Modality Documentation

Table of Contents

Installation . . . 1
1. Install Java JDK . . . 1
2. Install WebFX CLI . . . 2
3. Install IntelliJ IDEA . . . 2
4. Install Git . . . 2
5. Create the Modality root . . . 2
6. Clone the codebase . . . 2
7. Install Postgres database . . . 2
8. Configure Postgres database . . . 2
Configure Modality for Development . . . 3
Resolve all Modality Dependencies . . . 3
Modality Server . . . 3
Back-Office Desktop . . . 5
Front-Office Desktop . . . 5
Build & Run Modality on Development . . . 5
Postgres * . . . 6
Modality Server * . . . 6
Back-Office Desktop . . . 6
Back-Office Web . . . 7
Back-Office Tablet . . . 9
Front-Office Desktop . . . 9
Front-Office Web . . . 9
Front-Office Mobile . . . 9

Installation

1. Install Java JDK

Modality is developed entirely in the Java language, and requires at least JDK 17+. Check whether this is installed:

```
java --version
```

If it is not installed, or is an older version, please refer to this guide.

2. Install WebFX CLI

We use the WebFX CLI to compile Modality for the web. Please follow this guide to install it.

3. Install IntelliJ IDEA

We develop Modality using the free, community edition of IntelliJ IDEA, and recommend you install this if you do not already have an IDE. IntelliJ allows you to easily compile and run the Modality server and clients, for the purpose of local development and testing.

All subsequent IDE-based examples given in this documentation will be based on IntelliJ.

4. Install Git

A git client is needed to retrieve the Modality codebase from GitHub. Check if git is installed:

```
git --version
```

If it is not installed, you may wish to refer to this guide.

5. Create the Modality root

```
mkdir -vp modality
export MODALITY_ROOT=${PWD}/modality
```

6. Clone the codebase

Git clone the Modality codebase via the terminal (or IntelliJ etc):

```
cd $MODALITY_ROOT
git clone https://github.com/modalityone/modality.git .
```

7. Install Postgres database

You may choose to install this natively, or you can use the modality-local-services repository, which contains a Docker-based solution to easily stand up Postgres on your machine.

8. Configure Postgres database

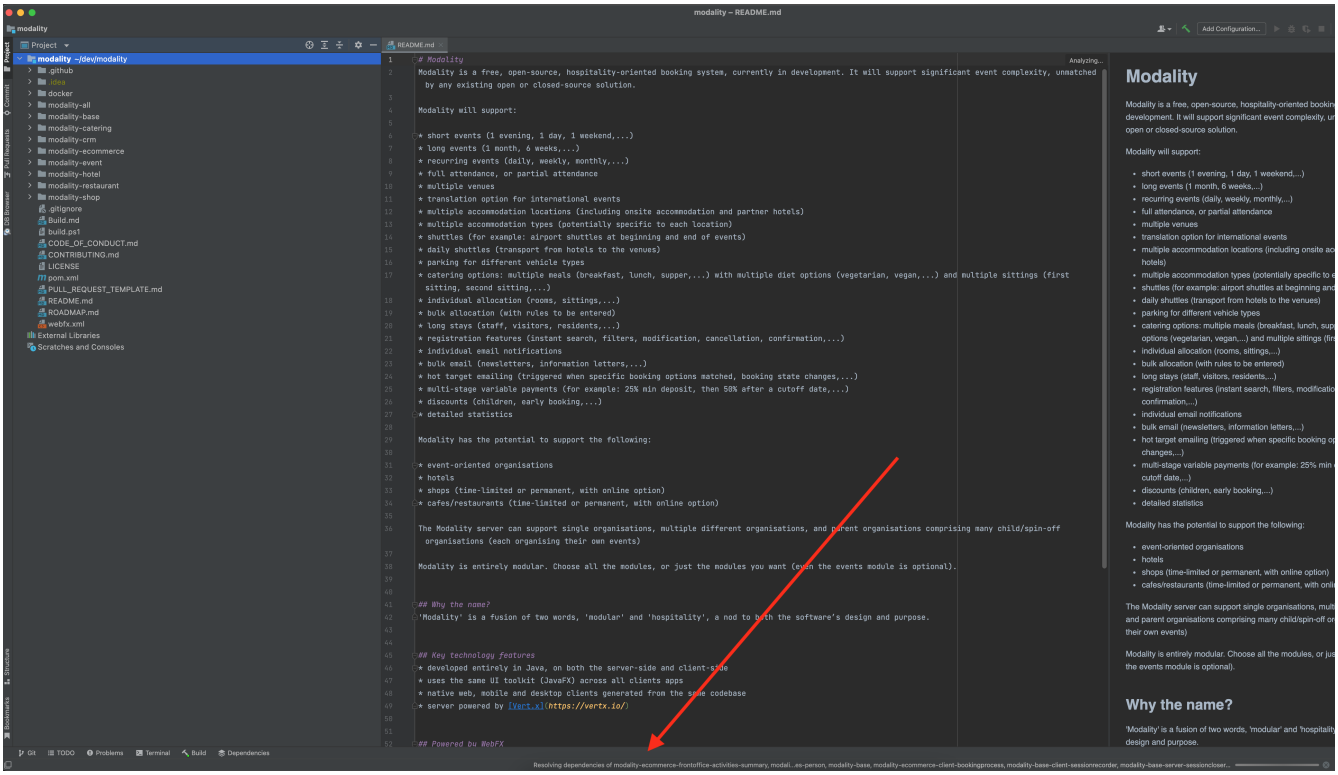
Modality expects a database and set of tables, and when running in local development mode expects to connect to them using default values. The database setup scripts and default

configuration values are described in the modality-local-services repository README.

Configure Modality for Development

Resolve all Modality Dependencies

Open Modality in IntelliJ and wait for all dependencies to be automatically resolved:

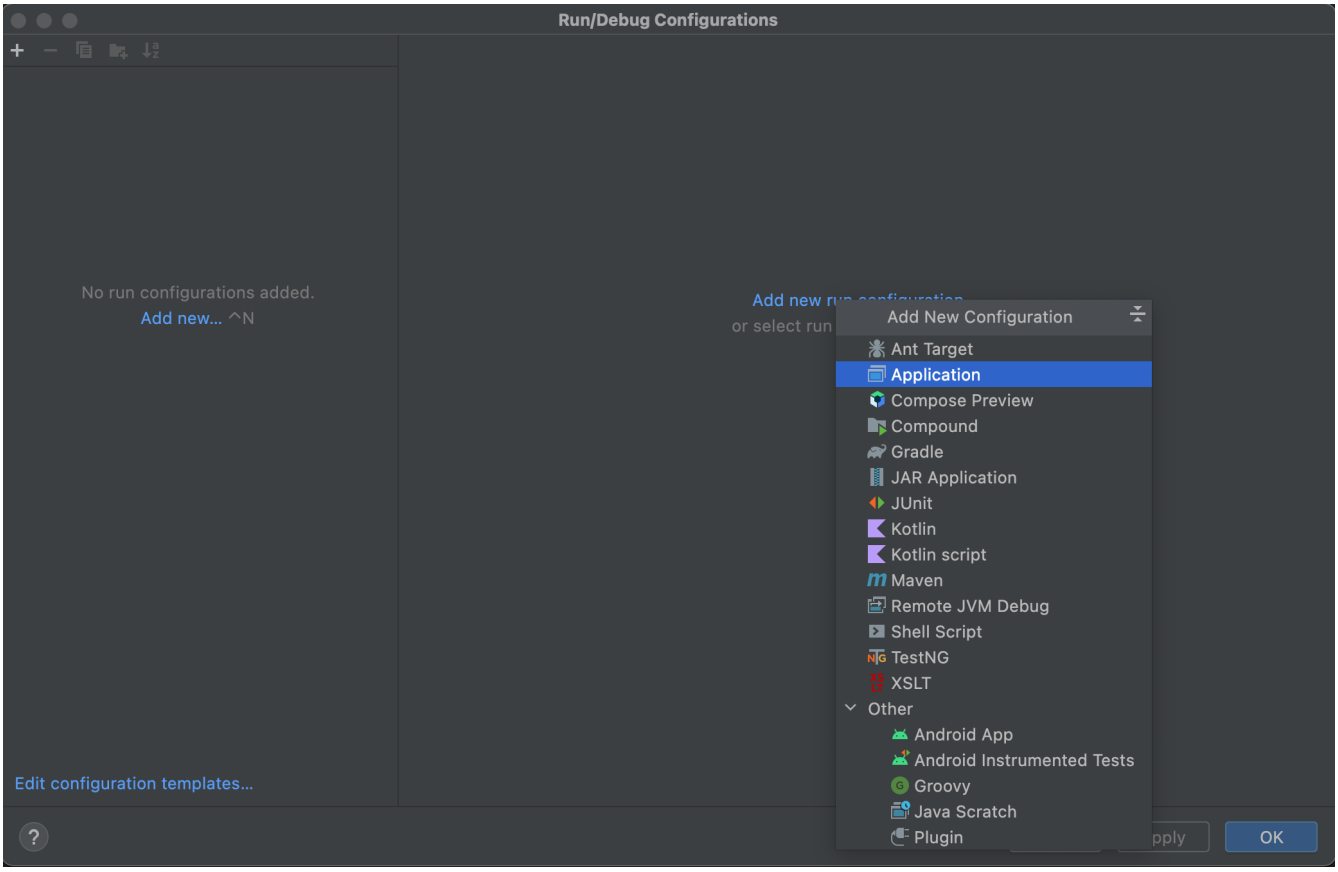


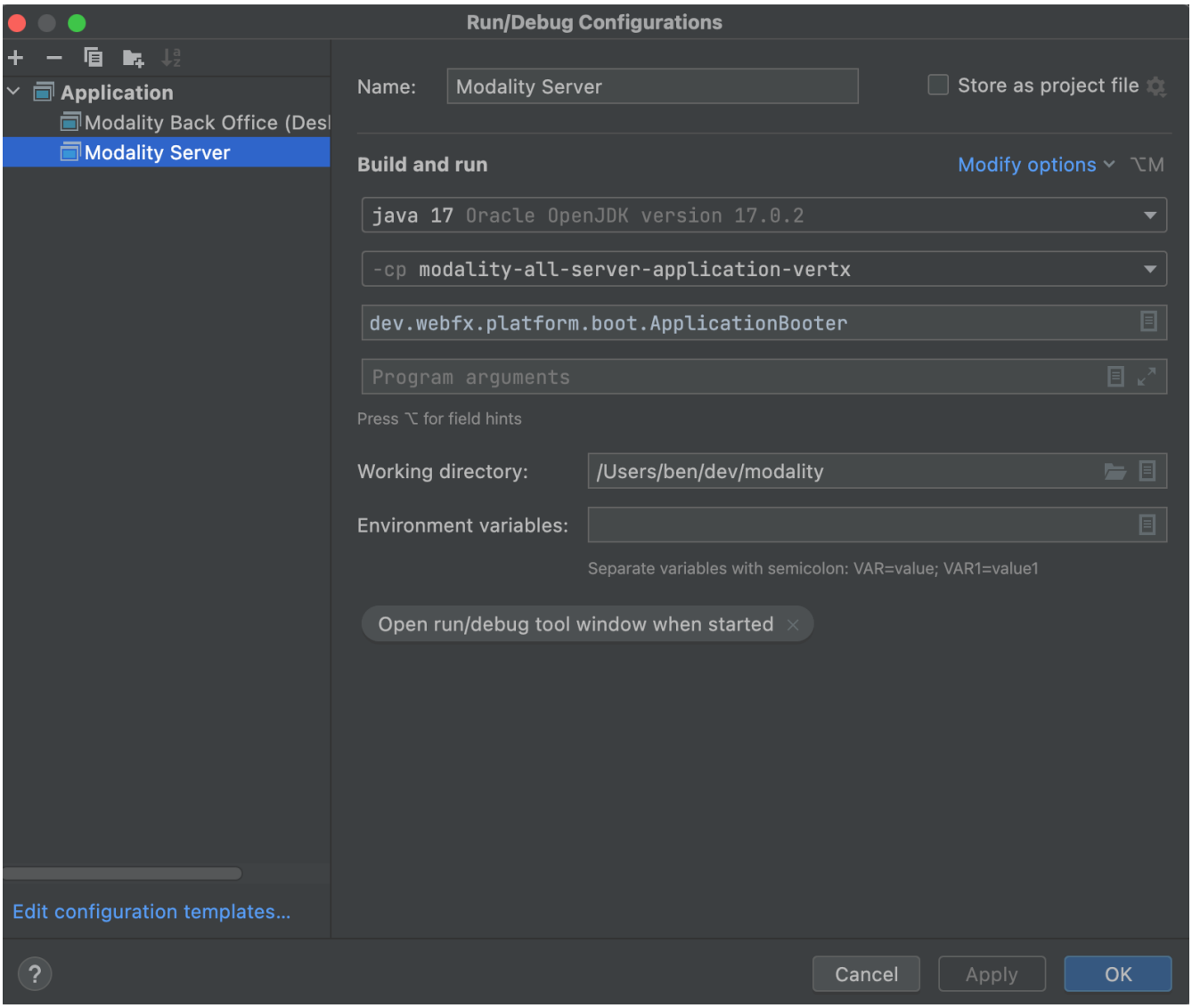
Modality Server

In order to run any of the Modality client applications, the Modality Server should first be running. The Modality Server is a Vert.x server that proxies requests to the database and is responsible for establishing and maintaining user sessions.

The easiest way to stand up the server locally is to create an application run configuration in the IDE.

In the IntelliJ menu, click `Run → Edit Configurations` to display the following dialog, and populate with the same details:


Run/Debug Configurations
No run configurations added.
Add new... ^N
Edit configuration templates...
Add New Configuration
Ant Target
Application
Compose Preview
Compound
Gradle
JAR Application
JUnit
Kotlin
Kotlin script
Maven
Remote JVM Debug
Shell Script
TestNG
XSLT
Other
Android App
Android Instrumented Tests
Groovy
Java Scratch
Plugin
OK

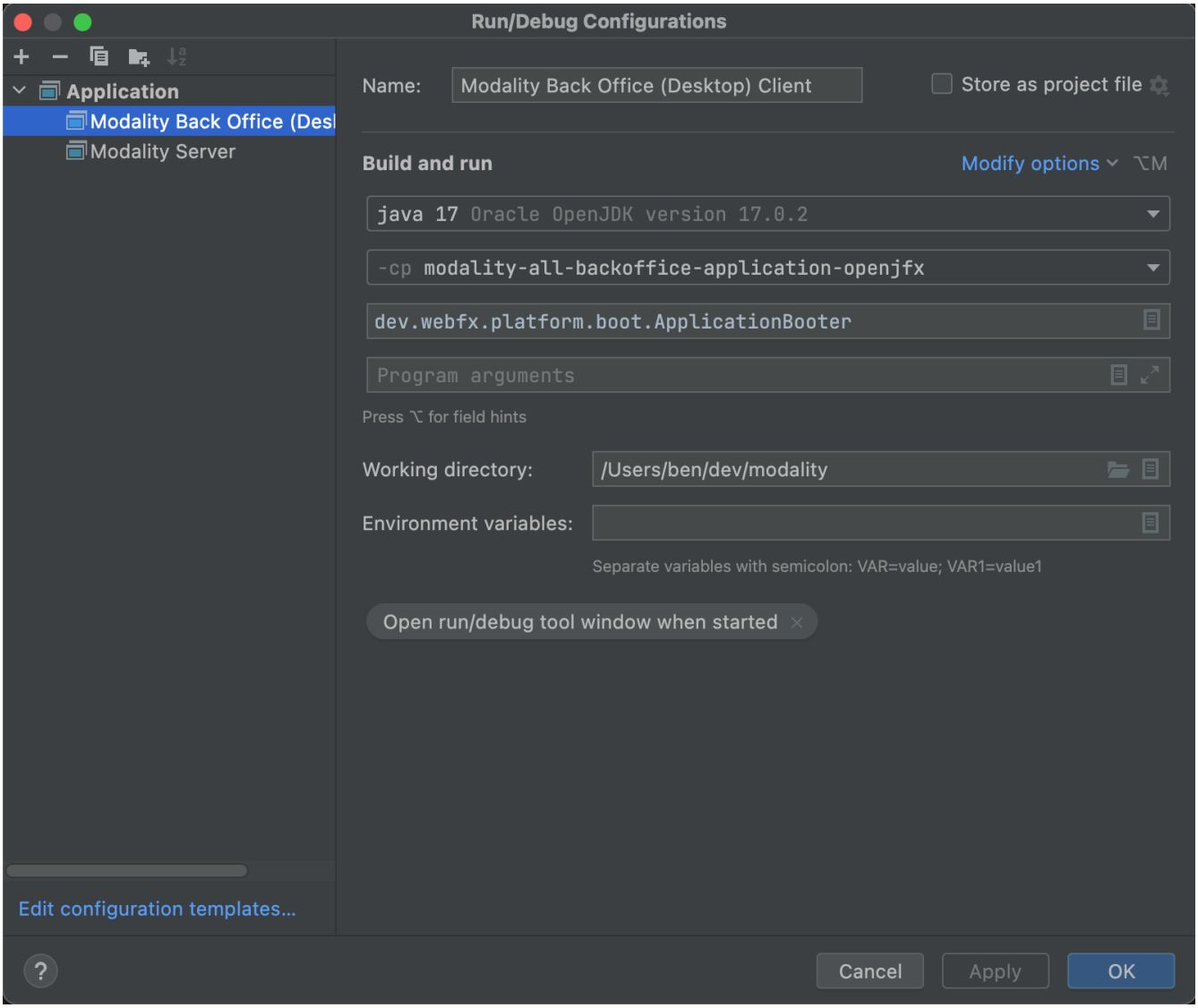

Run/Debug Configurations
Application
Modality Back Office (Des
Modality Server
Name:
Modality Server
Store as project file
Build and run
Modify options ⌥M
java 17 Oracle OpenJDK version 17.0.2
-cp modality-all-server-application-vertx
dev.webfx.platform.boot.ApplicationBooter
Program arguments
Press ⌥ for field hints
Working directory:
/Users/ben/dev/modality
Environment variables:
Separate variables with semicolon: VAR=value; VAR1=value1
Open run/debug tool window when started
Edit configuration templates...
Cancel
Apply
OK

Click 'OK' to save the configuration and close the dialog.

Back-Office Desktop

The Back-Office Desktop client is an application used by developers of Modality, and emulates the Web user interface used by administrators of the system. The idea is that developers build/modify screens using this client, then finally compile it for the Web using WebFX.

Create another run configuration and populate it with the details given in the screenshot below:



Click 'OK' to save the configuration and close the dialog.

Front-Office Desktop

The Front-Office Desktop client is not yet implemented.

Build & Run Modality on Development

The Modality clients run independently of each other, but all require the Modality Server to be

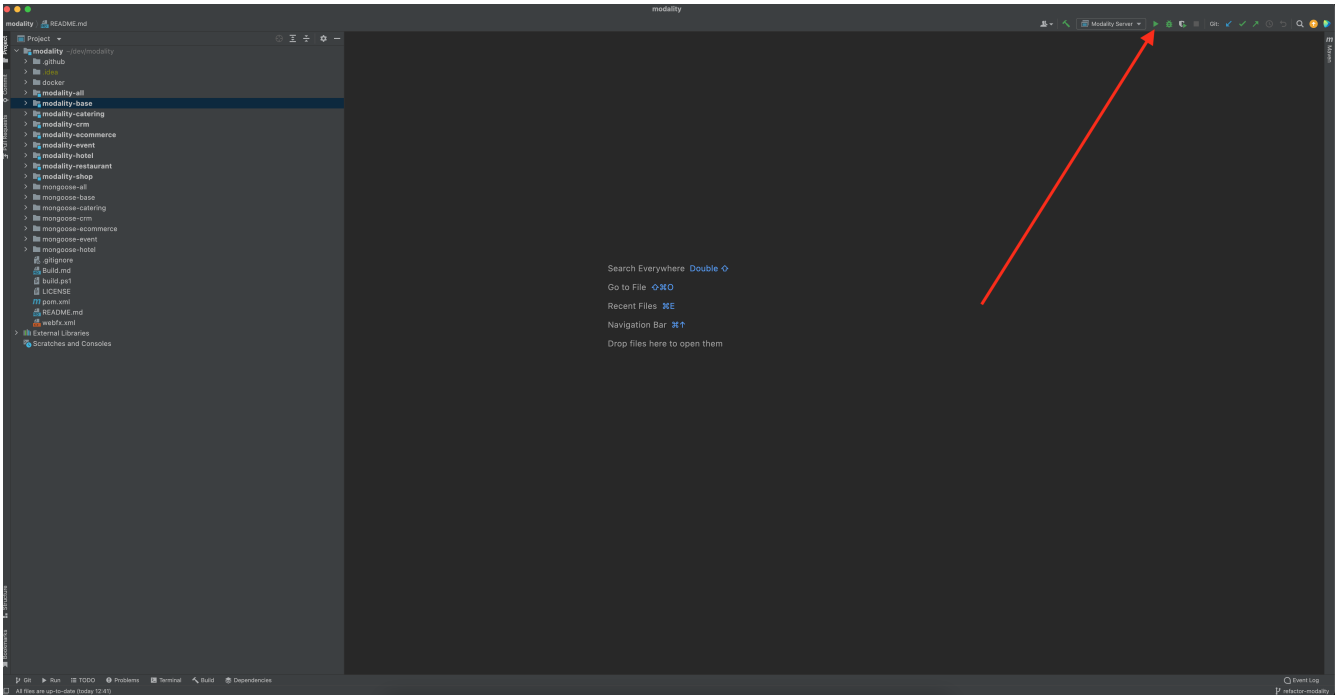
running, which in turn requires Postgres to be running. Therefore, the first two steps below are mandatory before running one or more of the Modality clients locally.

Postgres *

Ensure that Postgres is running.

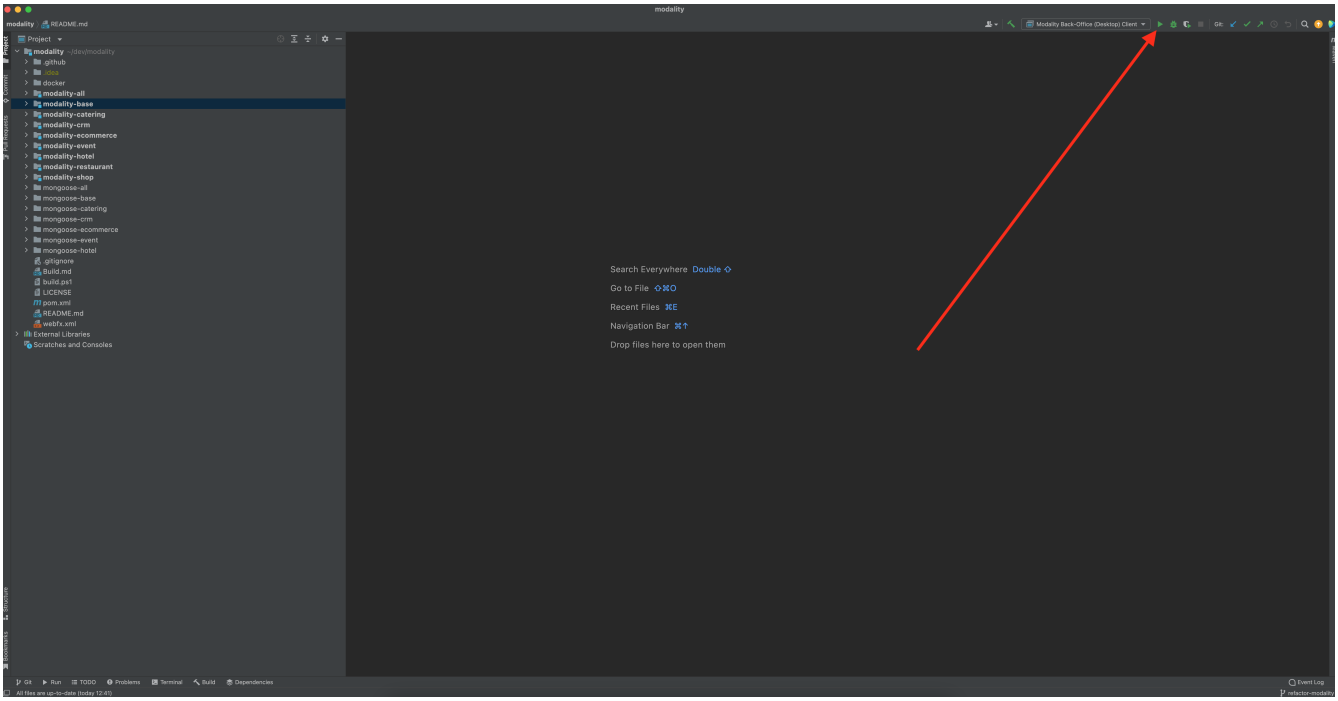
Modality Server *

Build and run the server by executing its run configuration:



Back-Office Desktop

Build and run the Back-Office Desktop client by executing its configuration:



The Back-Office Desktop client will then be ready to use.

Back-Office Web

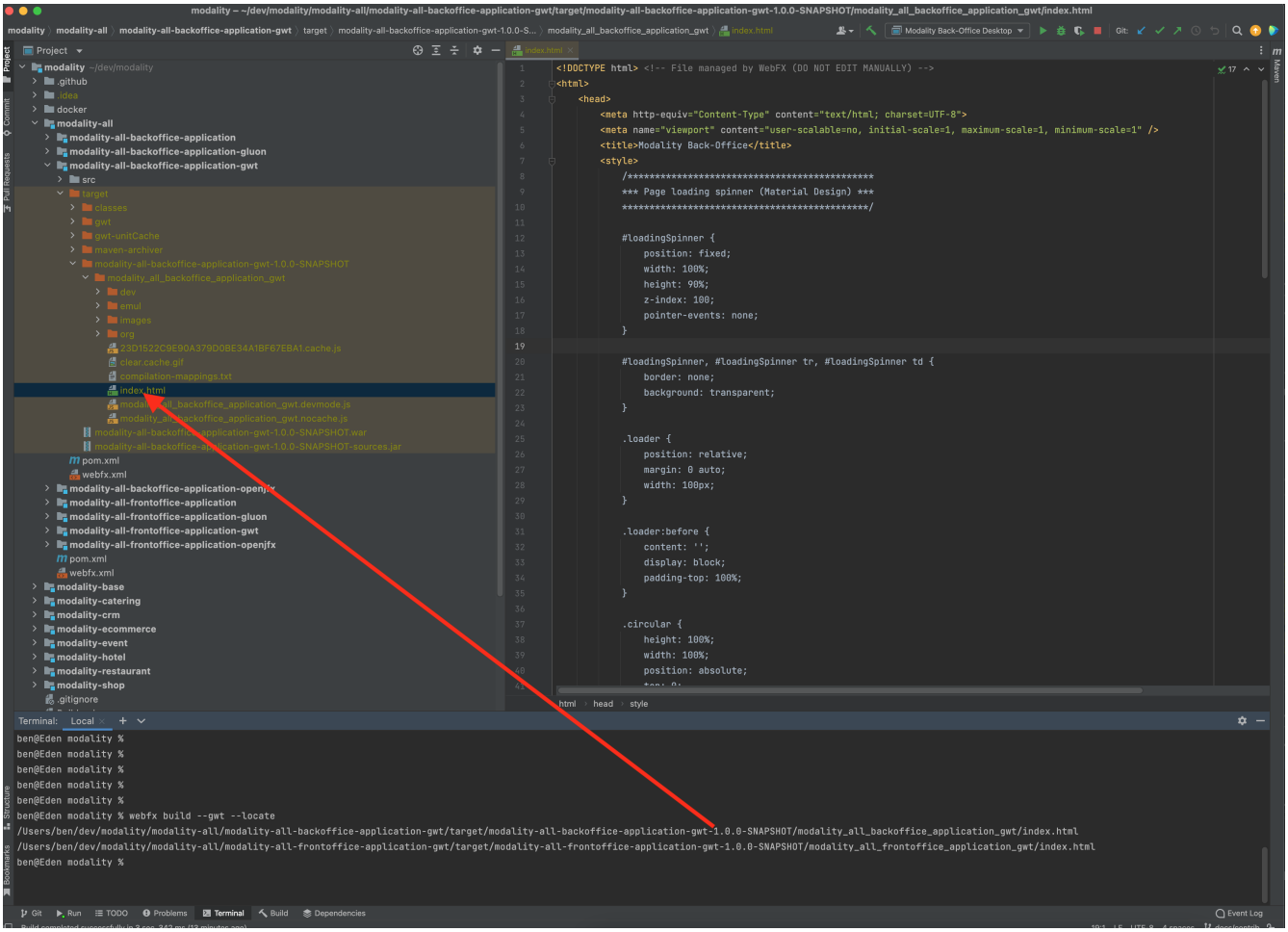
① First **build** the index.html file:

```
cd $MODALITY_ROOT
webfx build --gwt
```

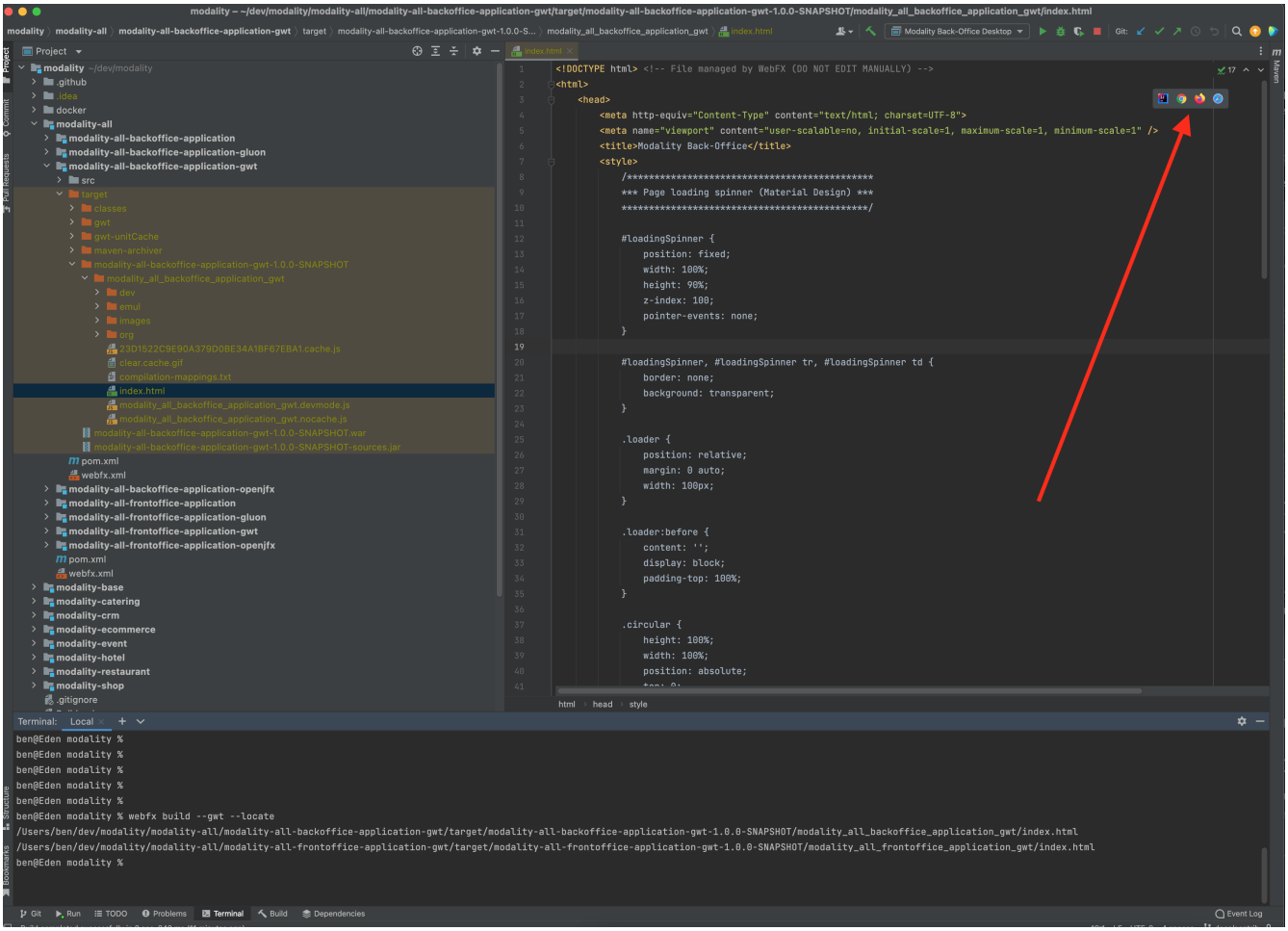
② Then **locate** the resultant index.html file on the filesystem:

```
webfx build --gwt --locate
```

③ In the IntelliJ Project window, navigate to the index file and double click.



① Hover the mouse over the index.html source code to display the browser options.



① Click on one of the browser icons to run the index.html in a browser.

Back-Office Tablet

Full instructions (and limitations) for compiling the codebase to native Android and iOS apps are available on the WebFX documentation site here.

Front-Office Desktop

The Front-Office Desktop client is not yet implemented.

Front-Office Web

The Front-Office Web client is not yet implemented.

Front-Office Mobile

The Front-Office Mobile client is not yet implemented.