# **Development**

## **Communication**

* General questions and discussion: portability-discuss@googlegroups.com
* [Slack channel](https://portability.slack.com/), email portability-maintainers@googlegroups.com for invitation
* Weekly Skype call, email portability-maintainers@googlegroups.com for invitation

## **First Run/Setup**

### **From top level directory**

* Install Git: sudo apt-get install git-all
* Install Gradle: sudo apt-get install gradle

### **From /client directory**

* Install NVM: curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
* Install Node: nvm install node
* Install Angular: npm install -g @angular/cli
* Install modules: npm install
* Test ng serve --port 3000 --proxy-config proxy.conf.json --environment local
* Test using SSL: ng serve --ssl --port 3000 --proxy-config proxy.https.conf.json --environment localhttps

## **IntelliJ setup**

The following instructions work for IntelliJ IDEA version 2017.2.6.

#### **Import the project**

* Open IntelliJ -> Import Project
* Navigate to the 'data-transfer-project' directory where you have git cloned the repo
* Import project from external model (Gradle) -> uncheck 'Create separate module per source set' -> Finish

#### **Enable annotation processing**

* Go to File -> Settings -> Build, Execution, Deployment -> Compiler -> Annotation Processors
* Check 'Enable annotation processing'
* Store generated sources relative to: 'Module content root' (not the default)
* Production sources directory: build/classes/java/

#### **Setup formatting**

* File -> Settings -> General -> Ensure line feed at file end on Save
* Install the 'google-java-format' plugin, and enable it in Settings

#### **Setup automatic license header**

* Go to File -> Settings -> Editor -> Copyright -> Copyright Profiles
* Click the green + button, add a new profile called "Data-Transfer-Project-Authors". Use this as the text:

Copyright $today.year The Data Transfer Project Authors.  
  
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at  
  
https://www.apache.org/licenses/LICENSE-2.0  
  
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.  
Create a new temp file (e.g. Test.java) in Intellij.  
Inside the file, enter Alt-Insert -> Copyright  
It should prompt you to select the new Copyright profile  
Test out another new test file - the copyright should be imported automatically (note: it might be collapsed so not immediately obvious)

* Create a new temp file (e.g. Test.java) in Intellij.
* Inside the file, enter Alt-Insert -> Copyright
* It should prompt you to select the new Copyright profile
* The copyright should be added to all new files from now on

## **Environment-specific settings**

Environment-specific settings (values for PortabilityFlags) are stored per environment in the environments/ directory. There is a subdirectory for each environment: local/, qa/, test/, and prod/. Each environment subdirectory contains a settings.yaml and common.yaml for PortabilityFlags values.

For local development and testing, a secrets.csv file should be included in config/environments/local. This should contain all your API keys, see secrets\_template.csv (TODO: add sample) for an example.

## **Building/Running the API server locally**

The following builds and optionally runs the API server on port:8080

* bin/build\_and\_run\_jar.sh api local
  + This copies over LOCAL secrets and settings and compiles the api jar.
  + This will also prompt you to run the jar.

## **Building/Running the Worker locally**

The following builds and optionally runs the worker binary

* bin/build\_and\_run\_jar.sh worker local
  + This copies over LOCAL secrets and settings and compiles the worker jar.
  + This will also prompt you to run the jar.

## **Running angular in dev mode**

The following commands will run the angular frontend in dev mode locally proxying requests to the local webserver.

* cd client/
* [optional] npm install #required first time through
* ng serve --port 3000 --proxy-config proxy.conf.json --environment local
* Or, using SSL: ng serve --ssl --port 3000 --proxy-config proxy.https.conf.json --environment localhttps

## **Running angular tests**

The following commands will run the angular frontend in dev mode locally proxying requests to the local webserver.

* cd client/
* ng test --env=local -sm=false
  + Flag for environment set to local
  + Flag for sourcemaps set to false gives better debugging messages

## **Deploying in production**

[See config/README.md](https://github.com/google/data-transfer-project/blob/master/config/README.md)

## **This is not an official Google product**