## Client Side Setup

This section describes how to install the KNIME Big Data Executor for Spark extension in the [KNIME Analytics Platform](https://www.knime.org/knime) or the [KNIME Server](https://www.knime.org/knime-server).

### Installation

The KNIME Big Data Executor for Spark extension can be installed via the KNIME Update Manager. Got to *File-> Install KNIME Extensions…* and select the appropriate extension from the *KNIME.com Extension Store* category. For further details see the [KNIME installation instructions](https://tech.knime.org/installation-instructions).

If you don't have direct internet access you can also download a [zipped update site](http://update.knime.org/store/UpdateSite_Store_latest.zip) and register it in *File → Preferences → Install/Update → Available Software Sites*. Then follow the steps above.

Note that the KNIME Big Data Executor for Spark extension requires a license which you can purchase via the [KNIME Store](https://tech.knime.org/knime-store).

### Configuration

After installing the KNIME Big Data Executor for Spark extension you have to configure the client to work with your environment especially that the connection settings correspond with the settings of your spark-jobserver installation.

To setup the client go to the KNIME installation directory. In the directory go to *plugins\com.knime.bigdata.spark\_x.x.xxx\conf*. In the *conf* directory you will find the *application.conf* file which you need to open with a text editor e.g. notepad.

To establish the connection to the spark-jobserver you have to adapt the following setting of the *application.conf* file to your environment:

1. **jobServer:** This is the IP address or [DNS](https://en.wikipedia.org/wiki/Domain_Name_System) name of the Linux server the spark-jobserver is installed on.
2. **jobServerPort:** The port the spark-jobserver is listening for job requests as defined in the [*environment.conf*](#h.fzudxd45t37r) file.
3. **contextName:** The name of the Spark context. This should be the same for all users of the same spark-jobserver since Spark does not support multiple contexts within the same Java VM. If you want to use multiple contexts you have to install a spark-jobserver for each context.
4. **numCPUCores:** The number of cores to reserve for the Spark context. Equivalent to the Spark [configuration](https://spark.apache.org/docs/1.2.1/configuration.html) parameter *spark.cores.max*.
5. **memPerNode:** The amount of memory per node to reserve for the Spark context. Corresponds to the Spark [configuration](https://spark.apache.org/docs/1.2.1/configuration.html) parameter *spark.executor.memory*.

## Spark-Jobserver Installation

This section describes how to install the spark-jobserver on a Linux server. KNIME utilizes the spark-jobserver in the KNIME Big Data Executor for Spark.

### Background

The spark-jobserver provides a RESTful interface for submitting and managing [Apache Spark](http://spark-project.org) jobs, jars, and job contexts. Please check out the [spark-jobserver GIT hub repository](https://github.com/spark-jobserver/spark-jobserver) for further information, including licensing conditions, contributors and mailing lists.

### Version

The packaged version provided by KNIME and described in this document is based on Version 0.5.0 of the spark-jobserver. It has been deployed and tested with Spark Versions 1.2.1 and 1.2.2 and Hadoop Version 2.6.0.

### Installation

The spark-jobserver must be installed on a (Linux) server that is co-located in the same network as your Hadoop / Spark installation. It can be installed on the Hadoop master server or any other server that has unrestricted access to the Hadoop installation.

1. Create some directory for the spark-jobserver installation. For example */home/spark/job-server*.
2. Download the pre-packaged spark-jobserver file at <https://download.knime.org/bigdata/SparkJobServer4KNIME.tar.gz>
3. Unpack (tar xfvz SparkJobserver4KNIME.tar.gz) the pre-packaged file into your installation directory.
4. Edit *environment.conf* as appropriate. The most important settings are:
   1. **master** use master = "yarn-client" (default) when running [YARN](http://spark.apache.org/docs/1.2.1/running-on-yarn.html) or master = "spark://localhost:7077" (or similar) when running in [stand-alone](http://spark.apache.org/docs/1.2.1/spark-standalone.html) mode or master = “local[4]” for debugging.
   2. **rootdir** use rootdir = /tmp/jobserver/data (default) or some other temporary directory. This is where job information and jar files are kept (see also [Hints](#h.f2n9avxe5uye) below). Please note that some settings are overwritten by the KNIME configuration. Examples of overwritten settings are num-cpu-cores and memory-per-node.
5. Edit *settings.sh* as appropriate.
   1. **INSTALL\_DIR** this should point to the installation directory e.g. */home/spark/job-server* (but is not used)
   2. **SPARK\_HOME**, please change if Spark is not installed under /usr/local/spark
   3. **LOG\_DIR**, set to some log directory
6. Edit *log4j-server.properties* as appropriate.

### Starting the Spark-Jobserver

On the remote server, start the spark-jobserver in the installation directory with *server\_start.sh*.

### Stopping the Spark-Jobserver

Stop the spark-jobserver with *server\_stop.sh*.

### Hints

Point your browser to *http://<server>:8090* to check out the status of the spark-jobserver. Three different tabs provide information about active and completed jobs, contexts and jars.

The spark-jobserver does currently not support user credentials. This means that anybody who has access to the server where the spark-jobserver is running can start and stop contexts and jobs.

It might be advisable to re-start the spark-jobserver every once in a while and to clean-up the rootdir. Either remove the entire directory or only the jar files under rootdir.

At times, the spark-jobserver cannot be restarted when large tables were serialized from KNIME to Spark. It fails with a message similar to *java.io.UTFDataFormatException: encoded string too long: 6653559 bytes*. In that case it is advisable to clear the entire rootdir.