



KSB 인공지능 프레임워크 실행하기

KSB 인공지능 프레임워크를 실행하는 방법에 대해서 설명합니다.

아래와 같은 절차로 KSB 프레임워크를 구동합니다.

- . [Host PC] KSB 툴박스 docker 컨테이너 실행하기
- . [Docker 컨테이너] KSB 툴박스 docker 컨테이너 초기화
- . [Docker 컨테이너] KSB 인공지능 프레임워크 API serving 구동

[Host PC] KSB 툴박스 docker 컨테이너 실행하기

[다운로드](#) 페이지에서 KSB 툴박스 (ksb_toolbox_v1_0.tar.gz 파일)를 다운받아 압축을 해제하여 생성된 `/home/csle/ksb_csle/docker/1.0-host` 로 이동합니다.

아래의 명령을 통해 KSB 툴박스 docker 컨테이너를 실행하고 접속합니다.

```
csle@csle1:~/ksb-csle/docker/1.0-host$ ./startDockerCsle.sh
Error response from daemon: network csle_standalone not found
csle1
start csle1 slave container...
4918997cd2dac93fe0b52f3df3a45a46db6df1f3f052854061696695451d2f22
csle@csle1:/$
```

. (참고사항) KSB 툴박스 docker 컨테이너를 실행할 경우, Host pc의 /home/csle/ksb-csle 폴더를 volume 마운트해서 사용합니다. 따라서 예제 파일이나 모델 파일 등 사용자 파일은 ksb-csle 폴더를 통해 docker 컨테이너에서 접근 할 수 있습니다.

[Docker 컨테이너] KSB 툴박스 docker 컨테이너 초기화하기

KSB 툴박스 docker 컨테이너 접속한 후, 다양한 KSB 툴박스 예제들을 실행하기 위해서 아래 명령을 통해 docker 컨테이너를 초기화 합니다. Zookeeper/Hadoop/Kafka/Hbase/웹툴킷 등 초기화를 진행합니다. PC사양에 따라 다소 시간이 걸릴 수 있습니다. 아래와 같은 메시지가 나오면 초기화가 완료된 상태입니다. (host pc의 IP는 192.168.0.5로 가정합니다. 자신의 Host PC IP가 출력되면 정상입니다.)

```
csle@csle1: cd
csle@csle1:~$ ./startService.sh
* Restarting OpenBSD Secure Shell server sshd      [ OK ]
* Restarting PostgreSQL 9.6 database server        [ OK ]
ZooKeeper JMX enabled by default
Using config: /home/csle/zookeeper-3.4.9/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
Warning: Permanently added 'master,192.168.0.5' (ECDSA) to the list of
known hosts.
ZooKeeper JMX enabled by default
Using config: /home/csle/zookeeper-3.4.9/bin/../conf/zoo.cfg
Starting zookeeper ... already running as process 174.
Warning: Permanently added 'csle1,192.168.0.5' (ECDSA) to the list of
known hosts.
ZooKeeper JMX enabled by default
Using config: /home/csle/zookeeper-3.4.9/bin/../conf/zoo.cfg
Starting zookeeper ... already running as process 174.
Starting namenodes on [csle1]
csle1: Warning: Permanently added 'csle1,192.168.0.5' (ECDSA) to the
list of known hosts.
```

< 중간 생략 >

```
[info] Loading project definition from
/home/csle/ksb-knowledge/KBE/project
[info] Set current project to KBE
(in build file:/home/csle/ksb-knowledge/KBE/)
[warn] The start command is deprecated, and will be removed in a
future version of Play.
[warn] To run Play in production mode, run 'stage' instead, and then
execute the generated start script in target/universal/stage/bin.
[warn] To test your application using production mode, run 'testProd'
instead.
[info] Wrote /home/csle/ksb-knowledge/KBE/target/scala-2.11/
kbe_2.11-1.0-SNAPSHOT.pom
Warning: node.js detection failed, sbt will use the Rhino based
Tritreme JavaScript engine instead to run JavaScript assets compilation,
which in some cases may be orders of magnitude slower than using
node.js.
```

(Starting server. Type Ctrl+D to exit logs, the server will remain in background)

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>

  <title>Select inference rules</title>
  <link rel="stylesheet" media="screen"
    href="/assets/stylesheets/main.css">
  <link rel="shortcut icon" type="image/png"
    href="/assets/images/favicon.png">
  <script src="/assets/javascripts/hello.js"
    type="text/javascript"></script>
</head>
<body>

```

< 중간 생략 >

```

<!DOCTYPE html>
<html lang="en">
  <head>

    <title>Welcome to Play</title>
    <link rel="stylesheet" media="screen"
      href="/assets/stylesheets/main.css">
    <link rel="shortcut icon" type="image/png"
      href="/assets/images/favicon.png">
    <script src="/assets/javascripts/hello.js"
      type="text/javascript"></script>
  </head>
  <body>

    <h1>addDeviceControl</h1>
  </body>
</html>

```

```
csle@csle1:~$
```

jps명령을 이용하여 초기화 상태를 확인합니다.

```
csle@csle1:~$ jps
2336 HRegionServer
2161 HMaster
3681 Jps
1890 NodeManager
1219 DataNode
885 QuorumPeerMain
1413 SecondaryNameNode
1078 NameNode
3542 ProdServerStart
2475 Kafka
3228 Bootstrap
1597 ResourceManager
```

[Docker 컨테이너] KSB Base Ontology 로드하기

KSB 툴박스에 포함된 RealtimeIngestToServingWithKbInTwoEngines 예제를 실행하기 위해 온톨로지를 로드합니다.

```
cd ~/ksb-csle/bin/
./loadKnowledge_BaseOntologies.sh localhost 9876
```

[Docker 컨테이너] KSB 인공지능 프레임워크 구동하기

KSB 툴박스의 다양한 예제를 실행하기 위해 아래 명령으로 KSB 인공지능 프레임워크 API serving을 실행합니다.

```
csle@csle1:~/ksb-csle/bin$ ./startKsbApiServing.sh
```

KSB 인공지능 프레임워크 사용을 위한 모든 준비가 완료되었습니다.

이제 [사용자 가이드](#)로 이동하셔서 워크플로우 예제들을 실행해 보시기 바랍니다.

FAQ

1. 호스트 PC에서 **sudo** 명령을 할때 **sudo: could not resolve host** 에러가

날때

. [Host PC] 호스트 PC의 hostname이 csle1으로 설정되었는지 확인합니다.

. sudo vi /etc/hosts 명령을 통해 아래의 줄이 추가되었는지 확인합니다.

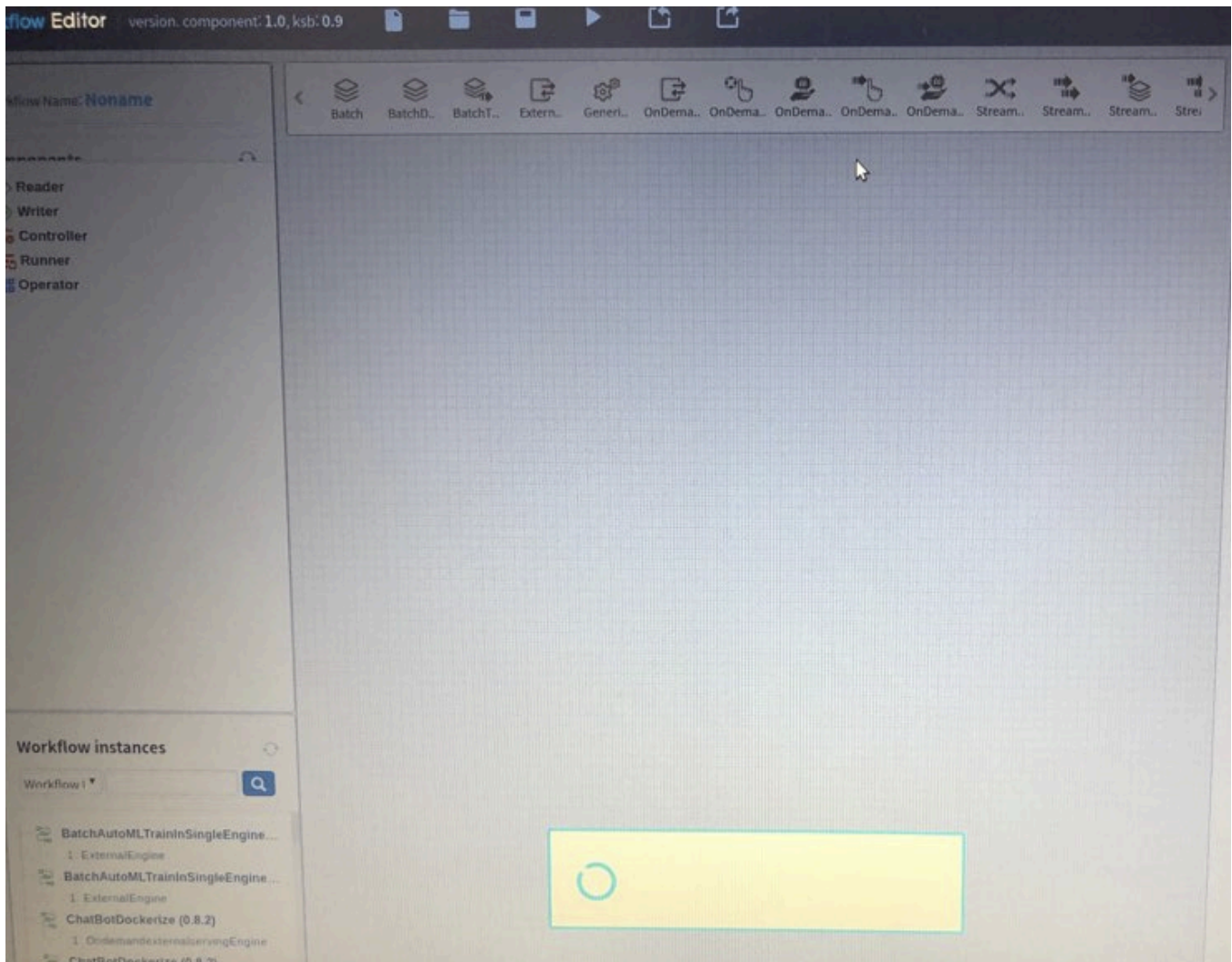
<자신의 IP> csle1 master

2. 웹툰킷 홈페이지(csle1:8080)에 접속이 안될때

. [Docker 컨테이너] tomcat이 제대로 구동되지 않았을 경우가 있습니다. 따라서 docker내에서 아래의 명령으로 tomcat을 재구동한 후 재접속합니다.

```
csle@csle1:/$ ~/ui_of_csle/apache-tomcat-7.0.81/bin/catalina.sh stop
csle@csle1:/$ ~/ui_of_csle/apache-tomcat-7.0.81/bin/catalina.sh start
```

3. 크롬을 이용해서 웹툰킷에 잘 접속 되었으나, 아래와 같이 웹페이지가 업데이트 되지 않고 멈춰있을때



. [Host PC] 기존의 웹툰킷 버전이 캐시에 남아있어서 문제가 되는 경우입니다.

먼저, 크롬 브라우저에서 F12를 클릭합니다. 그리고, 아래 그림과 같이 '새로고침' 아이콘에서 마우스 오른쪽 버튼을 클릭하면 '캐시 비우기 및 강력 새로고침'을 클릭합니다. 그리고 F12를 클릭해서 디버깅 모드를 빠져나옵니다.

