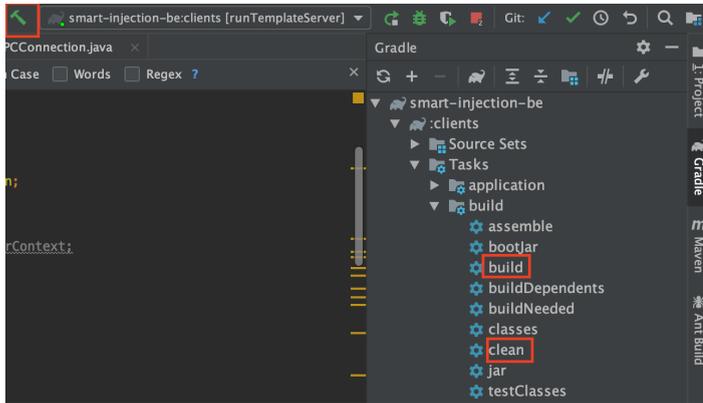


Smart Injection Setup Procedure

Please refer to [part 1](#) to setup the backend, [part 2](#) to setup the frontend and [part 3](#) to run the application.

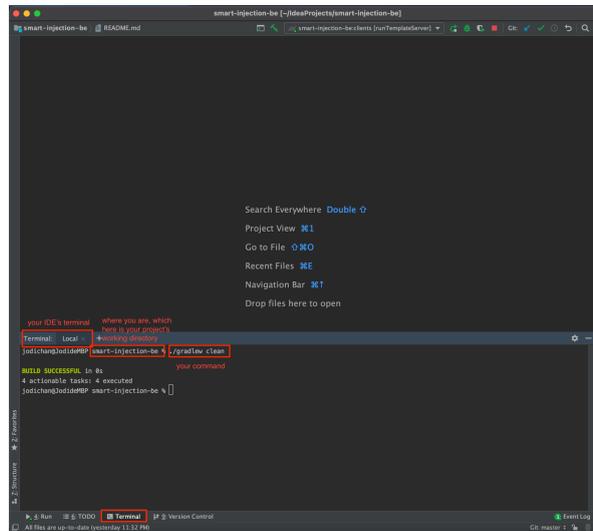
Part 1: Running the back-end code with **smart-inject-be**

Note: you can access each gradle tasks I've instructed from the UI, if it's easier. Otherwise follow along the instructions.



1. Clean your project

- a. Run the command in your IDE's terminal (where you will be in your project's directory):
`./gradlew clean` (deletes all your old jar files)
 - i. This is what you should see:

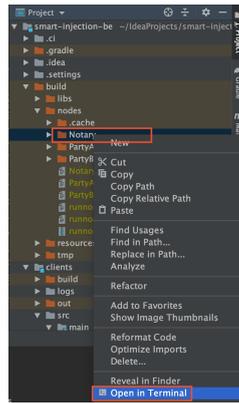


1.

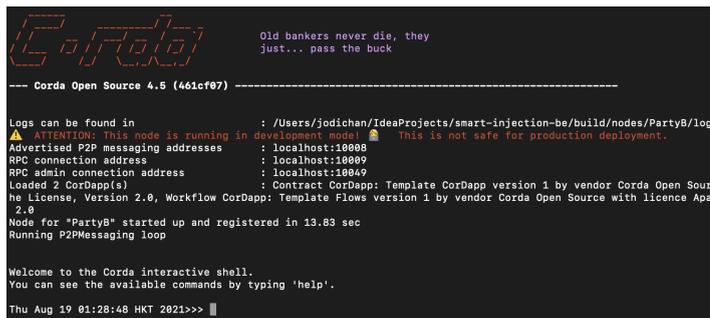
2. Deploy your corda nodes (otherwise the build folder will not have generated the nodes)

- a. Run the command in your IDE's terminal: `./gradlew deployNodes -x test` (build corda jar files for all your nodes)
- b. Two ways to run your 3 nodes:
 - i. Easiest is to run this in your IDE's terminal: `build/nodes/runnodes`
 1. Don't move your mouse, let it load on its own. Expect three windows (for the 3 nodes) to open up on your computer's terminal.

- ii. If the three windows aren't opening properly, you can run the nodes individually:

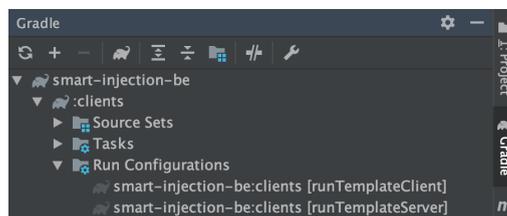


1. Go into 'build' folder, under 'Notary', right click & click 'Open in terminal' (so you are in this folder on the terminal).
 - a. In this terminal, run: `java -jar corda.jar`
 2. Go into 'build' folder, under 'PartyA', right click & click 'Open in terminal'.
 - a. In this terminal, run: `java -jar corda.jar`
 3. Go into 'build' folder, under 'PartyB', right click & click 'Open in terminal'.
 - a. In this terminal, run: `java -jar corda.jar`
- c. This is what you would see in the terminal when a node (in this case, Party B) is successfully running:



- i.

3. Build your project
 - a. Run this in your IDE's terminal: `./gradlew build -x test` (building new jar file of your app). Alternatively, from the UI, click on the green hammer icon.
4. Run template client & template server
 - a. In gradle's tool bar here:



- i.

- ii. Find & double click each of the following:
 1. smart-injection-be → clients → run configurations → smart-injection-be:clients [runTemplateClient]
 - a. Note: you will see this after

```
Run: smart-injection-be:clients [runTemplateClient]... smart-injection-be:clients [runTemplateSer...
12:03:54 AM: Executing task 'runTemplateClient'...
> Task :contracts:compileJava UP-TO-DATE
> Task :contracts:processResources NO-SOURCE
> Task :contracts:classes UP-TO-DATE
> Task :contracts:configureCordappFatJar
> Task :contracts:jar UP-TO-DATE
> Task :workflows:compileJava UP-TO-DATE
> Task :workflows:processResources UP-TO-DATE
> Task :workflows:classes UP-TO-DATE
> Task :workflows:configureCordappFatJar
> Task :workflows:jar UP-TO-DATE
> Task :clients:compileJava
> Task :clients:processResources UP-TO-DATE
> Task :clients:classes
> Task :clients:bootJar UP-TO-DATE
> Task :clients:jar SKIPPED
> Task :clients:assemble UP-TO-DATE
> Task :clients:runTemplateClient
I 00:02:58 | KCClient:logElapsedTime - Startup took 1872 msec
I 00:03:58 | Client.main - (NodeInfo{addresses=[localhost:10005], legalIdentitiesAndCerts=[0=PartyA, L=London, C=GB], platformVersion=7, serial=162})
```

i.

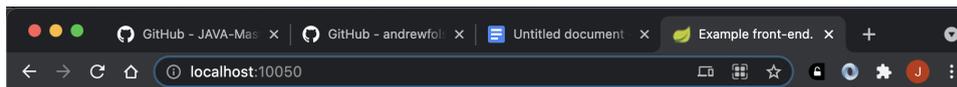
2. smart-injection-be → clients → run configurations → smart-injection-be:clients [runTemplateServer]

a. Note: you will see this after

```
Run: smart-injection-be:clients [runTemplateClient]... smart-injection-be:clients [runTemplateSer...
12:04:14 AM: Executing task 'runTemplateServer'...
> Task :contracts:compileJava UP-TO-DATE
> Task :contracts:processResources NO-SOURCE
> Task :contracts:classes UP-TO-DATE
> Task :contracts:configureCordappFatJar
> Task :contracts:jar UP-TO-DATE
> Task :workflows:compileJava UP-TO-DATE
> Task :workflows:processResources UP-TO-DATE
> Task :workflows:classes UP-TO-DATE
> Task :workflows:configureCordappFatJar
> Task :workflows:jar UP-TO-DATE
> Task :clients:compileJava
> Task :clients:processResources UP-TO-DATE
> Task :clients:classes
> Task :clients:bootJar UP-TO-DATE
> Task :clients:jar SKIPPED
> Task :clients:assemble UP-TO-DATE
> Task :clients:runTemplateServer
I 00:04:20 | Starter.logStarting - Starting Starter on JodideMPB with PID 25077 (/Users/jodichan/IdeaProjects/smart-injection-be/clients/build/clas
I 00:04:20 | Starter.logStartupProfileInfo - No active profile set, falling back to default profiles: default
I 00:04:20 | AnnotationConfigServletWebServerApplicationContext.prepareRefresh - Refreshing org.springframework.boot.web.servlet.context.Annotation
I 00:04:21 | TomcatWebServer.initialize - Tomcat initialized with port(s): 10050 (http)
I 00:04:21 | HikariJdbcDataSource - Initializing DataSourceHandler ["Hikari-nio-10050"]
```

i.

5. Check that it's working by navigating to <http://localhost:10050>. You should see this:



Define your front-end here.

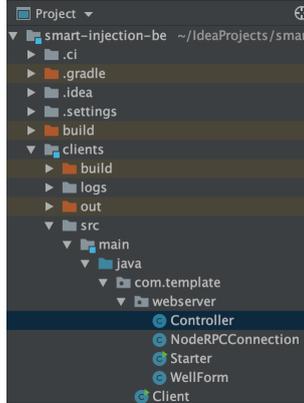
a.

Part 2: Running the front-end code with ui-smartinjection

1. Before you start, make sure you completed part 1 in building up the BE part.
2. Open your project & run this command in the terminal of your IDE: `npm install`
3. Once it's done, run: `npm serve`
4. Navigate to <http://localhost:4200/> (you will see a blank page when you click on it). Use this when accessing the different API endpoints (e.g., <http://localhost:4200/wo/new-project>, <http://localhost:4200/wo/my-wells>)

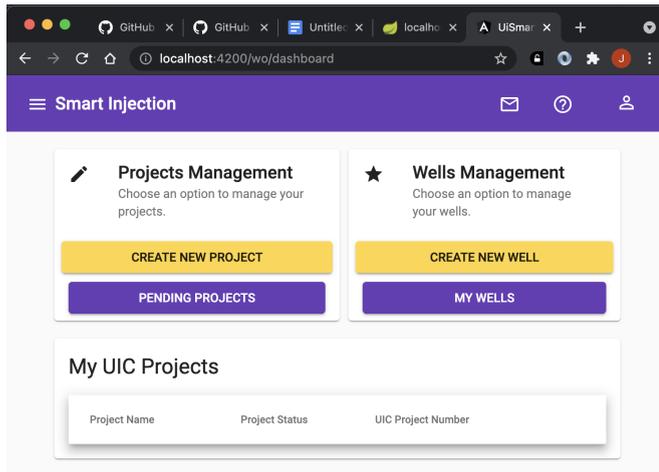
Part 3: Running the whole application together

1. Before you start, please note you must complete part 1 & 2.
2. Refer to the following file for all the API endpoints for which you will be using to navigate your app.
 - a. File path: smart-injection-be/clients/src/main/java/com/template/webserver/Controller.java
 - i. Or on the UI from the back-end code:



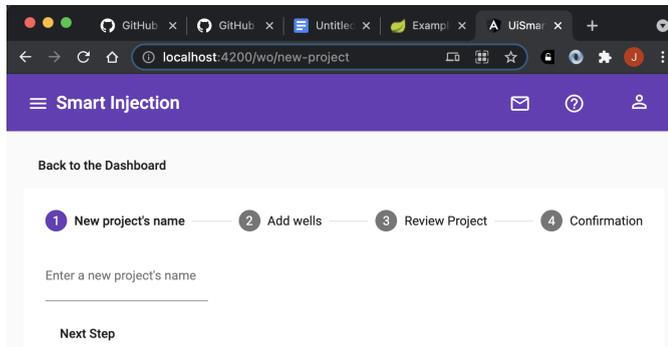
3. Examples when accessing the webpage:

- a. Dashboard view:



i.

- b. Create a new project:



i.