Running Modelix locally
I spent a long time trying to get Modelix working. So far, the only success I've had is going about things in the following way:

Modelix was developed using MPS 2020.1.1, so I recommend using that version of MPS when experimenting with Modelix. It can be downloaded here: https://confluence.jetbrains.com/display/MPS/Download+MPS+2020.1.1

I've also found that adding certain plugins to MPS can cause Modelix to stop working, so I recommend starting with a clean install of MPS 2020.1.1.

If things were working before, but they suddenly don't any more, chances are a change has been made in the app data of MPS due to some setting change. I'm not exactly sure what the problem is when this happens, but deleting the app data and restarting MPS fixed the issue for me.

I do the following to get things working on Windows:

1. I cloned the Modelix git repository

2. Running 'gradlew.bat' in the root folder did not work for me, so instead I did the following:
    a. Using CygWin (https://cygwin.com/install.html*) I navigate to the Modelix folder
    b. Once there, I run the following 2 commands:

    ```
    find . -name "*.sh" | xargs dos2unix
    find . -name "gradlew" | xargs dos2unix
    ```

    to change the line endings of the shell script and gradle wrapper files to Unix format. Otherwise CygWin will complain about the line endings.
        i. *In order to do this you must make sure CygWin has the dos2unix packet installed
    c. Then I run:

    ```
    ./gradlew
    ```

    d. This will take a while to build. It will probably produce some errors, that's okay, as long as the build completes. If the build fails it's probably due to a Gradle or JDK version mismatch. Check the error message and make sure you have the right version running. For example, I had JDK 8 running. Setting it to 11 fixed it for me.
    e. Once this is done, start up MPS 2020.1.1 and open the 'mps' directory within the Modelix directory as an MPS project.
    f. Wait for the project to finish loading and indexing etc. Once this is done, you should see the yellow 'Cloud' view tab in the bottom of the screen.

3. Completing step 2 allows you to preview Modelix locally, so the MPS client is basically opening a socket for a UI client to connect to it. If you want to actually collaborate on an MPS project with multiple different clients on different computers, you need to start up a model server. The model server needs a database to store its files:
    a. Install Docker Desktop if you don't already have it and grant it the necessary privileges: https://www.docker.com/products/docker-desktop
    b. Once Docker is installed, start it up.
    c. With Docker up and running, open up a CygWin terminal and navigate to the Modelix directory. Run the following shell scripts:

```
./docker-build-db.sh
./docker-run-db.sh
```

    d. That's going to run the database which the Modelix model server will make use of. It might complain a little, which should be fine as long as it completes.
    e. Keep that terminal running. Open a new Cygwin terminal and navigate to the Modelix subdirectory called 'model-server'.
    f. There run the following command:

```
../gradlew run
```

    g. Keep that terminal running too. It keeps the model server up.

4. With all this done, an MPS client can connect to the model server in the following way:
    a. Open the "Cloud" view in the bottom left corner
    b. In the context menu of the root node labeled "Cloud" choose "Add Repository"
    c. Type the following:

```
http://<IP address of host>:28101/
```

        i. For the computer running the server that will just be 'localhost'.
        ii. For other computers on the local network you will need the IP address of the host computer.
        iii. If you want to connect to it from outside of the local network, you will have to set your router up to forward the packets to the host computer.
    d. Navigate to "default tree (default)" > "data [master]" > "ROOT #1", right-click it and choose "Add Module". Give your module a name.
    e. Right-click the module and choose "Add Model". Give your model a name.
    f. Right-click the module again and click "Bind to Transient Module"
    g. You should now see that module and the new model in the "Cloud" section at the end in the "Project" view.
    h. Open the "Model Properties" of the new model and add at least one language dependency.
    i. You should be able to see the new module appear via both the browser and in the 'Cloud' view of other connected MPS clients