보여지는 1~10까지의 아이콘을 통해 오늘을 기준으로 전날의 랭킹 1위에서 10위까지를 볼 수 있습니다. 보여지는 이미지의 왼쪽 하단부에 써있는 주제를 선택하게 되면 다음 페이지로 이동하게 되고, 이미지의 오른쪽 하단부에 써있는 Play/Pause Slideshow를 통해 이미지를 자동으로 순서대로 보여주게 됩니다. 주제를 선택하여 다음페이지로 이동하면 일별 랭킹 및 관계망, 관련 뉴스 기사를 볼 수 있습니다. 일별 랭킹에 해당하는 키워드를 누르면 그에 해당하는 관계망 및 뉴스 기사를 볼 수 있고, 달력에 해당하는 날짜를 누르게 되면 그 날에 해당하는 키워드의 순위를 볼 수 있습니다. 또한 관련 뉴스의 제목을 누르게 되면 해당 뉴스 페이지로 이동하여 직접 기사를 볼 수 있습니다. 현재 저희가 크롤링을 시행하여 볼 수 있는 데이터의 날짜는 **(11/29, 11/30, 12/01, 12/02, 12/03, 12/04, 12/05, 12/06, 12/07, 12/08) 이렇게 10일에** 해당하는 분량입니다.

**< 주의사항 >**

1. 크롤러 동작시, 보통 하루치의 기사가 1000개를 넘으므로 시간이 약 8시간정도 소요됩니다. 따라서 시험 동작시에는 중간에 끊으시고 다음 처리를 하시는 것을 추천드립니다.
2. 크롤러를 작동시키지 않고, 기존에 이미 분석되어 있는 데이터를 통해 그 다음의 과정(수행번호로 치자면 1~6)을 수행할 경우, 기존 디비에 데이터가 저장 되어있기 때문에 오류가 발생할 수 있습니다.
3. Maven을 통하여 Library를 추가하였기 때문에 해당 프로그램을 설치하셔야 동작합니다.