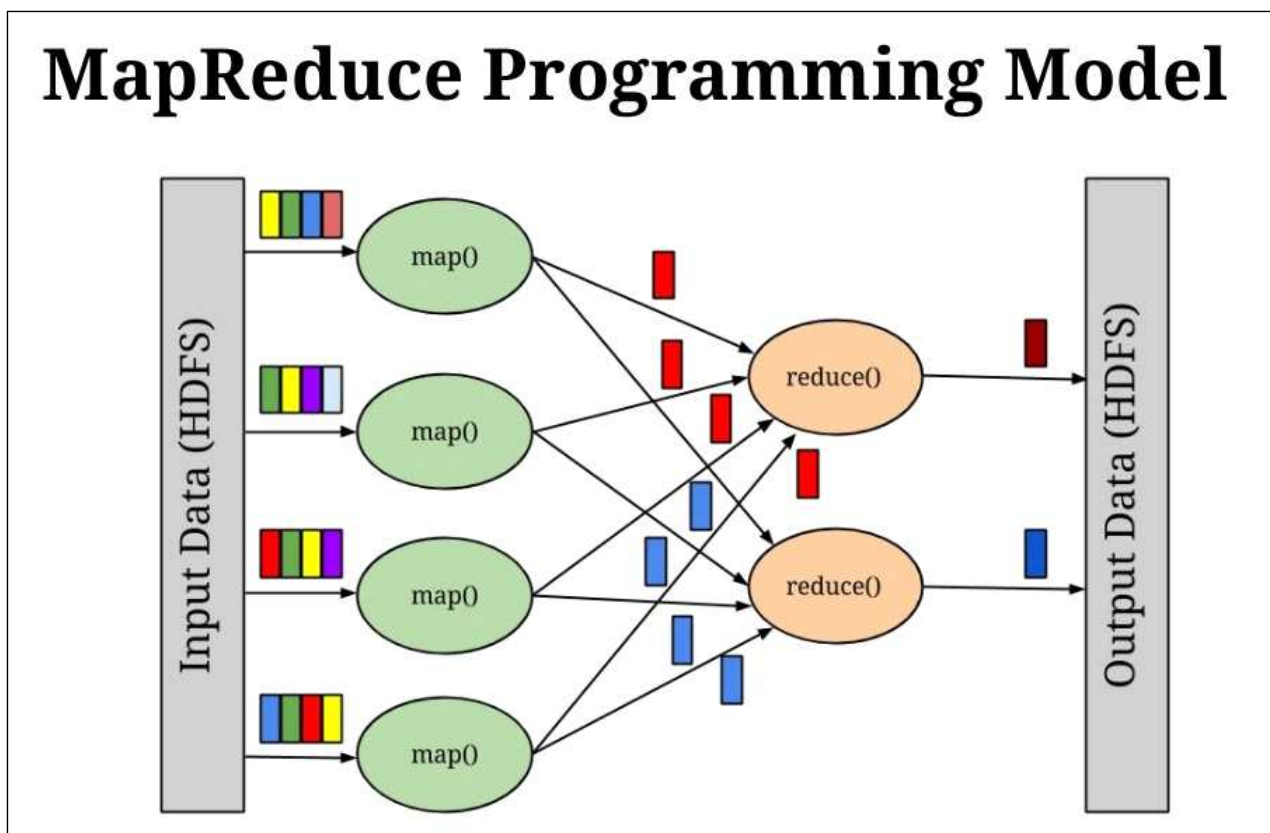


## ■ Spark 출현배경

### 1. Hadoop MapReduce의 한계

- 성능의 문제: 느린데다가 많은 Disk I/O
- 항상 HDFS에서 읽어서 작업, 단계별 처리를 하기 때문에 머신러닝같이 반복 작업에 비효율적
- Interactive Computing 부재
- 복잡한 configuration 설정과 튜닝



### 2. Hadoop과 Spark의 차이

#### ① 역할

- 하둡: 분산형 데이터 인프라 스트럭처이며, 대량의 데이터 구조를 서버 클러스터 환경 내에서 복수의 서버들에 분산시키는 역할
- 스파크: 하둡 또는 분산형 데이터 구조 위에서 동작하는 데이터 프로세싱 서비스

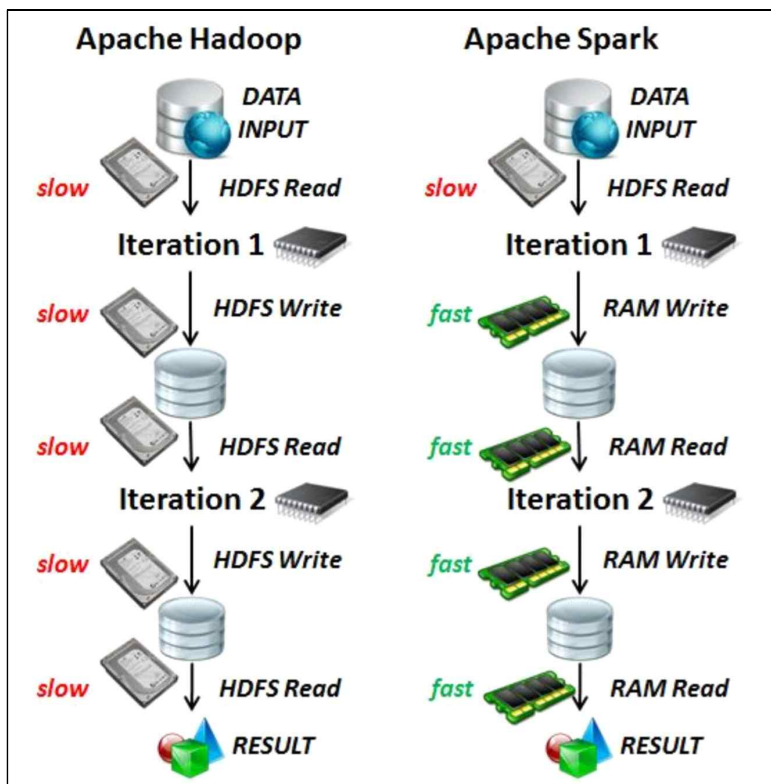
#### ② 필요관계

- 하둡: 기본 코어는 HDFS+YARN으로 구성되고 스파크가 반드시 필요하진 않음

- 스파크: 하둡 위에서만 실행되지는 않지만 하둡 기반으로 설계가 되었기 때문에 하둡 환경 위에서 최상의 성능이 나옴

### ③ 성능

- 하둡(Map Reduce)이 스파크보다 성능이 낮음
- 데이터 프로세싱의 차이에 기반
- 하둡은 Disk 기반, 스파크는 Memory 기반
- 하둡: 단계별 처리 방식
- 하둡: HDFS에서 데이터 Read -> 작업실행 -> HDFS에 데이터 Write -> 다시 update된 데이터 Read -> 다음 작업 후, HDFS에 데이터 Write
- 스파크: 전체 데이터셋을 한번에 처리
- 스파크: HDFS에서 데이터 Read -> 메모리상 필요 작업 실행 및 HDFS에 데이터 Write하는 모든 과정이 동시에 진행



### ④ 적용 영역

- 하둡: 정적인 데이터 운영 및 보고 등, 배치(Batch)의 경우 M/R로 구성하면 성능도 괜찮음
- 스파크: 스트리밍 데이터 처리, 머신러닝 알고리즘 기반의 비즈니스 로직 개발, 실시간 로그 분석 및 처리, ETL 등

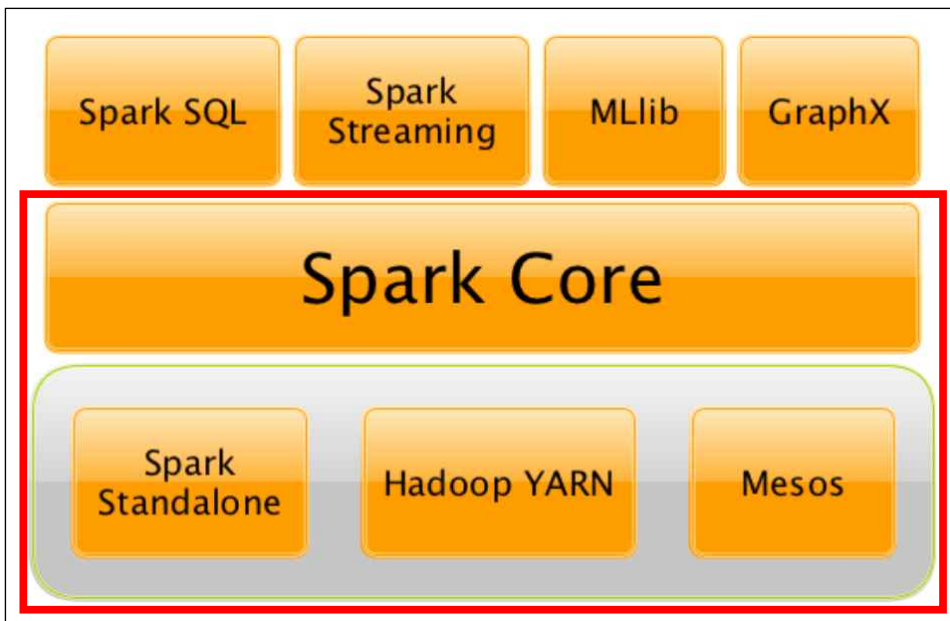
## ■ Spark 개요



- 빅데이터 처리를 위한 오픈소스 병렬 분산처리 플랫폼
- 인메모리 기반(In-Memory)의 대용량 데이터 고속 처리 엔진
- JAVA, Python, Scala, R을 기반으로 구동
- 2009년에 UC Berkeley AMPLab 연구 프로젝트로 처음 출현
- 2010년에 오픈소스로 공개
- 2013년에 아파치 인큐베이터 프로젝트로 선정

## ■ Spark 특징

- In-Memory Processing: 하둡의 MR(MapReduce) 작업처리 속도보다 100배 이상 빠른 성능 구현
- JAVA, Python, Scala, R 인터페이스 지원
- 8000개 이상의 노드 추가 가능한 확장성 보장
- 데이터 저장 및 활용을 위해 HDFS 뿐 아니라, Cassandra, HBase, Elasticsearch, Amazon S3 등 지원
- 분산 클러스터 컴퓨팅 환경 구축을 위해 3가지 환경을 지원
- 자체적인 Standalone Scheduler환경(Single / Cluster)



- 하둡의 종합 플랫폼인 YARN, Docker 가상화 플랫폼인 Mesos 위에서도 기동 가능
- 적절한 클러스터의 수를 설정하지 않는다면, Standalone 모드가 더 빠르기 때문에 수업에서는 Standalone 모드로 Spark 환경 구성
- Spark on yarn

The screenshot shows the Hadoop web interface for an application named 'application\_1529885501282\_0001'. The interface is divided into several sections:

- Cluster:** A sidebar menu with links to About, Nodes, Node Labels, Applications, NEW, NEW\_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, and Scheduler.
- Application Overview:** A section displaying application details:
  - User: bigdata
  - Name: Spark shell
  - Application Type: SPARK
  - Application Tags:
  - Application Priority: 0 (Higher Integer value indicates higher priority)
  - YarnApplicationState: FINISHED
  - Queue: default
  - FinalStatus Reported by AM: Application has not completed yet.
  - Started: 월 6월 25 09:40:48 +0900 2018
  - Elapsed: 23sec
  - Tracking URL: History
  - Log Aggregation Status: DISABLED
  - Application Timeout (Remaining Time): Unlimited
  - Diagnostics: Shutdown hook called before final status was reported.
  - Unmanaged Application: false
  - Application Node Label expression: <Not set>
  - AM container Node Label expression: <DEFAULT\_PARTITION>
- Application Metrics:** A section displaying resource usage:
  - Total Resource Preempted: <memory:0, vCores:0>
  - Total Number of Non-AM Containers Preempted: 0
  - Total Number of AM Containers Preempted: 0
  - Resource Preempted from Current Attempt: <memory:0, vCores:0>
  - Number of Non-AM Containers Preempted from Current Attempt: 0
  - Aggregate Resource Allocation: 25110 MB-seconds, 21 vcore-seconds
  - Aggregate Preempted Resource Allocation: 0 MB-seconds, 0 vcore-seconds
- Attempt Table:** A table showing application attempts with columns for Attempt ID, Started, Node, Logs, Nodes blacklisted by the app, and Nodes blacklisted by the system.
 

Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
appattemp 1529885501282_0001_000002	Mon Jun 25 09:40:59 +0900 2018	http://server04:8042	Logs	0	0
appattemp 1529885501282_0001_000001	Mon Jun 25 09:40:48 +0900 2018	http://server01:8042	Logs	0	0

## - Spark on mesos

Apache

Mesos

Frameworks

Agents

Roles

Offers

Maintenance

Master / Frameworks

Active Frameworks

Find...

ID ▾	Host	User	Name	Roles	Principal	Active Tasks	CPUs	GPUs	Mem	Disk	Max Share	Registered	Re-Registered
...bf8fe93f1af2-0000	server01	bigdata	Spark shell	*		0	0	0	0 B	0 B	0%	a minute ago	-

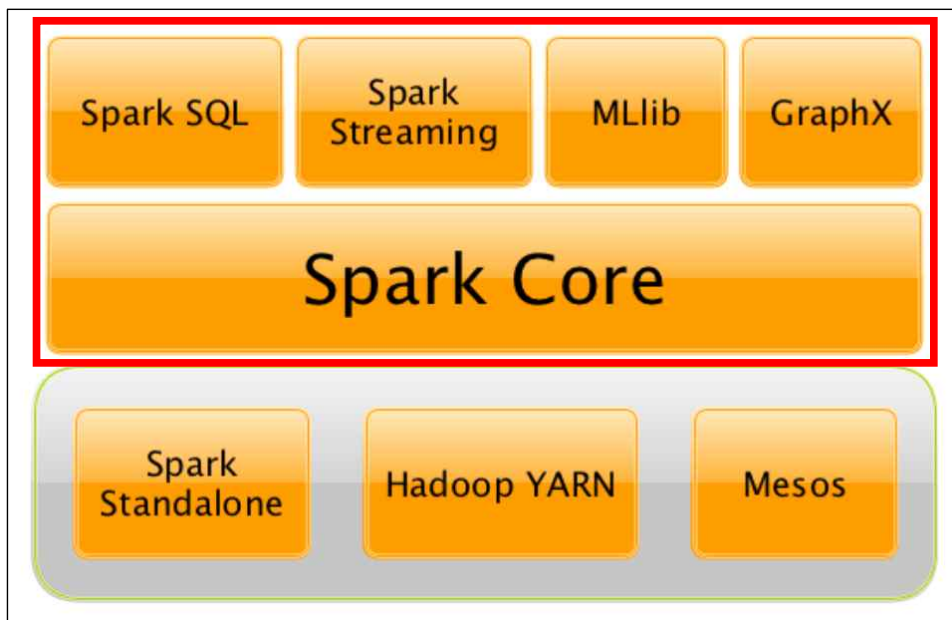
Inactive Frameworks

Find...

ID ▾	Host	User	Name	Roles	Principal	Active Tasks	CPUs	GPUs	Mem	Disk	Max Share	Registered	Re-Registered
------	------	------	------	-------	-----------	--------------	------	------	-----	------	-----------	------------	---------------

Completed Frameworks

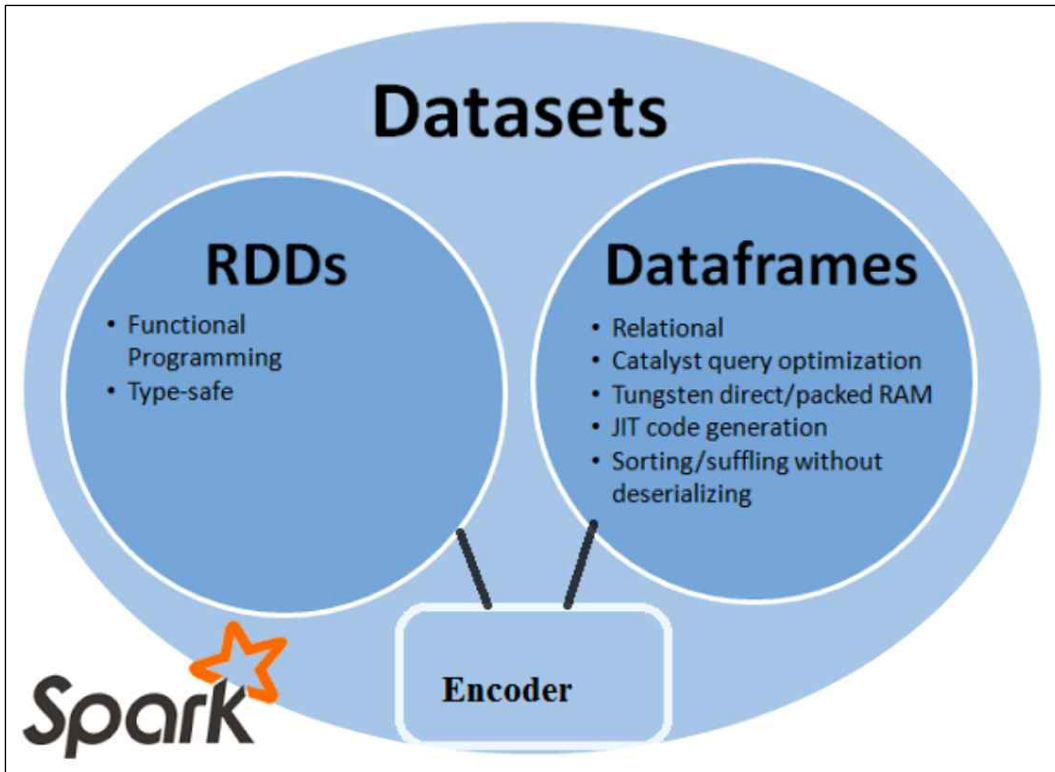
ID ▾	Host	User	Name	Roles	Principal	Registered	Unregistered
------	------	------	------	-------	-----------	------------	--------------



- 기존 데이터 분석을 위해서는 여러 플랫폼이 필요함
- 스파크는 이를 통합하여 공식 지원
- Spark SQL: SQL
- Spark Streaming: 단일/분산 시스템에서 배치/스트리밍 처리
- Spark MLlib: 머신러닝
- Spark GraphX: 그래프 프로세싱

## ■ Spark API

- Spark API는 스파크가 제공하는 Data Interface
- Spark에서 처리/분석 할 데이터의 객체 형태로, 여러 가지 Operation을 할 수 있음
- RDD -> Dataframe -> Dataset 순으로 업데이트



### ① RDD

- Spark 1.0부터 있던 API
- Lamda를 이용한 transformations와 actions
- transformations: map(), filter(), reduce
- actions: collect(), saveAsObjectFile()
- compile-time type safety(자료형에 대한 안정성) 보장하지만 이해하기 어려워 접근이 어려움

```
JavaPairRDD<String, Integer> counts = textFile
    .flatMap(s -> Arrays.asList(s.split(" ")).iterator())
    .mapToPair(word -> new Tuple2<>(word, 1))
    .reduceByKey((a, b) -> a + b);
```



## ② Dataframe

- Spark1.3부터 있던 API
- 데이터에 스키마를 적용하여 RDB처럼 Column과 Row형식의 Table로 표현
- 높은 레벨의 추상화가 가능하여 누구나 쉽게 적응 가능
- Catalyst optimization을 이용하여 훨씬 빠르게 처리
- Compile-time에서 type mismatch같은 에러를 잡지 못함

```
DataFrame errors = df.filter(col("line").like("%ERROR%"));  
// Counts all the errors  
errors.count();  
// Counts errors mentioning MySQL  
errors.filter(col("line").like("%MySQL%")).count();  
// Fetches the MySQL errors as an array of strings  
errors.filter(col("line").like("%MySQL%")).collect();
```

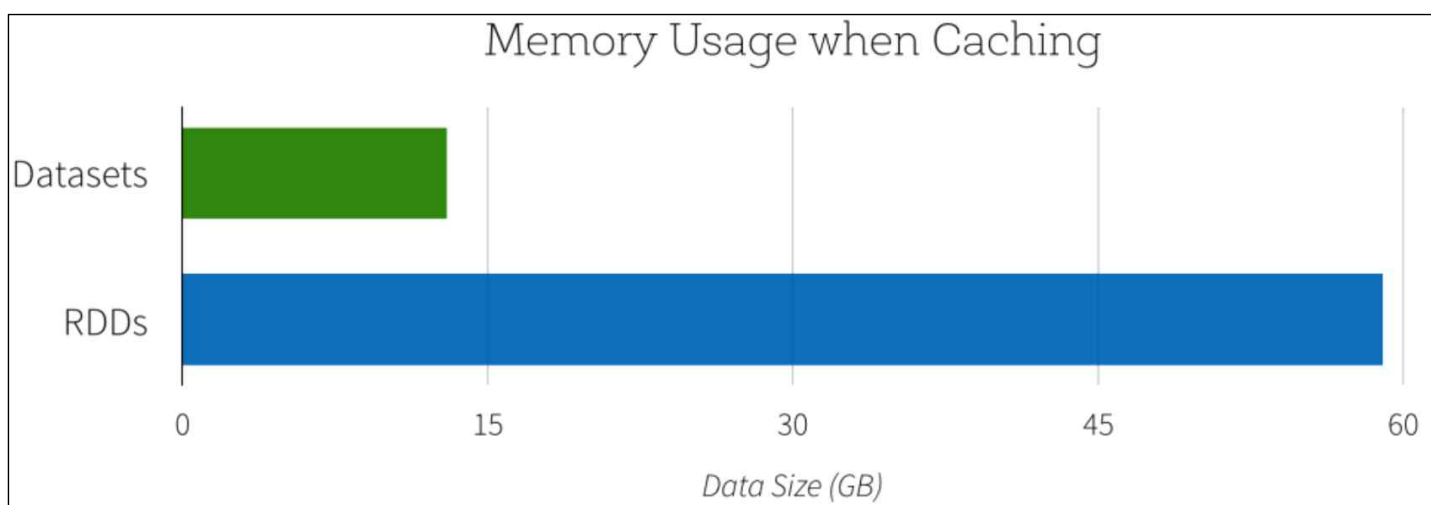
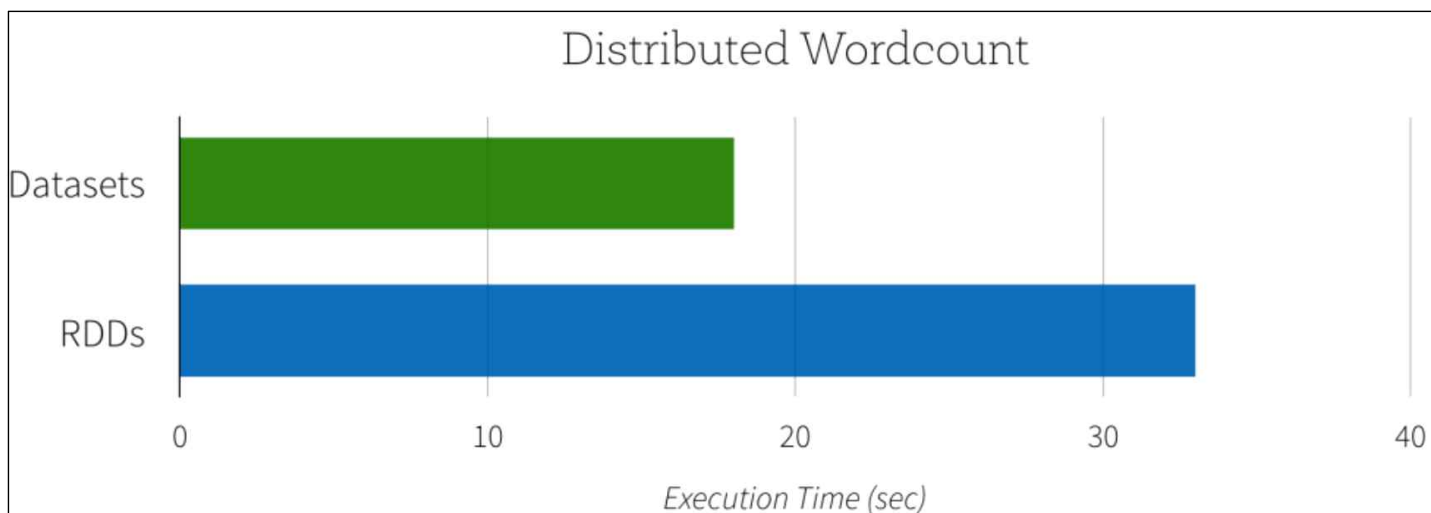
## ③ Dataset

- RDD와 Dataframe의 장점 (Catalyst optimizer와 type-safety보장)을 합치면 어떨까?
- Spark2.0부터 존재하는 API
- Dataframe의 장점인 높은 레벨의 추상화, Catalyst optimizer을 이용하여 성능 향상
- Java Generics를 이용하여 RDD의 장점인 Compile-time type-safety 보장

```
Dataset<Row> ds = spark.read().json("examples/src/main/resources/people.json");  
  
ds.select("name").show();  
// +-----+  
// |   name|  
// +-----+  
// |Michael|  
// |   Andy|  
// |  Justin|  
// +-----+  
  
ds.createOrReplaceTempView("people");  
Dataset<Row> sqlDS = spark.sql("SELECT * FROM people");  
sqlDS.show();  
// +-----+-----+  
// |  age|   name|  
// +-----+-----+  
// |null|Michael|  
// |  30|   Andy|  
// |  19|  Justin|  
// +-----+-----+
```

- Spark API 비교

	RDD	Dataframe	Dataset
성능	느림	빠름	빠름
메모리 관리	누수 존재	최적화	최적화
Type-safety	보장	보장X	보장
확장성	유연함	제한적	유연함





## ■ Spark 설치(standalone)

- Spark 다운로드 페이지에 접속해서 **Choose a Spark release:**에서 현재 최신 버전인 **2.3.1** 선택 후 **Download Spark: spark-2.3.1-bin-hadoop2.7.tgz** 클릭

<https://spark.apache.org/downloads.html>

**Download Apache Spark™**

1. Choose a Spark release: **2.3.1 (Jun 08 2018)**

2. Choose a package type: **Pre-built for Apache Hadoop 2.7 and later**

3. Download Spark: **spark-2.3.1-bin-hadoop2.7.tgz**

4. Verify this release using the [2.3.1 signatures and checksums](#) and [project release KEYS](#).

*Note: Starting version 2.0, Spark is built with Scala 2.11 by default. Scala 2.10 users should download the Spark source package and build with Scala 2.10 support.*

**Link with Spark**

Spark artifacts are hosted in [Maven Central](#). You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark
artifactId: spark-core_2.11
version: 2.3.1
```

**Installing with PyPi**

PySpark is now available in pypi. To install just run `pip install pyspark`.

**Spark Source Code Management**

If you are interested in working with the newest under-development code or contributing to Apache Spark development, you can also check out the master branch from Git:

```
# Master development branch
git clone git://github.com/apache/spark.git
```

**Latest News**

- Spark 2.3.1 released (Jun 08, 2018)
- Spark+AI Summit (June 4-6th, 2018, San Francisco) agenda posted (Mar 01, 2018)
- Spark 2.3.0 released (Feb 28, 2018)
- Spark 2.2.1 released (Dec 01, 2017)

[Archive](#)

**APACHECON**  
North America  
September 24-27, 2018  
Montréal, Canada

**Download Spark**

**Built-in Libraries:**

- [SQL and DataFrames](#)
- [Spark Streaming](#)
- [MLlib \(machine learning\)](#)
- [GraphX \(graph\)](#)

[Third-Party Projects](#)

- **미러사이트에서**

**<http://mirror.navercorp.com/apache/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz> 클릭 후 파일 다운받기**

Home » Dyn About Projects People Get Involved Download Support Apache

We suggest the following mirror site for your download:  
<http://mirror.navercorp.com/apache/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>  
 Other mirror sites are suggested below.  
 It is essential that you verify the integrity of the downloaded file using the PGP signature ( .asc file) or a hash ( .md5 or .sha\* file).  
 Please only use the backup mirrors to download KEYS, PGP and MD5 sigs/hashes or if no other mirrors are working.

**HTTP**

<http://apache.mirror.cdnetworks.com/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>  
<http://apache.tt.co.kr/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>  
<http://mirror.apache-kr.org/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>  
<http://mirror.navercorp.com/apache/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>

- 다운받은 스파크 파일을 **C:\jbm** 디렉터리에 압축을 풀기



- 아래의 주소로 들어가 **winutils** 파일 다운

○ winutils: 하둡 없이 스파크를 실행할 수 있게 해줌

<https://github.com/steveloughran/winutils>

## - hadoop-2.8.1 클릭

steveloughran / winutils

Watch 93 Star 634 Fork 920

<> Code Issues 4 Pull requests 0 Projects 0 Insights

Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Sign up

Dismiss

Windows binaries for Hadoop versions (built from the git commit ID used for the ASF release)

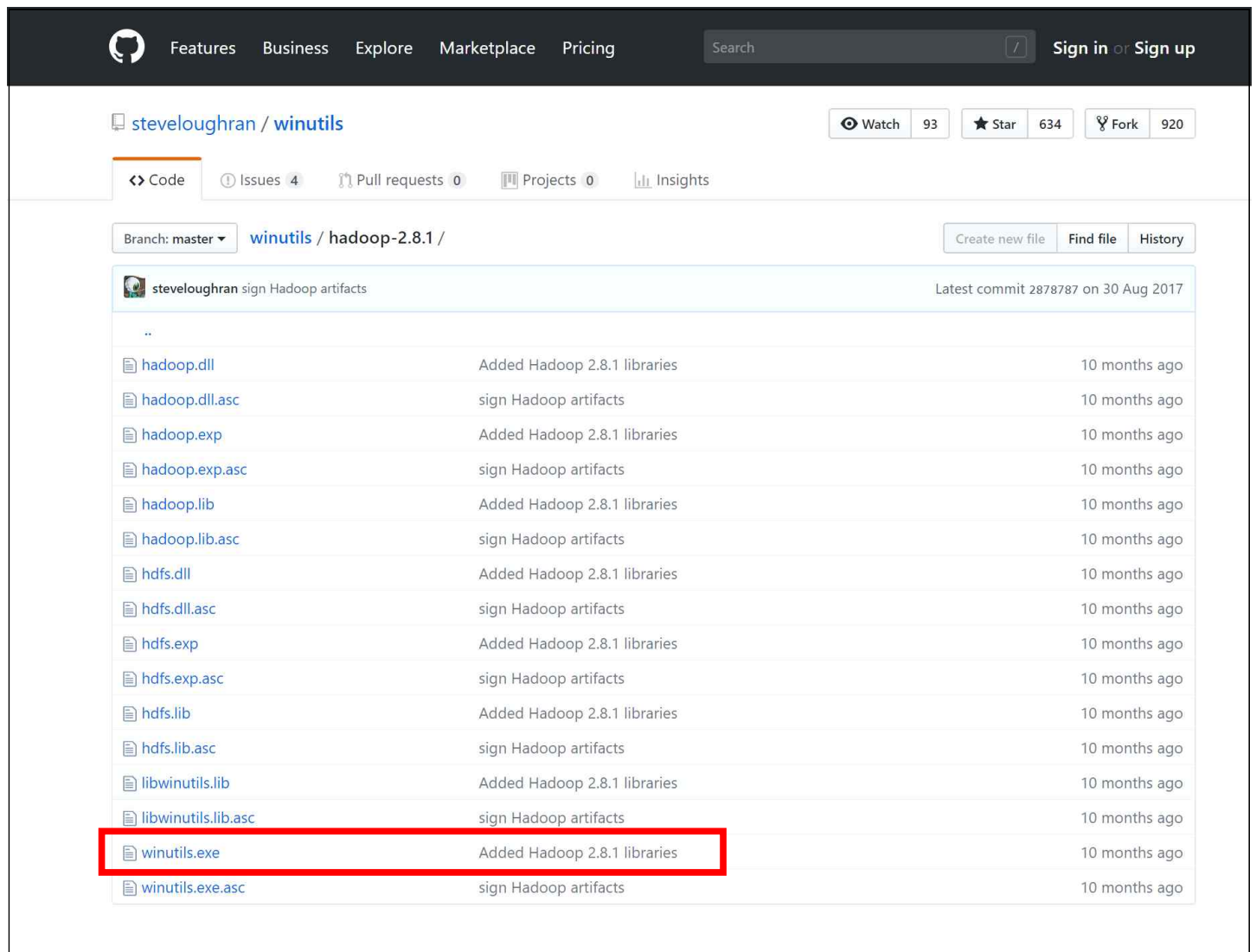
24 commits 1 branch 3 releases 2 contributors Apache-2.0

Branch: master New pull request Find file Clone or download

steveloughran committed on 21 Dec 2017 improving script for adding the artifacts Latest commit 19a39b1 on 21 Dec 2017

hadoop-2.6.0/bin	Add Hadoop-2.6.0/HDP-2.2 windows binaries	3 years ago
hadoop-2.6.3/bin	add gpg2 signatures	3 years ago
hadoop-2.6.4	add 2.6.4 and 2.7.1 windows binaries	2 years ago
hadoop-2.7.1	add 2.6.4 and 2.7.1 windows binaries	2 years ago
hadoop-2.8.0-RC3/bin	sign Hadoop artifacts	a year ago
<b>hadoop-2.8.1</b>	sign Hadoop artifacts	10 months ago
hadoop-2.8.3/bin	Windows binaries for hadoop-2.8.3	6 months ago
hadoop-3.0.0/bin	Hadoop 3.0.0 windows binaries; off the release 3.0 tag, patched with ...	6 months ago
.gitattributes	add gitattributes to try and keep line endings on the BAT files valid	a year ago
.gitignore	add 2.6.4 and 2.7.1 windows binaries	2 years ago

## - winutils.exe 클릭



steveloughran / winutils

Watch 93 Star 634 Fork 920

Code Issues 4 Pull requests 0 Projects 0 Insights

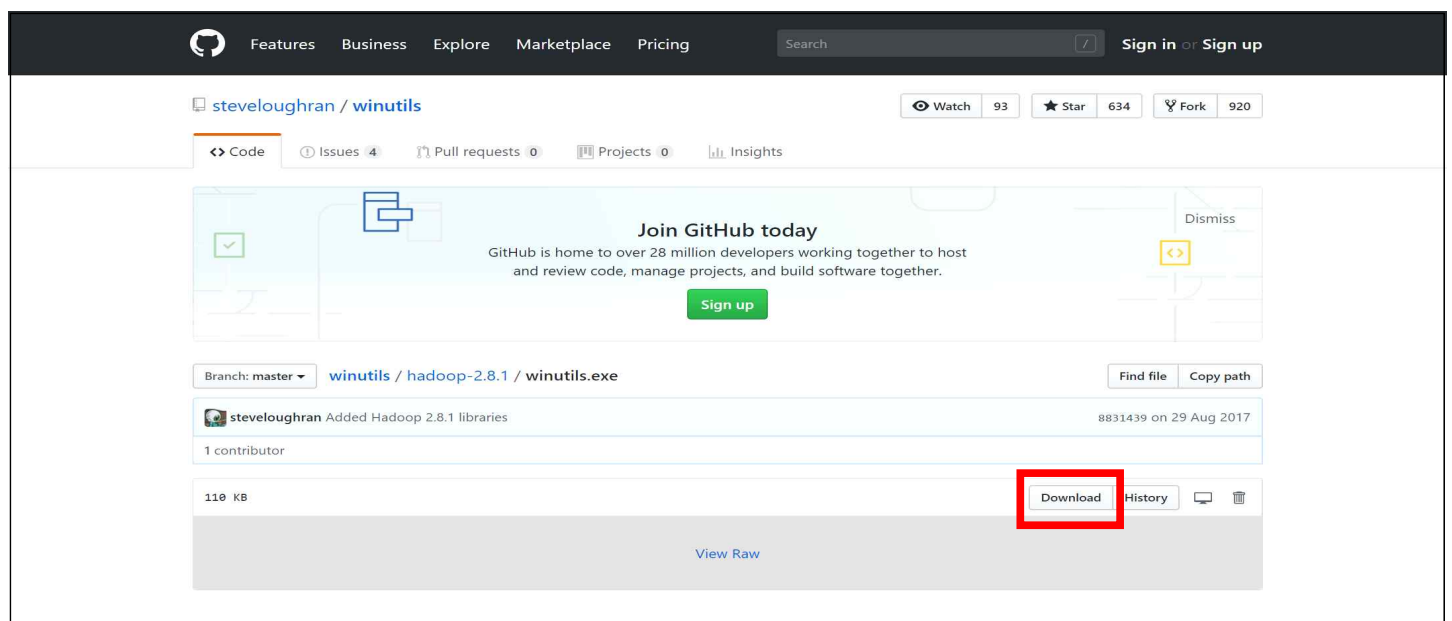
Branch: master winutils / hadoop-2.8.1 /

Create new file Find file History

steveloughran sign Hadoop artifacts Latest commit 2878787 on 30 Aug 2017

hadoop.dll	Added Hadoop 2.8.1 libraries	10 months ago
hadoop.dll.asc	sign Hadoop artifacts	10 months ago
hadoop.exp	Added Hadoop 2.8.1 libraries	10 months ago
hadoop.exp.asc	sign Hadoop artifacts	10 months ago
hadoop.lib	Added Hadoop 2.8.1 libraries	10 months ago
hadoop.lib.asc	sign Hadoop artifacts	10 months ago
hdfs.dll	Added Hadoop 2.8.1 libraries	10 months ago
hdfs.dll.asc	sign Hadoop artifacts	10 months ago
hdfs.exp	Added Hadoop 2.8.1 libraries	10 months ago
hdfs.exp.asc	sign Hadoop artifacts	10 months ago
hdfs.lib	Added Hadoop 2.8.1 libraries	10 months ago
hdfs.lib.asc	sign Hadoop artifacts	10 months ago
libwinutils.lib	Added Hadoop 2.8.1 libraries	10 months ago
libwinutils.lib.asc	sign Hadoop artifacts	10 months ago
winutils.exe	Added Hadoop 2.8.1 libraries	10 months ago
winutils.exe.asc	sign Hadoop artifacts	10 months ago

## - Download 클릭



steveloughran / winutils

Watch 93 Star 634 Fork 920

Code Issues 4 Pull requests 0 Projects 0 Insights

Branch: master winutils / hadoop-2.8.1 / winutils.exe

Find file Copy path

steveloughran Added Hadoop 2.8.1 libraries 8831439 on 29 Aug 2017

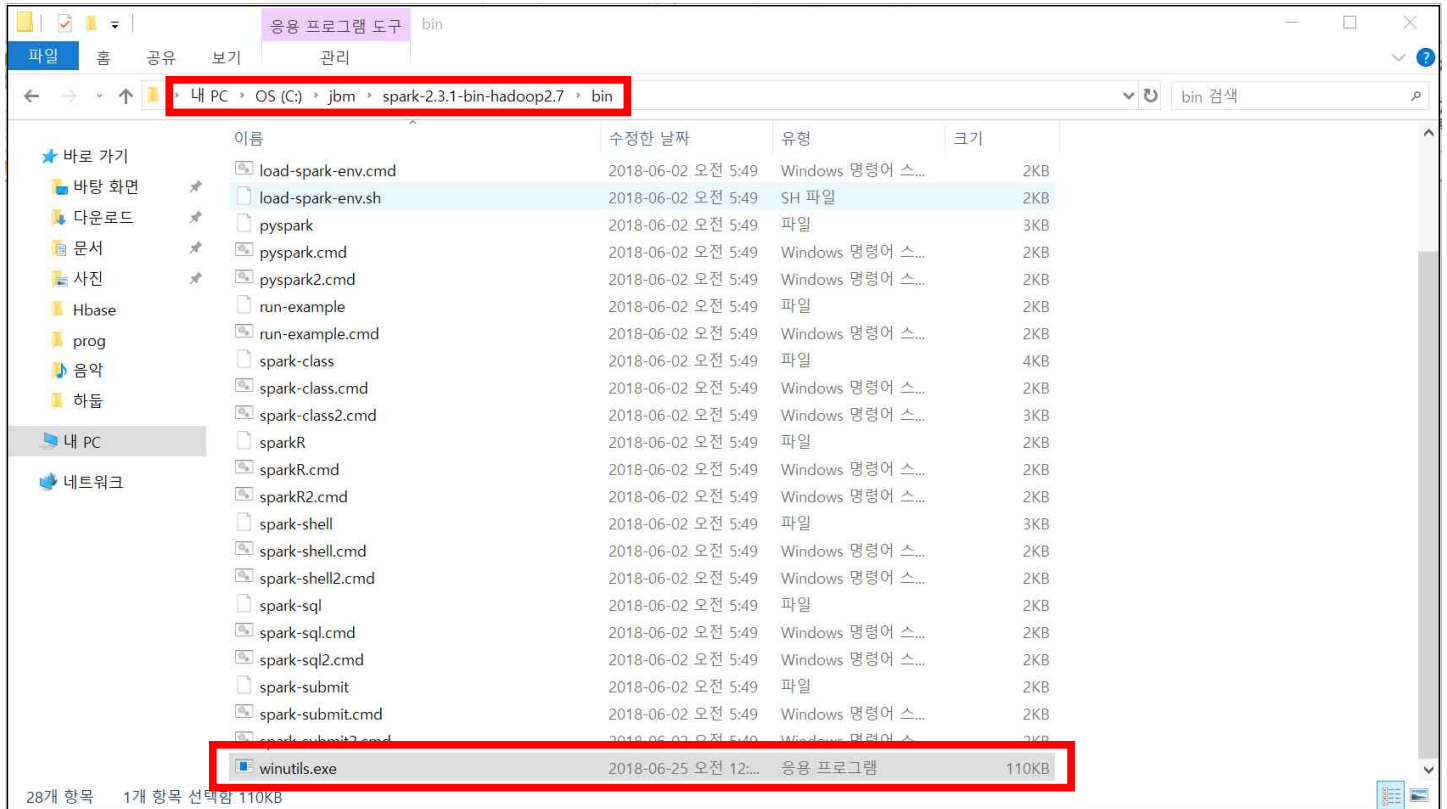
1 contributor

110 KB

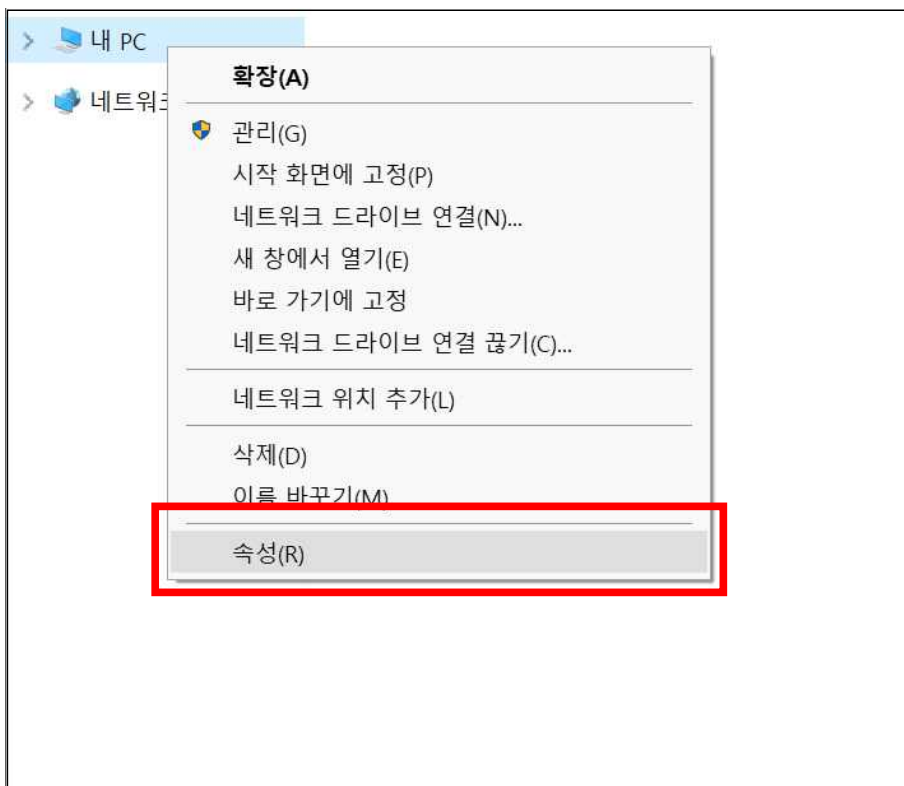
Download History

View Raw

- C:\jbm\spark-2.3.1-bin-hadoop2.7\bin 경로에 다운 받은 winutils.exe를 갖다 놓음



- 내 PC를 마우스 오른쪽 버튼으로 클릭해 속성 클릭



- 고급 시스템 설정 클릭



- 환경변수 클릭



- 새로 만들기 클릭 후 아래와 같이 변수 이름, 변수 값 입력
- 확인 클릭

변수 이름: SPARK\_HOME

변수 값: C:\jbm\spark-2.3.1-bin-hadoop2.7

환경 변수

ppsp7에 대한 사용자 변수(U)

변수	값
새 시스템 변수	
변수 이름(N):	SPARK_HOME
변수 값(V):	C:\jbm\spark-2.3.1-bin-hadoop2.7
<input type="button" value="디렉터리 찾아보기(D)..."/> <input type="button" value="파일 찾아보기(F)..."/> <input type="button" value="확인"/> <input type="button" value="취소"/>	

시스템 변수(S)

변수	값
OS	Windows_NT
Path	C:\oracle\exe\app\oracle\product\11.2.0\server\bin;%JAVA_HOME%\bin;C:\Program File...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARC...	AMD64
PROCESSOR_DEN...	Intel64 Family 6 Model 158 Stepping 10, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVI...	9e0a
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShe...
TEMP	C:\Windows\TEMP
TMP	C:\Windows\TEMP
USERNAME	SYSTEM



- 새로 만들기 클릭 후 아래와 같이 변수 이름, 변수 값 입력
- 확인 클릭

변수 이름: HADOOP\_HOME

변수 값: C:\jbm\spark-2.3.1-bin-hadoop2.7

환경 변수

ppsp7에 대한 사용자 변수(U)

변수	값
새 시스템 변수	
변수 이름(N):	HADOOP_HOME
변수 값(V):	C:\jbm\spark-2.3.1-bin-hadoop2.7
<input type="button" value="디렉터리 찾아보기(D)..."/> <input type="button" value="파일 찾아보기(F)..."/> <input type="button" value="확인"/> <input type="button" value="취소"/>	

시스템 변수(S)

변수	값
NUMBER_OF_PRO...	12
OS	Windows_NT
Path	C:\oracle\app\oracle\product\11.2.0\server\bin; %JAVA_HOME%\bin; C:\Program File...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARC...	AMD64
PROCESSOR_IDEN...	Intel64 Family 6 Model 158 Stepping 10, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVI...	9e0a
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules; C:\Windows\system32\WindowsPowerShe...
TEMP	C:\Windows\TEMP
TMP	C:\Windows\TEMP

- cmd 창을 열어 스파크의 bin 경로로 이동

```
> cd c:\jbm\spark-2.3.1-bin-hadoop2.7\bin
```

- spark-shell 접속해 설치 확인

```
> spark-shell
```

```
명령 프롬프트 - spark-shell
Microsoft Windows [Version 10.0.16299.492]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\wppsp7>cd c:\jbm\spark-2.3.1-bin-hadoop2.7\bin

c:\jbm\spark-2.3.1-bin-hadoop2.7\bin>spark-shell
Missing Python executable 'python', defaulting to 'c:\jbm\spark-2.3.1-bin-hadoop2.7\bin\..' for SPARK_HOME environment variable. Please install Python or specify the correct Python executable in PYSPARK_DRIVER_PYTHON or PYSPARK_PYTHON environment variable to detect SPARK_HOME safely.
2018-06-25 01:53:14 WARN NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://DESKTOP-8HRM11F:4040
Spark context available as 'sc' (master = local[*], app id = local-1529859199358).
Spark session available as 'spark'.
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___\
| |  | | \___/
|_|  |_|

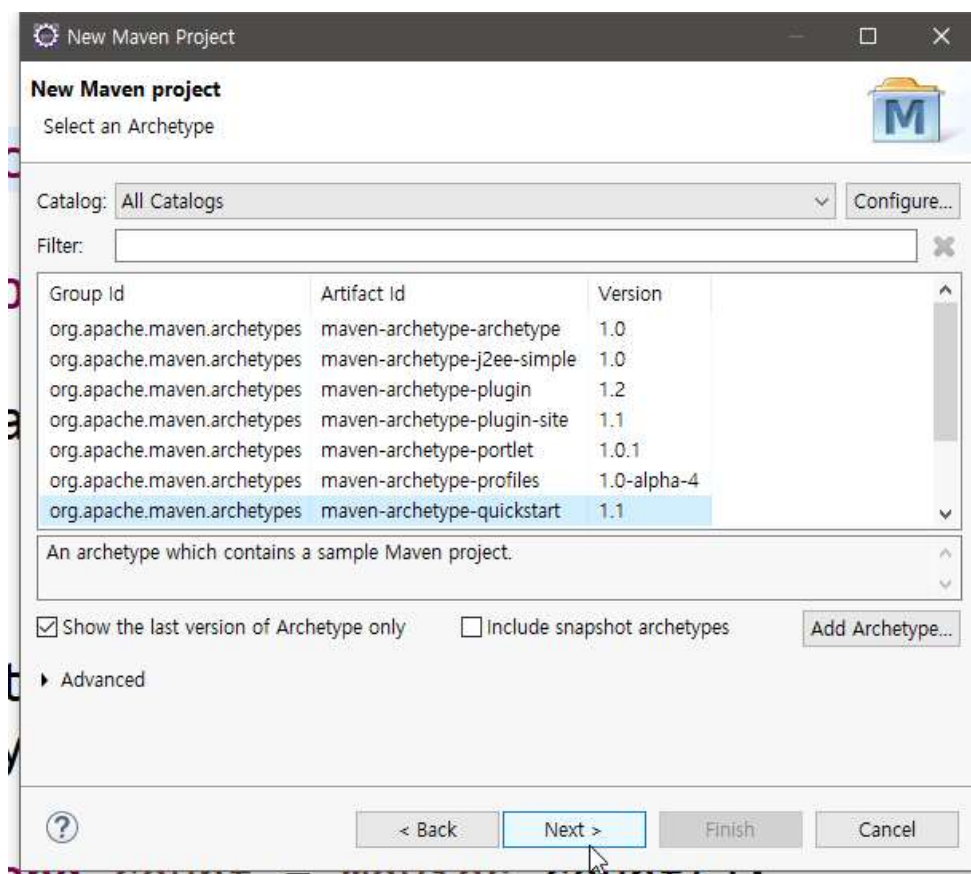
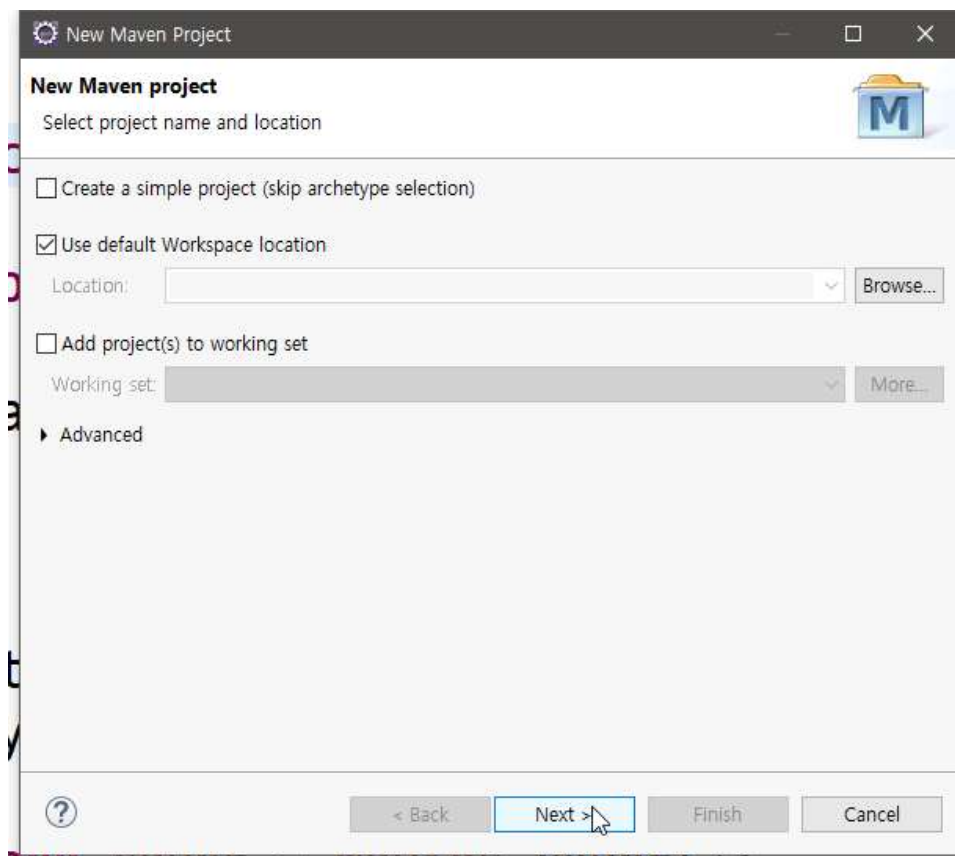
 version 2.3.1

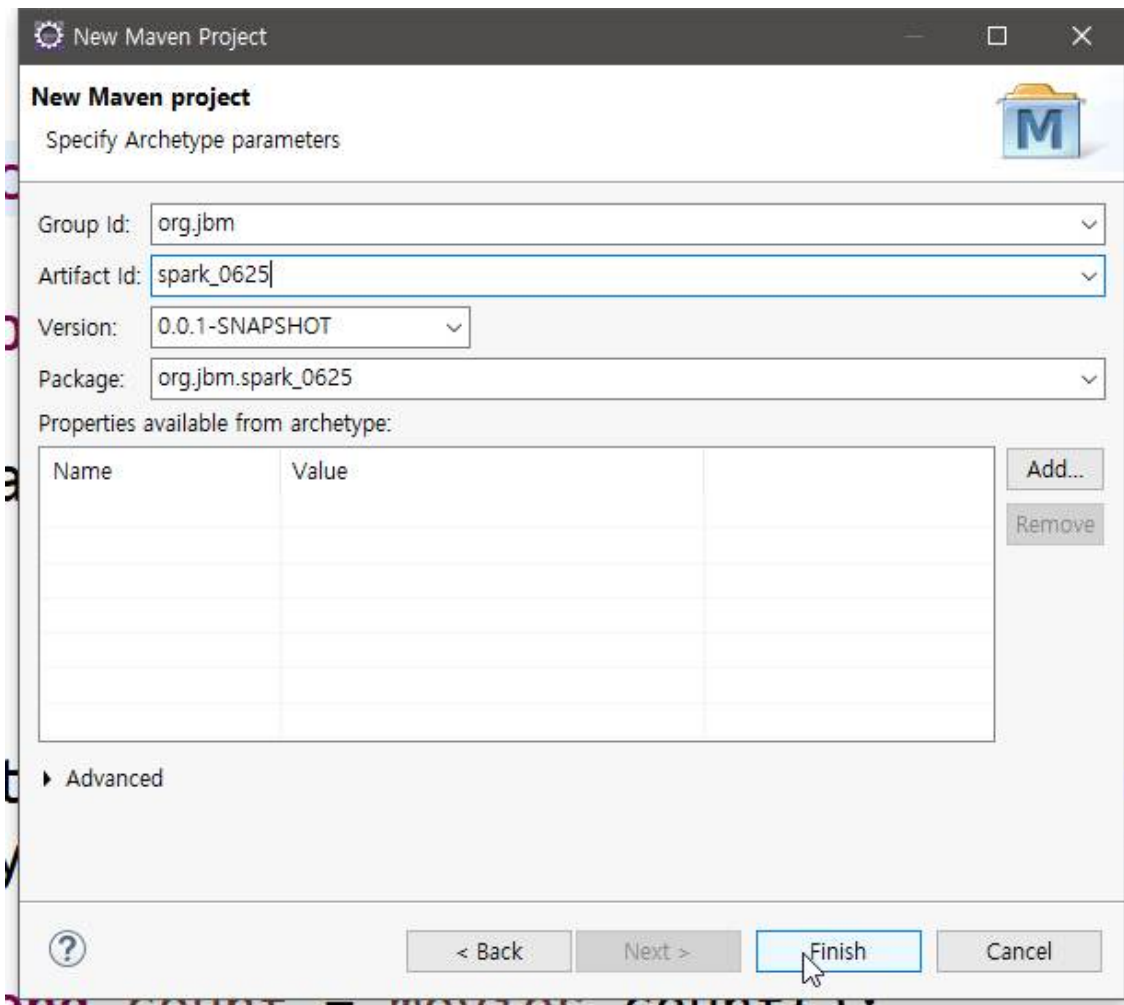
Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_172)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

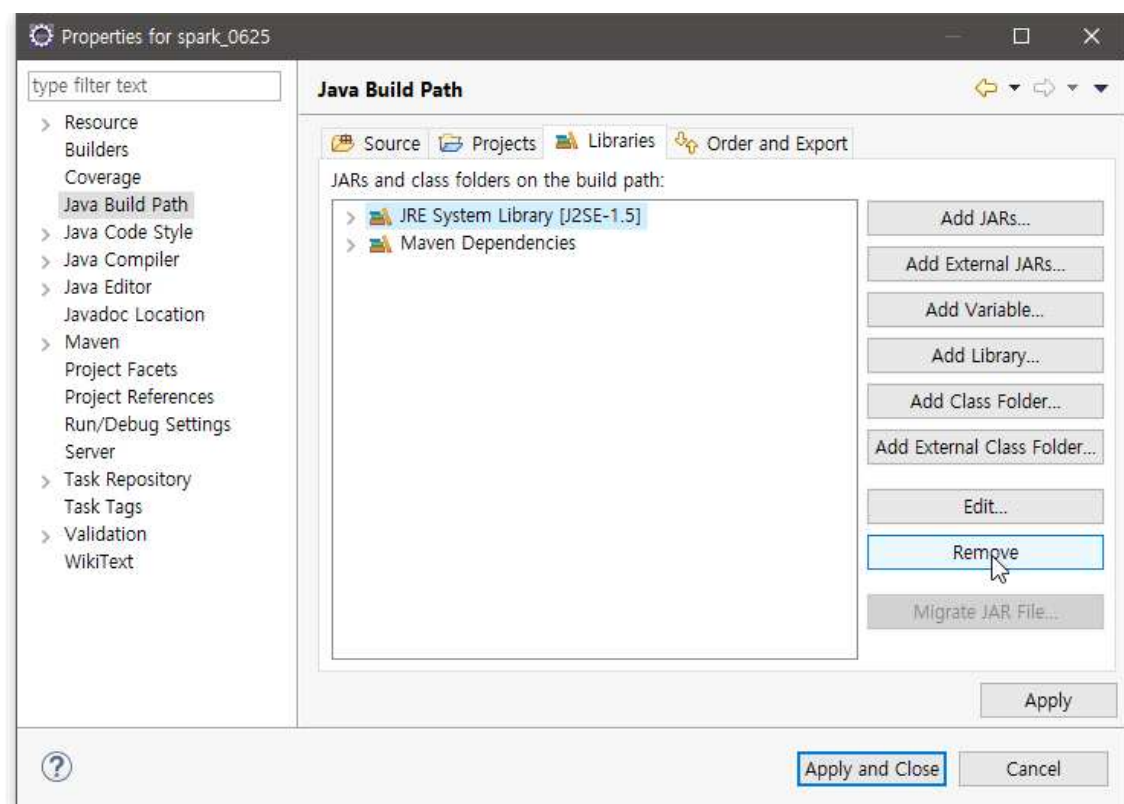
## ■ Spark 일반 Java Application 프로그래밍

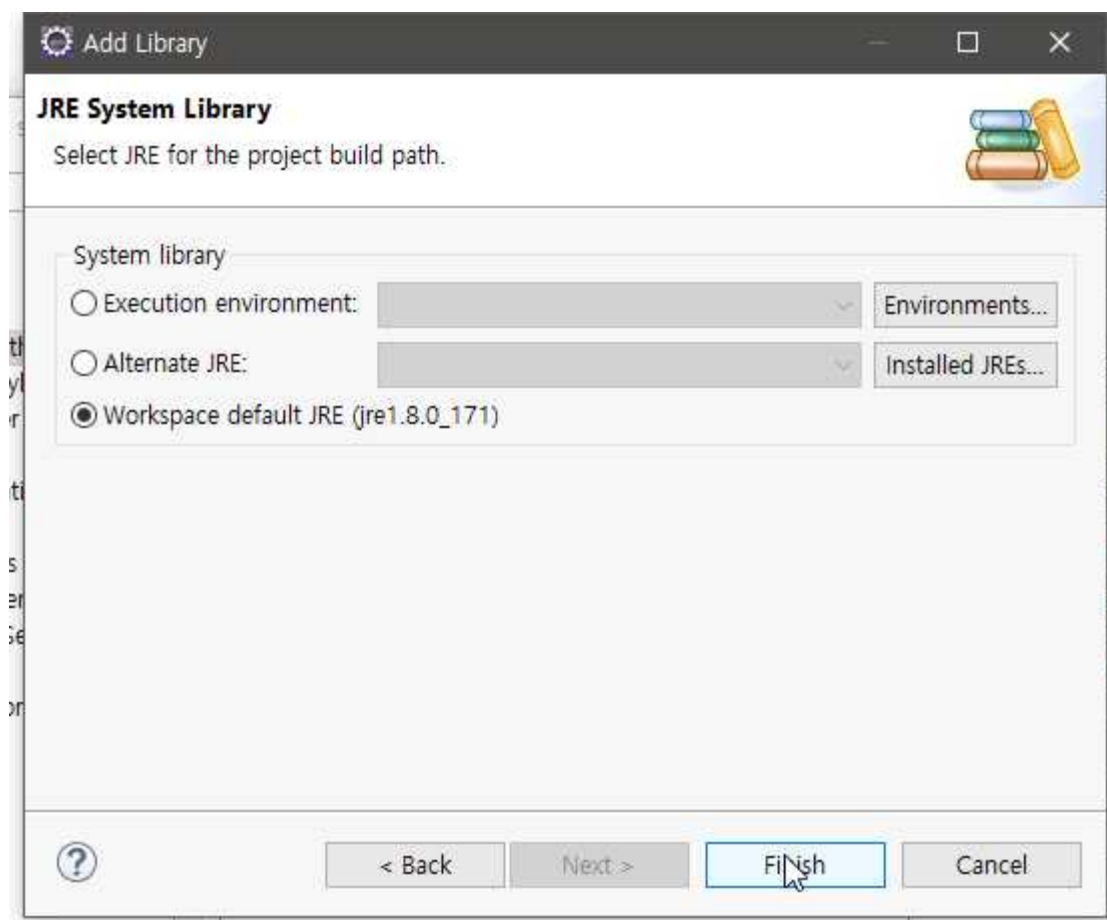
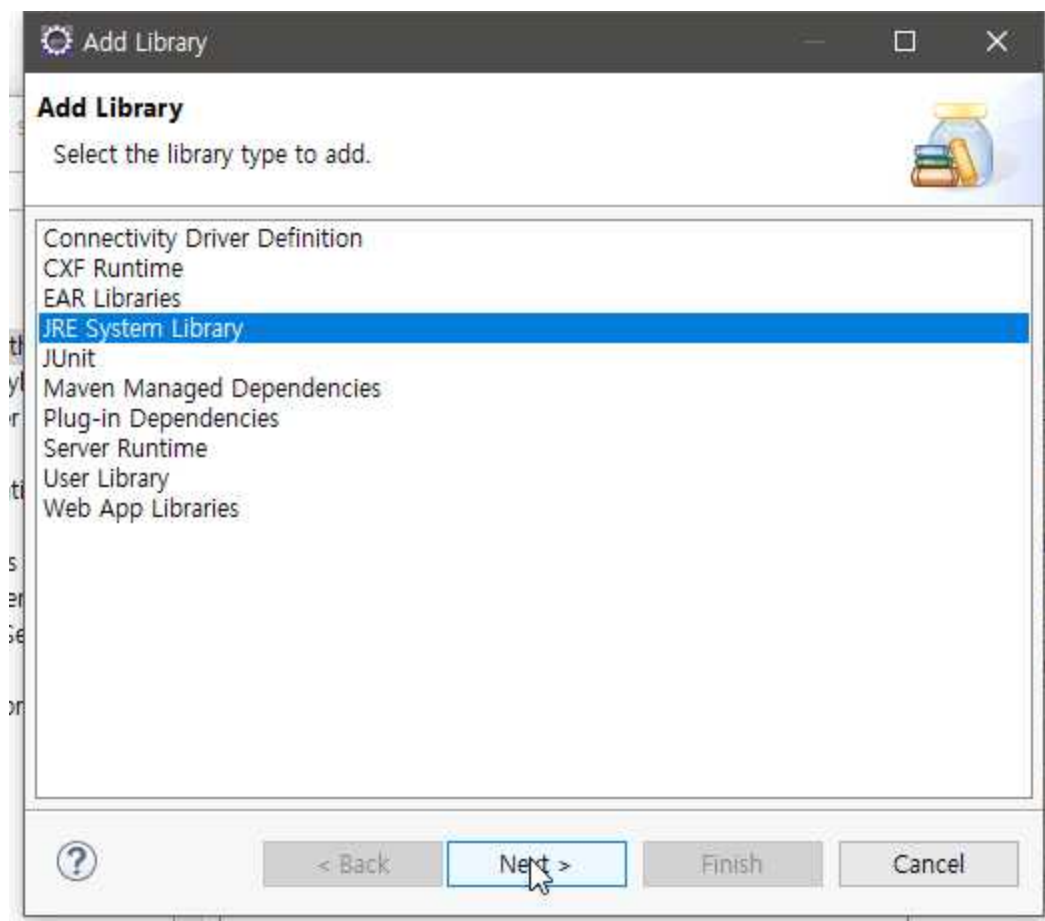
### 1) maven 프로젝트 생성

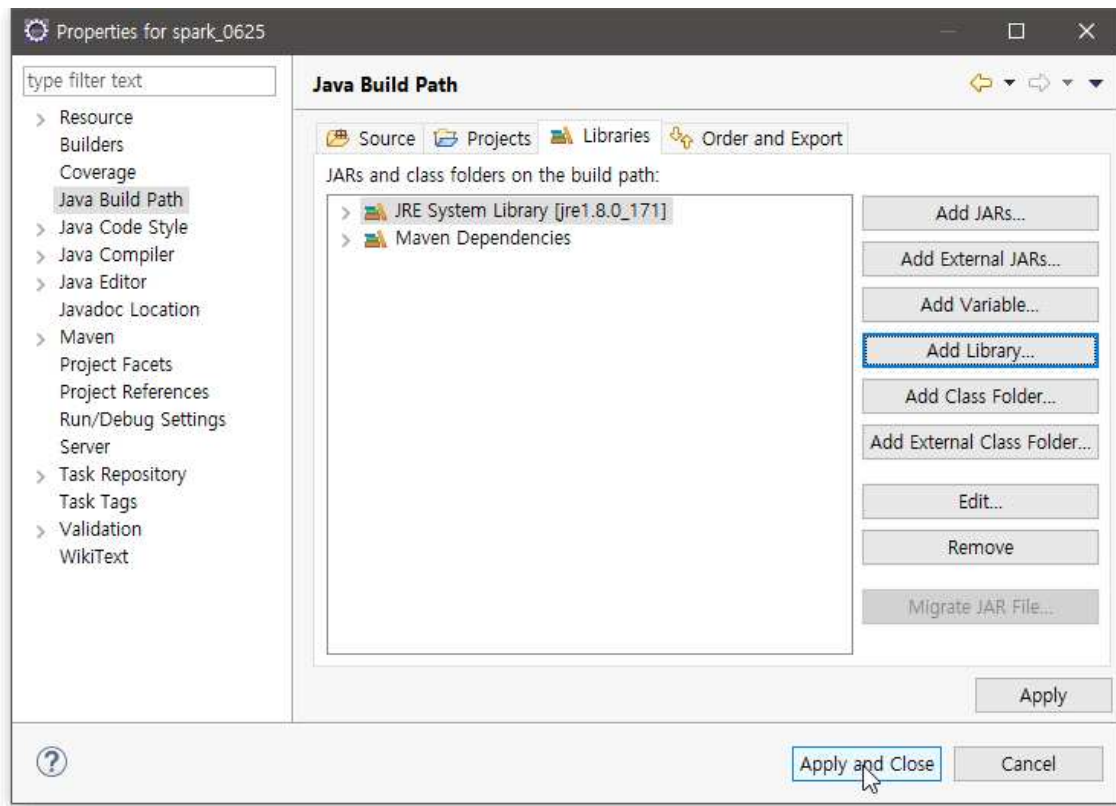




## 2) 자바 8 JRE system library로 변경







3) junit 삭제 및 test 패키지 삭제

4) 필요한 라이브러리 dependency 지정

```
<dependencies>

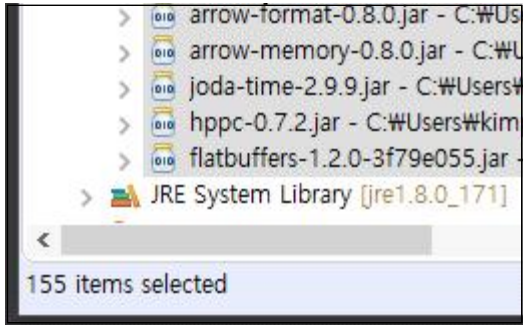
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.11</artifactId>
        <version>2.3.1</version>
    </dependency>

    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-sql_2.11</artifactId>
        <version>2.3.1</version>
        <scope>provided</scope>
    </dependency>

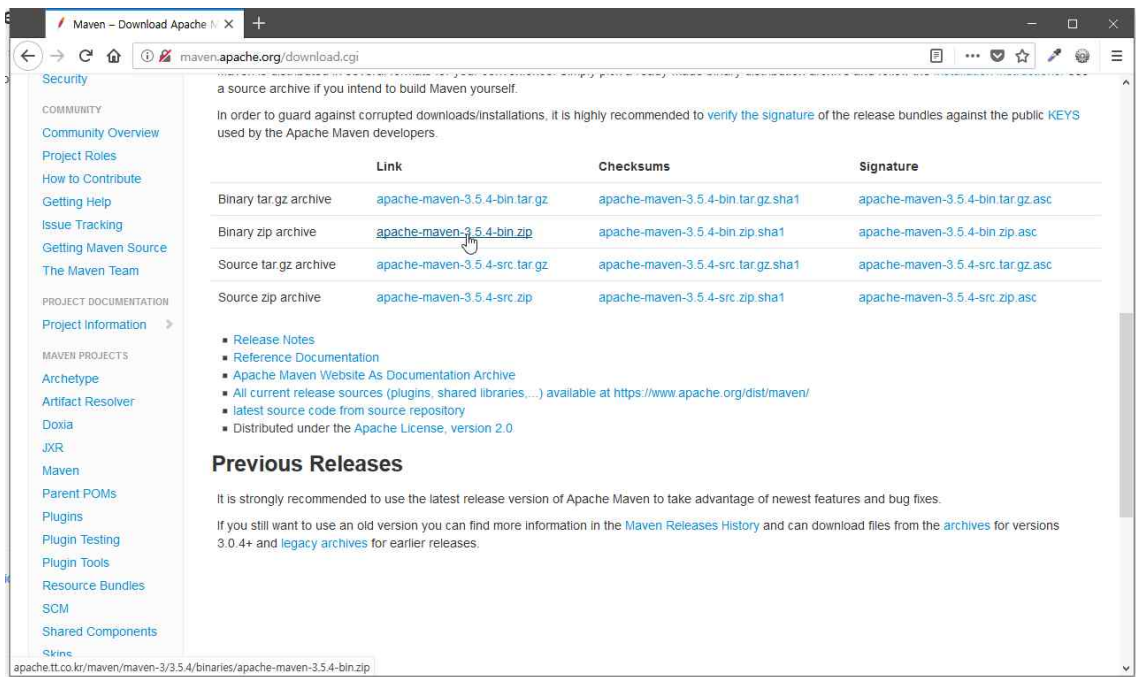
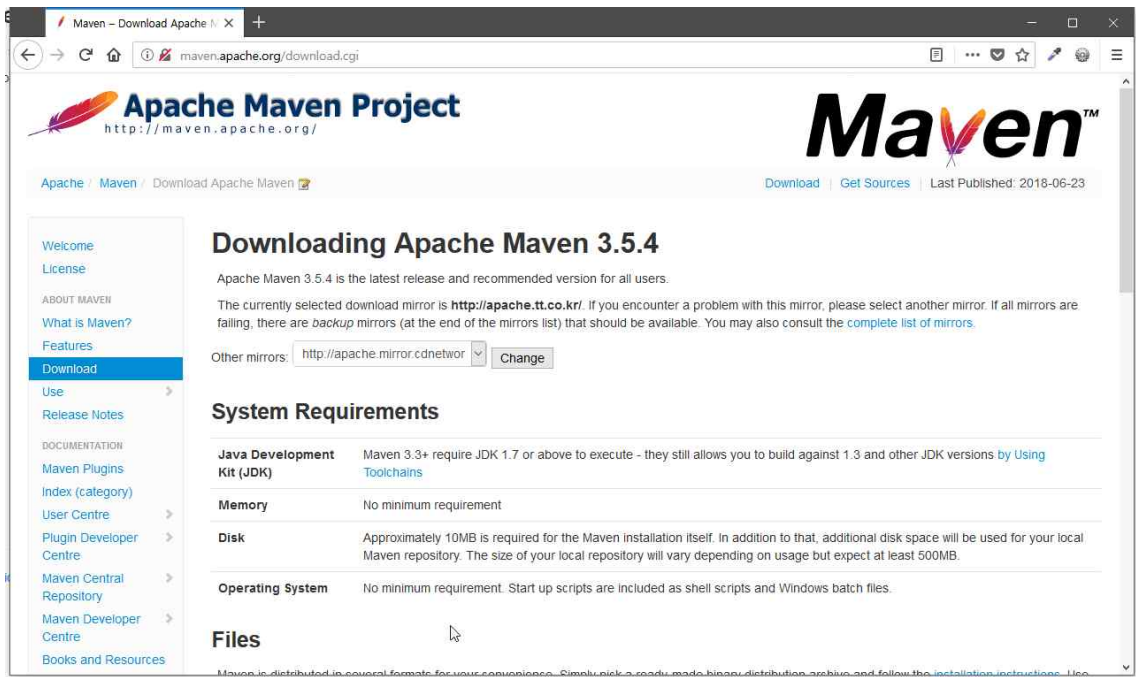
</dependencies>
```



## - 라이브러리 등록(155개)

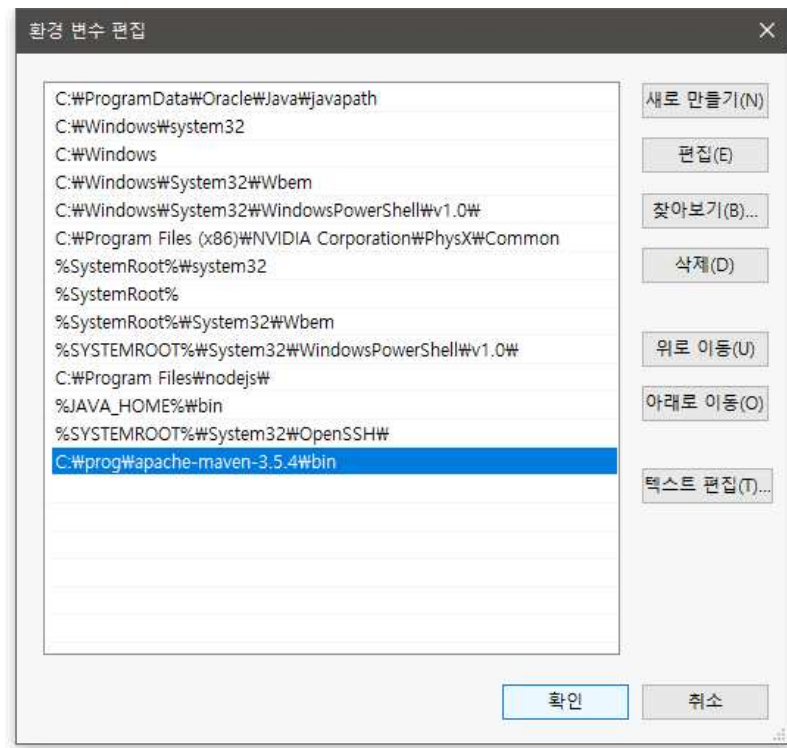


## - 이 jar파일들을 복사하려면 mvn을 설치해야 함

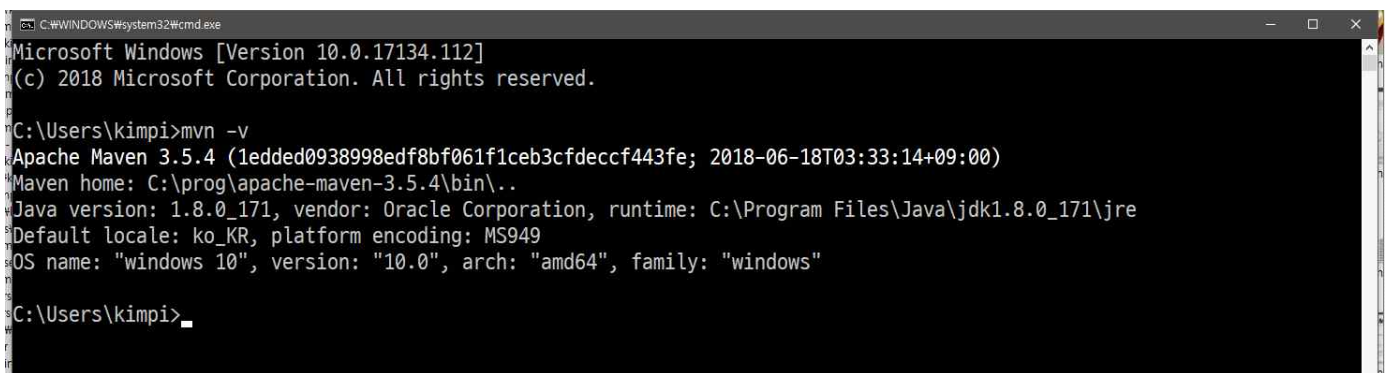




- path 지정(maven 설치 경로/bin)



- mvn 실행 테스트(mvn -v)



- pom.xml이 있는 폴더(프로젝트 폴더)에서

```
C:\jbm\workspace\spark\spark_0625>mvn dependency:copy-dependencies -DoutputDirectory=c:/jbm/lib
```

```

C:\jbm\workspace\spark\spark_0625>mvn dependency:copy-dependencies -DoutputDirectory=c:\jbm\lib
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-in
stall-plugin-2.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-in
stall-plugin-2.4.pom (6.4 kB at 5.1 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/2.7/maven-de
ploy-plugin-2.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/2.7/maven-dep
loy-plugin-2.7.pom (5.6 kB at 18 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.3/maven-site
-plugin-3.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.3/maven-site
-plugin-3.3.pom (21 kB at 47 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/1.3/maven-an
trun-plugin-1.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/1.3/maven-an
trun-plugin-1.3.pom (4.7 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/12/maven-plugins-1
2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/12/maven-plugins-12
.pom (12 kB at 37 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/1.3/maven-an
trun-plugin-1.3.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/1.3/maven-an
trun-plugin-1.3.jar (24 kB at 63 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/2.2-beta-5
/maven-assembly-plugin-2.2-beta-5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/2.2-beta-5
/maven-assembly-plugin-2.2-beta-5.pom (15 kB at 48 kB/s)

```

```

[INFO] Copying jersey-media-jaxb-2.22.2.jar to c:\jbm\lib\jersey-media-jaxb-2.22.2.jar
[INFO] Copying spark-unsafe_2.11-2.3.1.jar to c:\jbm\lib\spark-unsafe_2.11-2.3.1.jar
[INFO] Copying unused-1.0.0.jar to c:\jbm\lib\unused-1.0.0.jar
[INFO] Copying jets3t-0.9.4.jar to c:\jbm\lib\jets3t-0.9.4.jar
[INFO] Copying metrics-jvm-3.1.5.jar to c:\jbm\lib\metrics-jvm-3.1.5.jar
[INFO] Copying avro-ipc-1.7.7.jar to c:\jbm\lib\avro-ipc-1.7.7.jar
[INFO] Copying zstd-jni-1.3.2-2.jar to c:\jbm\lib\zstd-jni-1.3.2-2.jar
[INFO] Copying commons-collections-3.2.2.jar to c:\jbm\lib\commons-collections-3.2.2.jar
[INFO] Copying parquet-common-1.8.3.jar to c:\jbm\lib\parquet-common-1.8.3.jar
[INFO] Copying parquet-encoding-1.8.3.jar to c:\jbm\lib\parquet-encoding-1.8.3.jar
[INFO] Copying hk2-locator-2.4.0-b34.jar to c:\jbm\lib\hk2-locator-2.4.0-b34.jar
[INFO] Copying spark-sql_2.11-2.3.1.jar to c:\jbm\lib\spark-sql_2.11-2.3.1.jar
[INFO] Copying hppc-0.7.2.jar to c:\jbm\lib\hppc-0.7.2.jar
[INFO] Copying hadoop-yarn-client-2.6.5.jar to c:\jbm\lib\hadoop-yarn-client-2.6.5.jar
[INFO] Copying pyrolite-4.13.jar to c:\jbm\lib\pyrolite-4.13.jar
[INFO] Copying spark-launcher_2.11-2.3.1.jar to c:\jbm\lib\spark-launcher_2.11-2.3.1.jar
[INFO] Copying java-xmlbuilder-1.1.jar to c:\jbm\lib\java-xmlbuilder-1.1.jar
[INFO] Copying jcl-over-slf4j-1.7.16.jar to c:\jbm\lib\jcl-over-slf4j-1.7.16.jar
[INFO] Copying janino-3.0.8.jar to c:\jbm\lib\janino-3.0.8.jar
[INFO] Copying parquet-hadoop-1.8.3.jar to c:\jbm\lib\parquet-hadoop-1.8.3.jar
[INFO] Copying netty-all-4.1.17.Final.jar to c:\jbm\lib\netty-all-4.1.17.Final.jar
[INFO] Copying commons-httpclient-3.1.jar to c:\jbm\lib\commons-httpclient-3.1.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 13.113 s
[INFO] Finished at: 2018-06-25T07:55:38+09:00
[INFO] -----
C:\jbm\workspace\spark\spark_0625>

```

이름	수정함 날짜	유형	크기
activation-1.1.1.jar	2018-06-25 오전...	Executable Jar File	68KB
aircompressor-0.8.jar	2018-06-25 오전...	Executable Jar File	128KB
antlr4-runtime-4.7.jar	2018-06-25 오전...	Executable Jar File	327KB
aopalliance-repackaged-2.4.0-b34.jar	2018-06-25 오전...	Executable Jar File	15KB
apacheds-18n-2.0.0-M15.jar	2018-06-25 오전...	Executable Jar File	44KB
apacheds-kerberos-codec-2.0.0-M15.jar	2018-06-25 오전...	Executable Jar File	676KB
api-asn1-api-1.0.0-M20.jar	2018-06-25 오전...	Executable Jar File	17KB
api-util-1.0.0-M20.jar	2018-06-25 오전...	Executable Jar File	79KB
arrow-format-0.8.0.jar	2018-06-25 오전...	Executable Jar File	51KB
arrow-memory-0.8.0.jar	2018-06-25 오전...	Executable Jar File	78KB
arrow-vector-0.8.0.jar	2018-06-25 오전...	Executable Jar File	1,241KB
avro-1.7.7.jar	2018-06-25 오전...	Executable Jar File	427KB
avro-ipc-1.7.7.jar	2018-06-25 오전...	Executable Jar File	189KB
avro-ipc-1.7.7-tests.jar	2018-06-25 오전...	Executable Jar File	339KB
avro-mapred-1.7.7-hadoop2.jar	2018-06-25 오전...	Executable Jar File	177KB

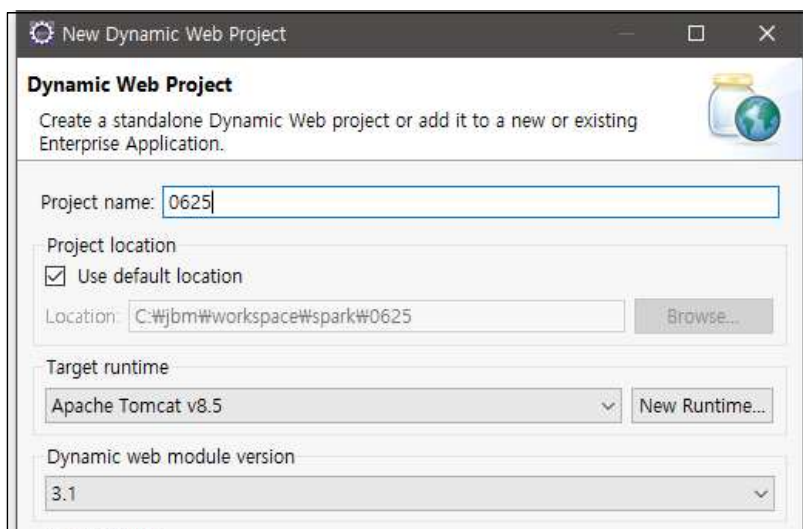
## - Spark가 가진 RDD / DataFrame / DataSet중 RDD 기본 예제 수행

```
public class SparkApp {  
  
    public static void main(String[] args) {  
        SparkConf conf = new SparkConf().setAppName("SparkTest").setMaster("local");  
        JavaSparkContext jsc = new JavaSparkContext(conf);  
        JavaRDD<String> ratingData =  
        jsc.textFile("hdfs://192.168.56.101:8020/user/hive/warehouse/chickens").cache();  
        // JavaRDD<String> ratingData = jsc.textFile("src/assets/").cache();  
        System.out.println("치킨 시킨 횟수 : "+ratingData.count());  
    }  
}
```

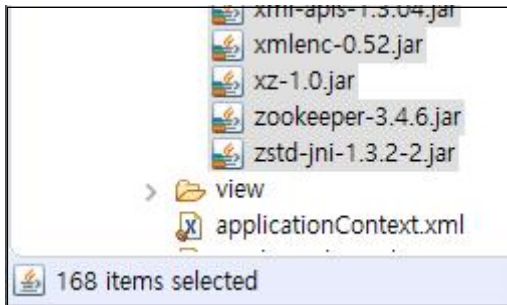
```
18/06/25 08:09:46 INFO MemoryStore: Block rdd_1_1 stored as values in memory (estimated size 4.0 MB, free 4.1 GB)  
18/06/25 08:09:46 INFO BlockManagerInfo: Added rdd_1_1 in memory on DESKTOP-MUP3PD0:8211 (size: 4.0 MB, free: 4.1 GB)  
18/06/25 08:09:46 INFO Executor: Finished task 1.0 in stage 0.0 (TID 1). 832 bytes result sent to driver  
18/06/25 08:09:46 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 118 ms on localhost (executor driver) (2/2)  
18/06/25 08:09:47 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool  
18/06/25 08:09:47 INFO DAGScheduler: ResultStage 0 (count at SparkApp.java:18) finished in 0.637 s  
18/06/25 08:09:47 INFO DAGScheduler: Job 0 finished: count at SparkApp.java:18, took 0.675487 s  
치킨 시킨 횟수 : 71149  
18/06/25 08:09:47 INFO SparkContext: Invoking stop() from shutdown hook  
18/06/25 08:09:47 INFO SparkUI: Stopped Spark web UI at http://DESKTOP-MUP3PD0:4040  
18/06/25 08:09:47 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!  
18/06/25 08:09:47 INFO MemoryStore: MemoryStore cleared  
18/06/25 08:09:47 INFO BlockManager: BlockManager stopped  
18/06/25 08:09:47 INFO BlockManagerMaster: BlockManagerMaster stopped  
18/06/25 08:09:47 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!  
18/06/25 08:09:47 INFO SparkContext: Successfully stopped SparkContext  
18/06/25 08:09:47 INFO ShutdownHookManager: Shutdown hook called  
18/06/25 08:09:47 INFO ShutdownHookManager: Deleting directory C:\Users\kimpi\AppData\Local\Temp\spark-83b55e64-fe7a-48f5-9542-e79f4d270de8
```

## ■ Spark Spring 연동 프로그래밍

### 1) Dynamic Web Project로 생성



## 2) lib에 spring + spark 라이브러리 전부 복사



## 3) 기본 Spring 설정은 변함 없음

### - web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
    <filter>
        <filter-name>encoding</filter-name>
        <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
        <init-param>
            <param-name>encoding</param-name>
            <param-value>UTF-8</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>encoding</filter-name>
        <url-pattern>*</url-pattern>
    </filter-mapping>
    <filter>
        <filter-name>httpMethodFilter</filter-name>
        <filter-class>org.springframework.web.filter.HiddenHttpMethodFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>httpMethodFilter</filter-name>
        <url-pattern>*</url-pattern>
    </filter-mapping>
    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>
    <servlet>
        <servlet-name>spark</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
```

```
<servlet-name>spark</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
```

## - applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:jee="http://www.springframework.org/schema/jee"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee-4.3.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd">

<context:annotation-config/>

<bean class="com.jbm.spark.configuration.WebApplicationConfig"/>

</beans>
```

## - spark-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:websocket="http://www.springframework.org/schema/websocket"
       xsi:schemaLocation="http://www.springframework.org/schema/websocket
http://www.springframework.org/schema/websocket/spring-websocket.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<mvc:annotation-driven/>

<!-- resource -->
<mvc:resources location="/css/" mapping="/css/**"/>
<mvc:resources location="/img/" mapping="/img/**"/>
```



```

<mvc:resources location="/upload/" mapping="/upload/**"/>
<mvc:resources location="/profile/" mapping="/profile/**"/>
<mvc:resources location="/poster/" mapping="/poster/**"/>
<mvc:resources location="/js/" mapping="/js/**"/>

<!-- /WEB-INF/view/와 .jsp가 반복 -->
<mvc:view-resolvers>
    <mvc:jsp
        prefix="/WEB-INF/view/"
        suffix=".jsp"/>
</mvc:view-resolvers>

<bean id="sparkService" p:sparkSession-ref="sparkSession" class="com.jbm.spark.util.SparkService"/>

<bean
p:sparkService-ref="sparkService"
class="com.jbm.spark.controller.AjaxController"/>

<bean p:test-ref="test"
class="com.jbm.spark.controller.IndexController"/>

</beans>

```

- RDD나 DataSet(DataFrame)을 사용하려면 SparkConf나 SparkSession이 필요함
- SparkConf나 SparkSession을 만들면 Spark Master가 실행됨

The screenshot shows the Spark Jobs UI in a web browser. The address bar indicates the URL is localhost:4040/jobs/. The page title is "test - Spark Jobs". The Spark logo and version 2.3.1 are visible. The "Jobs" tab is selected. The main content area shows "Spark Jobs (?)". Below this, it displays "User: kimp1", "Total Uptime: 1.0 min", "Scheduling Mode: FIFO", and "Completed Jobs: 1". A link for "Event Timeline" is present. Under "Completed Jobs (1)", a table lists the job details.

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	count at SparkService.java:25 count at SparkService.java:25	2018/06/25 08:26:55	1 s	2/2	3/3

- Spring 실행시 우리가 필요한 객체를 bean으로 생성하고 등록하려면 @Configuration을 이용하면 됨

```
package com.jbm.spark.configuration;

import org.apache.spark.sql.Session;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class WebApplicationConfig {

    @Bean
    public String test() {
        System.out.println("zzz");
        return "test입니다";
    }

    @Bean
    public Session sparkSession() {
        System.out.println("sparkSession 시작!");

        Session spark = Session.builder().appName("Spark")
            .master("local")
            .getOrCreate();

        System.out.println("sparkSession 끝!");
        System.out.println(spark);

        return spark;
    }
}
```



## - applicationContext.xml에서

```
<context:annotation-config/>
```

```
<bean class="com.jbm.spark.configuration.WebApplicationConfig"/>
```

## - Spring 시작시 실행 후 bean으로 등록됨

```
sparkSession 시작!  
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties  
18/06/25 08:26:16 INFO SparkContext: Running Spark version 2.2.0  
18/06/25 08:26:16 WARN NativeCodeLoader: Unable to load native library libgfortran; using fallback implementation  
18/06/25 08:26:16 INFO SparkContext: Submitted application: sparkSession 시작!  
18/06/25 08:26:17 INFO SecurityManager: SecurityManager: org.apache.spark.SecurityManager  
18/06/25 08:26:17 INFO SecurityManager: SecurityManager: org.apache.spark.SecurityManager  
18/06/25 08:26:17 INFO SecurityManager: SecurityManager: org.apache.spark.SecurityManager  
18/06/25 08:26:17 INFO SecurityManager: SecurityManager: org.apache.spark.SecurityManager  
18/06/25 08:26:17 INFO SecurityManager: SecurityManager: org.apache.spark.SecurityManager  
18/06/25 08:26:17 INFO Utiils: Successfully started Utiils: org.apache.spark.util.Utils$  
18/06/25 08:26:17 INFO SparkEnv: Registering RemoteRuntimeInfo  
18/06/25 08:26:17 INFO SparkEnv: Registering RemoteRuntimeInfo  
18/06/25 08:26:17 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper  
18/06/25 08:26:17 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper  
18/06/25 08:26:17 INFO DiskBlockManager: Created disk block manager  
18/06/25 08:26:17 INFO MemoryStore: MemoryStore started  
18/06/25 08:26:17 INFO SparkEnv: Registering RemoteRuntimeInfo  
18/06/25 08:26:18 INFO Utiils: Successfully started Utiils: org.apache.spark.util.Utils$  
18/06/25 08:26:18 INFO SparkUI: Bound SparkUI to 18080  
18/06/25 08:26:18 INFO Executor: Starting executor  
18/06/25 08:26:18 INFO Utiils: Successfully started Utiils: org.apache.spark.util.Utils$  
18/06/25 08:26:18 INFO NettyBlockTransferService: NettyBlockTransferService started  
18/06/25 08:26:18 INFO BlockManager: Using org.apache.spark.storage.DefaultTopologyMapper  
18/06/25 08:26:18 INFO BlockManagerMaster: Registered block manager  
18/06/25 08:26:18 INFO BlockManagerMasterEndpoint: Registered block manager  
18/06/25 08:26:18 INFO BlockManagerMaster: Registered block manager  
18/06/25 08:26:18 INFO BlockManager: Initial  
sparkSession 끝!  
org.apache.spark.sql.SparkSession@7bea8999
```

## - SparkService 클래스에 필요한 메서드를 정의함

```
public long getMovieRating() {  
  
    long start = System.currentTimeMillis();  
  
    Dataset<String> ratings =  
  
    sparkSession.read().textFile("hdfs://192.168.56.101:8020/user/hive/warehouse/test.db/ratings/");  
  
    String sql = "select * from ratings";  
    System.out.println("Running: " + sql);  
    ratings.show();  
    long count = ratings.count();  
    long time = System.currentTimeMillis() - start;  
    System.out.println("수행시간 : " + time + "ms");  
    return count;  
  
}
```

## - Controller에 주입받아 메서드 호출

```
package com.jbm.spark.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.jbm.spark.util.SparkService;

@RestController
public class AjaxController {

    private SparkService sparkService;

    public void setSparkService(SparkService sparkService) {
        this.sparkService = sparkService;
    }

    @RequestMapping("/ajax/chicken")
    public String count() {
        long count = sparkService.getCount();

        return "{\"chicken\":\"+count+\"}";
    }

    @RequestMapping("/ajax/rating")
    public String rating() {
        long count = sparkService.getMovieRating();

        return "{\"movie\":\"+count+\"}";
    }
}
```