Installation guide

To setup the project on your local machine, please follow the steps in this installation guide. For additional information refer to the chapter “Setup”.

1. Install Eclipse

Follow the corresponding guides and tutorials in the Frege Wiki to set up the environment.
https://github.com/Frege/frege/wiki/Getting-Started
https://github.com/Frege/eclipse-plugin/wiki/fregIDE-Tutorial

2. Install FregIDE Eclipse plugin

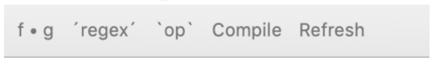
Follow the Installation tab in the FregIDE-tutorial (https://github.com/Frege/eclipse-plugin/wiki/fregIDE-Tutorial#installation)

3. Import project sources

Import the project sources by cloning and importing the git repository https://github.com/elrocqe/frege_spark.git as a project into the Eclipse IDE.

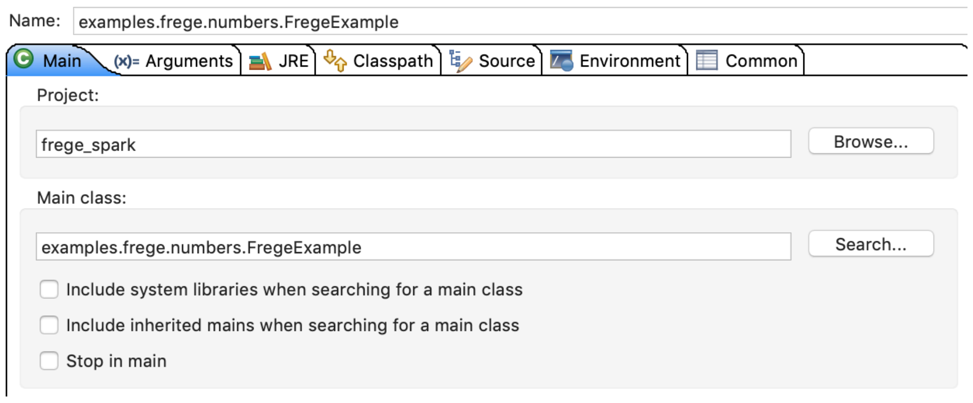
4. Enable Frege Builder

Right-click on the frege_spark project in one of the Explorer Views in Eclipse (Package or Project Explorer) to open the context menu and select “Enable Frege Builder”. Now your IDE should provide further toolbar entries when opening a .fr-file.



5. Create Run Configuration in Eclipse

Open the Run Configurations menu, which can be found through the Run Menu or by clicking on the arrow next to this icon.
Create a new Run Configuration for the application you would like to run. Possible applications can be found in src/main/frege/examples and src/main/java/examples.



Name: examples.frege.numbers.FregeExample

Project: frege_spark

Main class: examples.frege.numbers.FregeExample

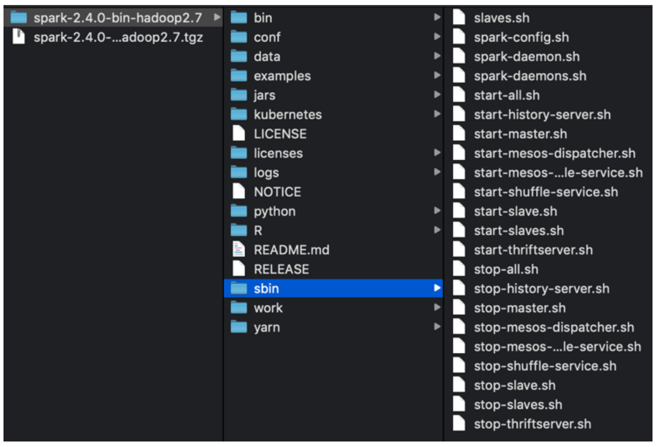
☐ Include system libraries when searching for a main class
☐ Include inherited mains when searching for a main class
☐ Stop in main

6. Run the application

Lastly, run the created RunConfiguration and check the console for output.

Local Apache Spark

The simplest way to run an Apache Spark application is just to add the dependency through the dependency management tool, in this case Maven, and run the application directly which will spawn an Apache Spark instance.
To run Apache Spark in a distributed cluster locally, a local distribution of Apache Spark is needed. The latest ones can be downloaded from https://spark.apache.org/downloads.html.
After unzipping the downloaded file, the following folder structure and files will be present:



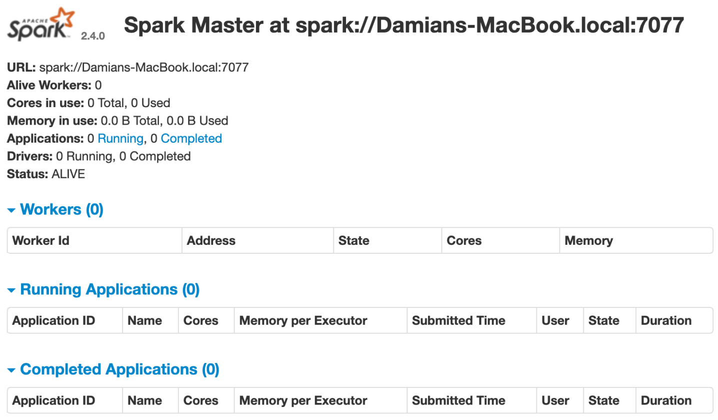
```
spark-2.4.0-bin-hadoop2.7    bin          slaves.sh
spark-2.4.0-...adoop2.7.tgz  conf         spark-config.sh
                             data         spark-daemon.sh
                             examples     spark-daemons.sh
                             jars         start-all.sh
                             kubernetes   start-history-server.sh
                             LICENSE      start-master.sh
                             licenses     start-mesos-dispatcher.sh
                             logs         start-mesos-...le-service.sh
                             NOTICE       start-shuffle-service.sh
                             python       start-slave.sh
                             R            start-slaves.sh
                             README.md    start-thriftserver.sh
                             RELEASE      stop-all.sh
                             sbin         stop-history-server.sh
                             work         stop-master.sh
                             yarn         stop-mesos-dispatcher.sh
                                          stop-mesos-...le-service.sh
                                          stop-shuffle-service.sh
                                          stop-slave.sh
                                          stop-slaves.sh
                                          stop-thriftserver.sh
```

To start a distributed cluster, a master and a slave node have to be spawned. This can be done by running the starter scripts in the sbin folder inside a terminal.

```
./start-master.sh
./start-slave <your-master-url>

# e.g. ./start-slave.sh spark://Damians-MacBook.local:7077
```

Afterwards, the status of the Apache Spark instance can be checked on http://localhost:8080



Spark Master at spark://Damians-MacBook.local:7077

URL: spark://Damians-MacBook.local:7077
Alive Workers: 0
Cores in use: 0 Total, 0 Used
Memory in use: 0.0 B Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (0)

Worker Id	Address	State	Cores	Memory

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Adjust the Apache Spark Config to

```
  sparkConfig.setMaster "spark://Damians-MacBook.local:7077"
```

instead of

```
   sparkConfig.setMaster "local"
```

and the application then will be run on the defined cluster.

In this web application, all the active workers, running and completed applications will be shown and you can also access the logs of the applications on the worker nodes.

Files Overview

The source files are separated into Frege source code, which is located under src/main/frege/ and the Java source code in src/main/java/.

src/main/frege

bindings	bindings to create Apache Spark functions
bindings.custom	custom bindings for examples
bindings.spark	native declarations for Apache Spark classes
bindings.spark.sql	native declarations for Apache Spark classes
bindings.testing	testing bindings for assertEquals
config	configuration file
examples	executable Frege Apache Spark applications
examples.helpers	Frege helper class for examples
functions	Frege FunctionPool to be executed (e.g. via interpreter)
snippets	Frege code snippets
utils	small tool to create example input file

src/main/java

bindings	corresponding Java code for the bindings
bindings.custom	corresponding Java code for the bindings
examples	executable Java application
functions	Java FunctionPool to be executed (e.g. via reflections)
script	helper class for Frege interpreter
snippets	Java code snippets

Executable Examples

Runnable Apache Spark applications are named …Example.fr or …Example.java and can be found in the examples-package.

```
src/main/frege
  examples
    DataSetIntegrationExample.fr
    FregeRDDExample.fr
    HelloSparkExample.fr
    IntegrationRDDExample.fr
    InteroperabilityRDDExample.fr
    InterpretationRDDExample.fr
    IOIntegrationExample.fr
    NativeModuleRDDExample.fr
    ReflectionRDDExample.fr

src/main/java
  examples
    InterpreterExample
    JavaDataFrameExample.java
    JavaRDDExample.java
```

These applications can be started by using dedicated runConfigurations in Eclipse.