1. Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서
   1. 사용한 JVM: OpenJDK 17 / WAS 제품 등의 종류과 설정 값: Spring 내장 톰캣 / 버전: IntelliJ IDEA 2023.1.3
   2. 빌드 시 사용되는 환경 변수 등의 내용 상세 기재
   3. 배포 시 특이사항 기재
   4. DB 접속 정보 등 프로젝트 (ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록
   5. 데이터 부분- 언어 및 버전 : python 3.11+ (pip)  
      웹 프레임워크 : FastAPI  
      핵심 라이브러리 : APScheduler, BeautifulSoup이 서버는 사실상 스크래핑 서버다.  
      왜냐하면 URL 수집을 네이버 뉴스에서 제공해주기 때문에 따로 크롤러를 사용하지 않았다.  
      따라서, 스크래퍼를 사용해서 해당 뉴스의 URL을 가져온 후, 해당 URL의 내용을 스크래핑 하는 방식으로 동작한다.  
      스케줄러에  
      뉴스 기사의 URL 수집 하는 기능, 해당 URL의 내용을 수집하는 기능 그리고 302 에러일 경우 다시 URL을 수집하는 기능을 등록해놓았다.  
      그래서 서버를 한번만 실행시키면  
      URL과 Contents 스크래핑은 1분마다 실행되어 약 30건의 뉴스를 수집할 수 있고,  
      하루에 한 번씩 302에러에 대해 리디렉션 된 URL을 수집할 수 있다.

<하둡 관련>

**1. 서버에서 하둡 실행**

**1) 자바 설치**

* sudo apt install openjdk-8-jre-headless: java 8 버전 필요
* **ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa** : pw 없이 접근 허용

**2) 하둡 설치**

* sudo apt install wget : wget 다운로드
* sudo wget [<https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz>](<https://downloads.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz>) hadoop 3.3.6 버전 다운로드
* tar -xvzf hadoop-3.3.6.tar.gz 압축 해제
* vi ~/.bashrc .bashrc 파일 수정
* .bashrc 맨 아래에 추가
* export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
* export HADOOP\_HOME=/home/ubuntu/hadoop
* export HADOOP\_INSTALL=$HADOOP\_HOME
* export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME
* export HADOOP\_COMMON\_HOME=$HADOOP\_HOME
* export HADOOP\_HDFS\_HOME=$HADOOP\_HOME
* export HADOOP\_YARN\_HOME=$HADOOP\_HOME
* export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native
* export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin
* export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"

진행 후 source .bashrc로 적용

* xml 파일 수정(hadoop 폴더 내 etc/hadoop/ 내의 xml 파일 수정)
  + core-site.xml
  + <configuration>
  + <property>
  + <name>hadoop.tmp.dir</name>
  + <value>/home/ubuntu/hadoop\_tmp</value>
  + </property>
  + <property>
  + <name>fs.default.name</name>
  + <value>hdfs://0.0.0.0:19000</value>
  + </property>
  + </configuration>
  + hdfs-site.xml
  + <configuration>
  + <property>
  + <name>dfs.replication</name>
  + <value>1</value>
  + </property>
  + </configuration>
  + mapred-stie.xml
  + <configuration>
  + <property>
  + <name>mapreduce.framework.name</name>
  + <value>yarn</value>
  + </property>
  + <property>
  + <name>yarn.app.mapreduce.am.env</name>
  + <value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME</value>
  + </property>
  + <property>
  + <name>mapreduce.map.env</name>
  + <value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME</value>
  + </property>
  + <property>
  + <name>mapreduce.reduce.env</name>
  + <value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME</value>
  + </property>
  + </configuration>
  + yarn-site.xml
  + <configuration>
  + <!-- Site specific YARN configuration properties -->
  + <property>
  + <name>yarn.nodemanager.aux-services</name>
  + <value>mapreduce\_shuffle</value>
  + </property>
  + <property>
  + <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  + <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  + </property>
  + </configuration>
* [hadoop-env.sh](http://hadoop-env.sh) 환경 추가
* export JAVA\_HOME=/usr

**3) 실행 순서**

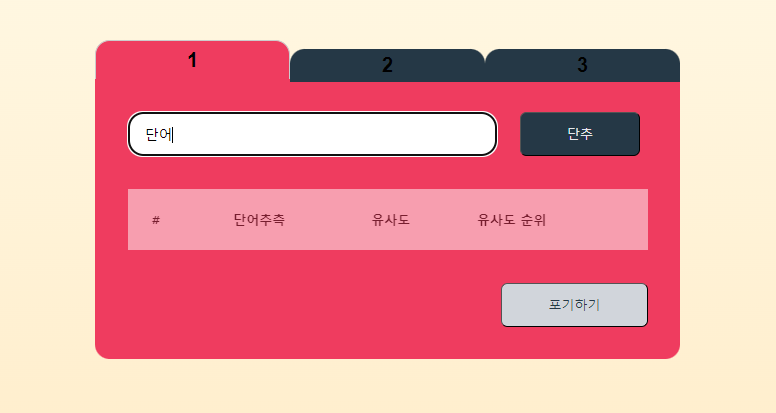
1. [start-all.sh](<http://start-all.sh>) : 실행 후 jps로 올라와있는 파일 확인
2. hdfs namenode -format :
3. hdfs dfs -ls / : 현재 hdfs 루트에 있는 파일 확인
4. hdfs dfs -put <파일경로 및 위치> <파일저장 위치> : 파일 hdfs에 입력 ex) hdfs dfs -put $HADOOP\_HOME/bin/news20230917.txt /
5. yarn jar <실행 jar 파일> <Driver에 저장된 시작 명령> <입력 파일> <출력 파일> : jar 파일로 wordcount 실행 ex) yarn jar $HADOOP\_HOME/bin/Wordcount.jar wordcount news extract
6. [stap-all.sh](http://stap-all.sh) : 하둡 Off

**2. Mecab 실행**

1. 프로젝트에서 사용하는 외부 서비스 정보를 정리하는 문서
   1. 소셜 인증: 없음
   2. 포톤 클라우드: 없음
   3. 코드 컴파일 등에 활용된 ‘외부 서비스’가입 및 활용에 필요한 정보: 없음
2. DB 덤프 파일 최신본
3. 사연 시나리오
   1. 시연 순서에 따른 site 화면별, 실행별(클릭 위치 등) 상세 설명

퀴즈 페이지

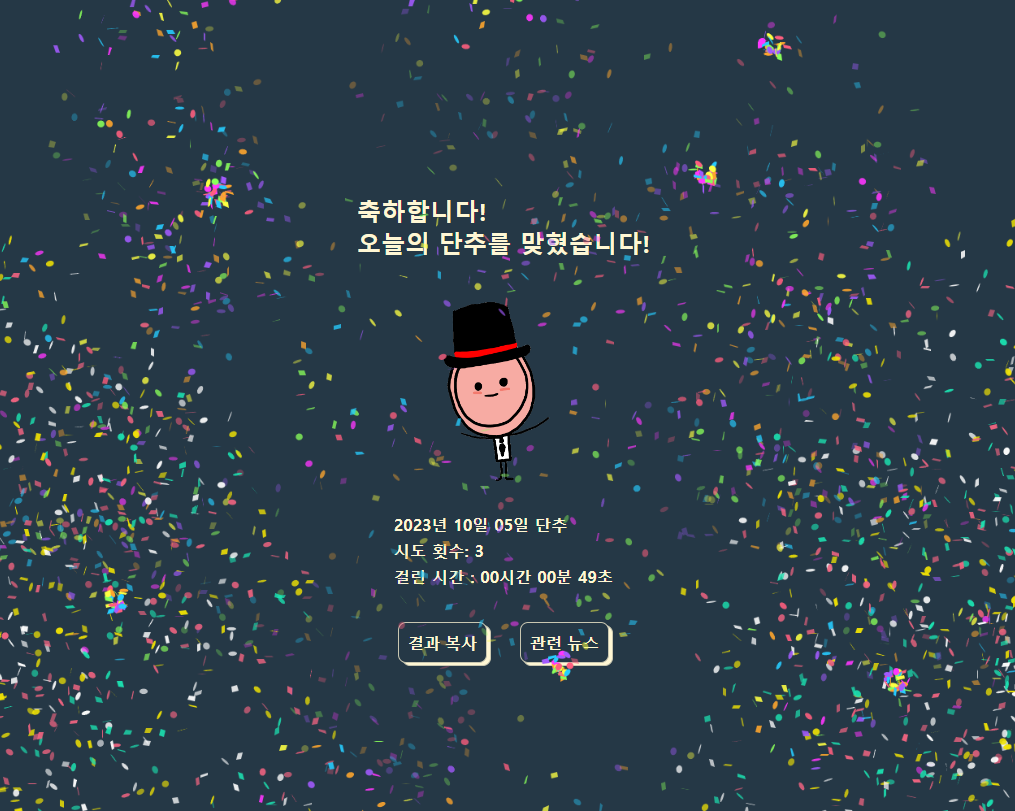
게임 설명

추측

퀴즈 변경 가능 

정답 입력 시



모든 정답 맞히면 

관련 뉴스 

뉴스 기사 클릭 시 해당 뉴스 기사로 이동