Running KEITH

- Setting up your Eclipse
- Java Application
- Setting up a KEITH developer setup...
- ... on linux:
- ... on windows:
- ... on mac:

- Stuff that may help
  - How to run KEITH in developer setup (socket)
  - Running the already build LS

- Known Issues

Setting up your Eclipse

For everything not mentioned here refer to Getting Eclipse guide. Choose the 2021-06 Eclipse version and I personally recommend Eclipse for RCP & RAP developers since the Plug-In Development perspective is the default one. Another helpful perspective might be the Git perspective.

Use the installer go to advanced mode, add the KIELER url. If you plan to develop for the semantic language server (e.g. for the compiler) or to work with the SCCharts language, you should select KIELER semantics; for diagram only KIELER pragmatics. In any case select the keith stream in semantics or the master stream in pragmatics.

If you plan to develop in elk at the same time first select the Eclipse Layout Kernel setup and after that either the semantics or pragmatics setup.

Wait till everything installs and the setup tasks finish. If you have any problems in this stage refer to the Getting Eclipse guide or ask your advisors.

Make sure that you have necessary forks of ELK/KLighD set up.

If you have problems in the workspace that are still there after a clean build do the following:

- Disable Project>Build automatically
- Select all KLighD and pragmatics plugins and do Project>Clean>Only selected and build only selected
- Do the same for the semantics projects
- Enable Project>Build automatically

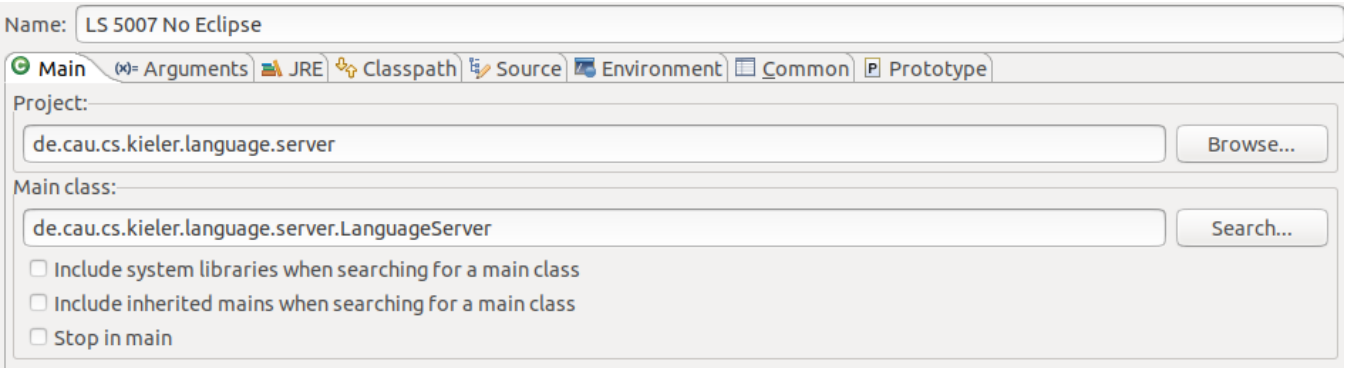
Java Application

If you have used the semantics setup, there is a pre-configured run configuration that you can use to execute the KIELER Language Server. For that, go into the **Run>[Run|Debug] Configurations>Java Application>LanguageServer with KLighD in Workspace** and execute that.

Described below is how you can set this configuration up for yourself for any manual configuration within the language server:

To run the language server go to *Run Configurations* create a new *Java Application* run configuration.

Select the ***Project*** *de.cau.cs.kieler.language.server* or *de.cau.cs.kieler.pragmatics.language.server* and the **Main class** *de.cau.cs.kieler.language.server.LanguageServer* or *de.cau.cs.kieler.pragmatics.language.server.PragmaticsLanguageServer*.



In the next step all projects that you want to include in your language server have to be added to the classpath.

Go to **Classpath**, select **User Entries**, click **Add Projects...**, and select all required projects (if you are unsure just add all of them).

Click on **Advanced**>**Add Folders** add select the project folders of all projects you added earlier.

In the **Arguments** tab make sure to add *-Dport=5007* to the **VM arguments**.

The default port to which KEITH tries to connect is 5007. You can of course change this for the language server but be aware that this has to be changed in KEITH too.

Setting up a KEITH developer setup...

General requirements:

- node (and additional dependencies see Theia developer guide)
- npm (whatever node installs)
- yarn (latest version)
- Python (2.7.X)
- gcc, g++, and make (for native dependencies of some npm packages)
- Visual Studio Code (latest version)
- a cloned keith repository

... on linux:

(Theia has a guide for extension development that might be helpful)

install node (for the version we refer to the Theia developer guide):

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.5/install.sh | bash
nvm install 10
```

Install python if you haven't (remember: Python 2: 👍, Python 3: 👎).

`Install yarn (a package manager build on the package manager npm):`

```
npm install -g yarn
```

... on windows:

Install node for windows. I personally used the `.msi`. For the version refer to the Theia developer guide.

Use that to install windows-build-tools by executing the command in an administrative powershell.

```
npm install -g windows-build-tools
```

This installs make, gcc, g++, python and all this. Somehow this does not really terminate. If nothing happens anymore it may be finished, just kill the process if it does not terminate.

All the installed executables are not in the path and that is okay. This is not needed since yarn/npm knows how to call them when needed.

Yarn can be downloaded and installed from here.

Known Problems in this step

If python3 was already installed this may cause some problems.

... on mac:

Get a package manager, something like brew.

Use brew to install all necessary stuff.

Apparently there is an issue with xcode-select: Theia developers recommend the following:

```
xcode-select --install
```

After doing this for your OS all that is missing is running KEITH (in developer setup) and setting up your Eclipse for language server development).

Stuff that may help

How to run KEITH in developer setup (socket)

Run the following to build and run KEITH in its developer setup (in socket mode, so the LS has to be started separately)

Running KEITH in the browser

```
yarn && cd keith-app && yarn run socket
```

yarn builds all the stuff. *yarn run socket* in keith-app starts the application. After an initial build via yarn you can run *yarn watch* to watch the changes in your repository. In another console you run yarn run socket in keith-app. Now refreshing your browser is enough to apply the changes.

Per default the KEITH opens on localhost:3000.

> If you previously build keith electron, you have to execute `yarn run rebuild:browser`

Run Launch in Chrome via VSCode to open a chrome browser on localhost:3000

This is necessary to be able to debug in VSCode.

Running KEITH as (unbundled) electron app

```
yarn && yarn run rebuild:electron && cd keith-app-electron && yarn run socket
```

yarn builds all the stuff. *yarn run socket* in keith-app-electron starts the application. After an initial build via yarn you can run *yarn watch* to watch the changes in your repository. In another console you run yarn run socket in keith-app-electron. Now refreshing your browser is enough to apply the changes.

> If you previously build keith electron, you have to execute `yarn run rebuild:electron`

Running the already build LS

> In the current builds, there seems to be a problem with the packaging of the jar file and executing will cause a ClassNotFoundException for org/eclipse/ui/IStorageEditorInput when initializing the LS. We are looking into this issue. For now, setup your Eclipse as described above.

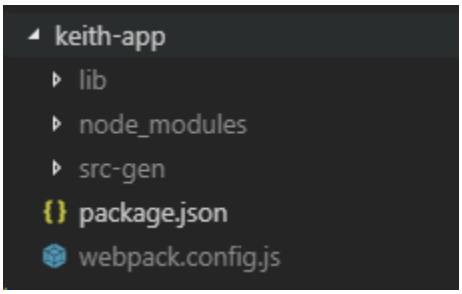
Go to the latest Bamboo build and go to Artifacts.

Select the language server for your OS (this will be a jar file) and run it via:

```
java -Dport=5007 -jar <name-of-the-jar-file>
```

5007 is the standard port KEITH is currently connecting to in socket mode. You can find this port in your Theia application at the following location:

Assume you are in the keith repository. Go to `keith-app`, you should see something like this:



Open the `package.json`. In the `package.json` are several scripts defined.

```
"scripts": {
        "prepare": "yarn run clean && yarn build",
        "clean": "theia clean",
        "build": "theia build --mode development",
        "start": "theia start --root-dir=../workspace",
        "socket": "node ./src-gen/backend/main.js --root-dir=../workspace --LSP_PORT=5007 --port=3000 --
loglevel=debug",
        "watch": "theia build --watch --mode development"
    },
```

The `LSP_PORT` option is used to activate the connection via socket. It is also possible to specify a relative location to a LS via `LS_PATH=<path to LS>`.

Known Issues

Known issues for windows:

nsfw.code not found: In the top level package.json exists a script called postinstall. Remove this on windows, delete the node_modules folder and rebuilt the application. This is a known issue of electron-builder.

Known issues on mac:

(this might already be resolved, has not been tested yet though)

Since SWT is still used as part of the diagram synthesis (but is not relevant anymore). Since it is not called on the main thread this causes a deadlock. Therefore mac just does not work.

Known issues:

- KEITH works in the browser/electron app, but not in the electron app/browser with the following error message: `symbol lookup error: ... symbol lookup error: .../keith/node_modules/nsfw/build/Release/nsfw.node: undefined symbol: _ZN2v816FunctionTemplate3NewEPNS_7IsolateEPFvRKNS_20FunctionCallbackInfoINS_5ValueEEEENS_5LocalIS4_EENSA_INS_9SignatureEEEiNS_19ConstructorBehaviorENS_14SideEffectTypeE Done in 0.90s.`
    - run `yarn run rebuild:electron/browser` after yarn to fix this. If it does not work, delete the `node_modules` folder and try again (for browser version rebuild browser is not needed, since `yarn` already builds the correct sources).