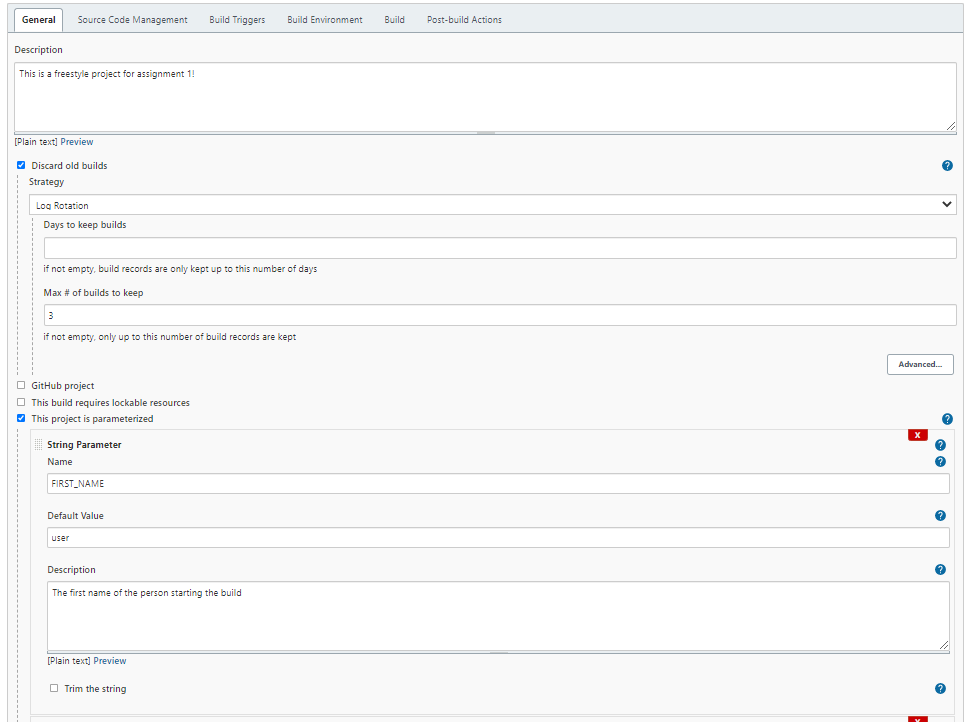
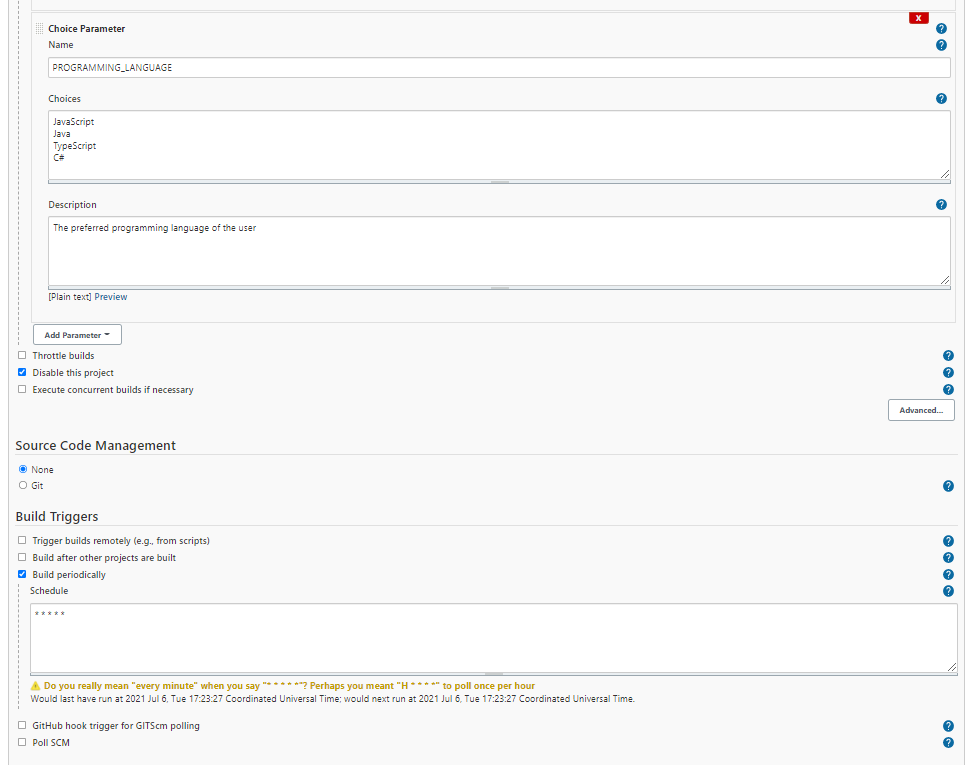
# Jenkins SIG - answers

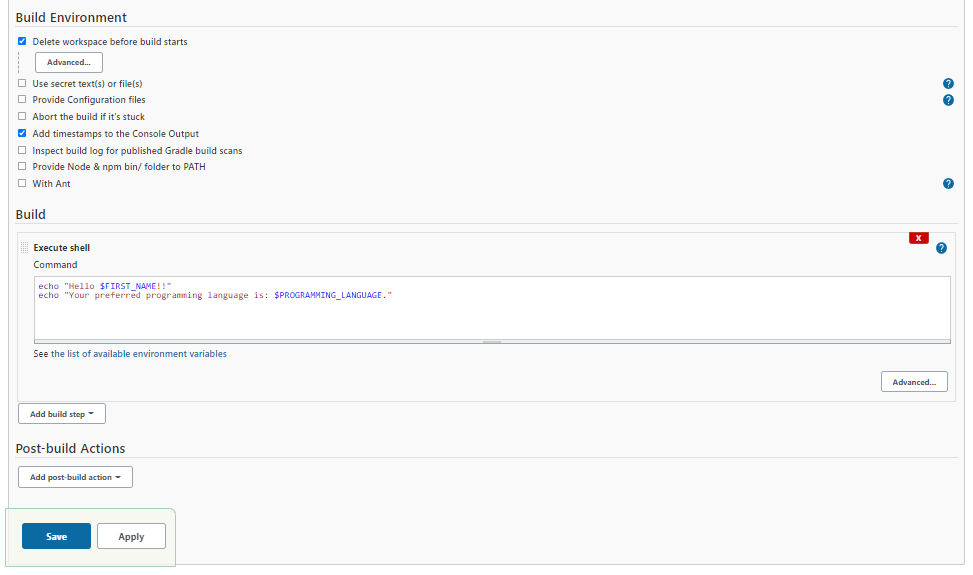
### Manually setup project/ pipeline

#### Assignment 1

Below screenshots show a possible solution to this assignment:





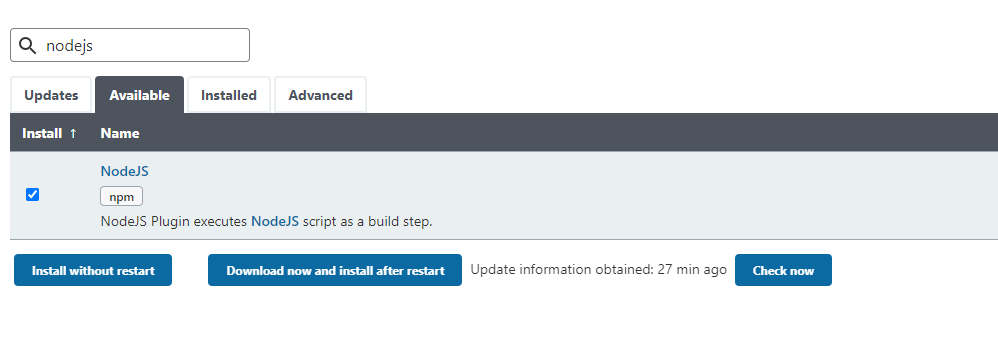


#### Assignment 2

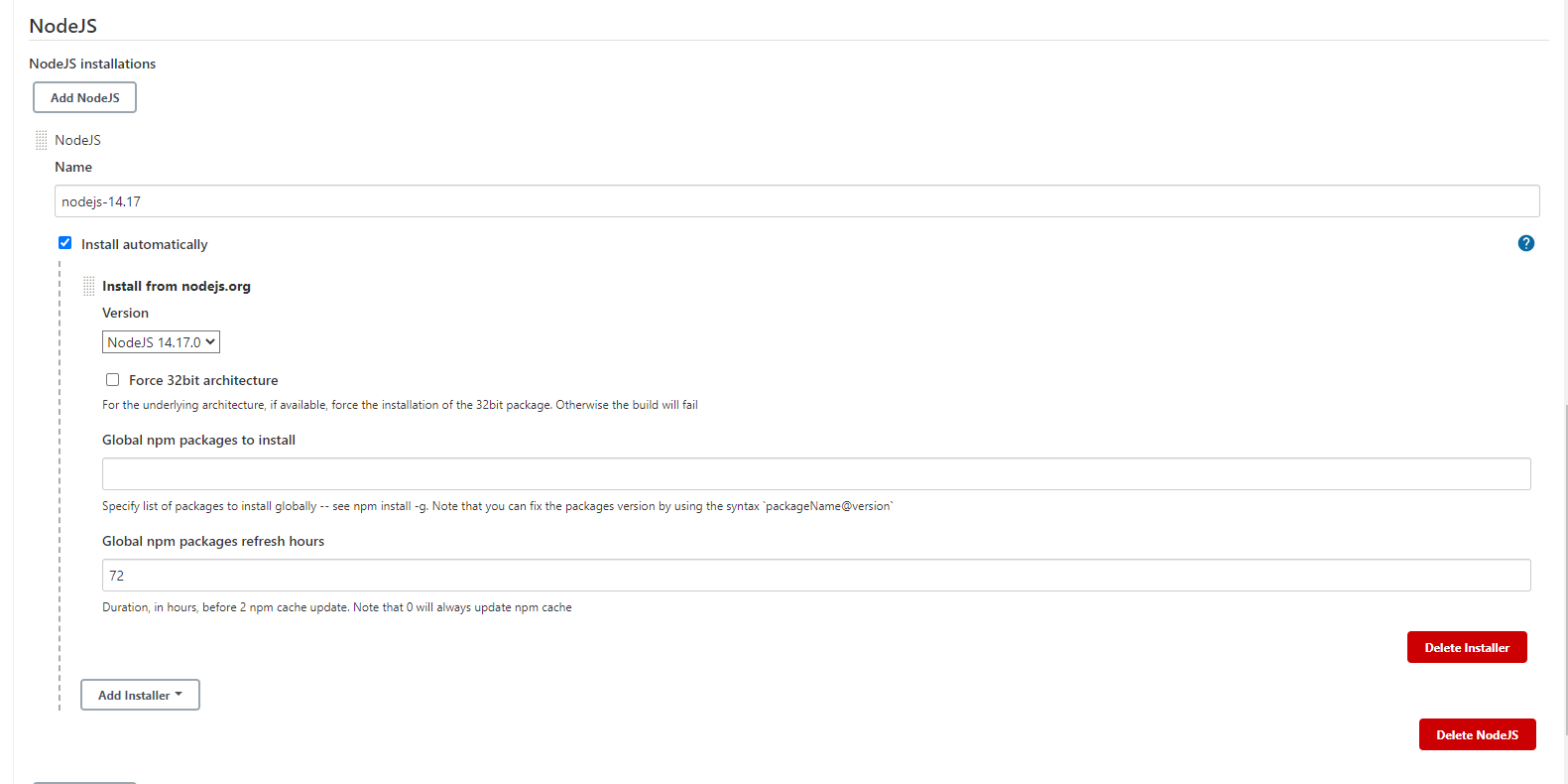
For this assignment there are two general solutions, one with node.js plugin and one with using a docker image with node and chrome. For the first solution (with plugin) no extra installation of chrome is necessary, since this is already included in the Jenkins-agent.

Answer - Node.js plugin**:**

* Install node.js plugin without restart



* Go to “Global Tool Configuration under “Manage Jenkins” to set a node.js version



The name you give this node version is the name you use in the “tools” section of the pipeline.

In the pipeline section something resembling the pipeline below should be typed:

pipeline {

agent any

tools {

nodejs "nodejs-14.17"

}

stages {

stage('check npm version') {

steps {

sh 'npm -v'

}

}

stage('Git pull') {

steps {

git branch: 'main', url: 'https://github.com/EmmyHermans/Jenkins-SIG'

}

}

stage('Install frontend dependencies') {

steps {

sh 'cd koa-angular-app/angular-frontend && npm ci'

echo "installed frontend end"

}

}

stage('Test frontend') {

steps {

sh 'cd koa-angular-app/angular-frontend && npm run test:ci'

echo "Tested frontend end"

}

}

stage('Install backend dependencies') {

steps {

dir('koa-angular-app/koa-backend') {

sh 'npm ci'

echo "installed backend end"

}

}

}

stage('Test backend') {

steps {

dir('koa-angular-app/koa-backend') {

sh 'npm test'

echo "Tested backend end"

}

}

}

stage('clean workspace') {

steps {

cleanWs()

}

}

}

}

##### Answer - Nodejs docker image:

No additional plugins need to be installed. In the pipeline section something resembling the pipeline below should be typed:

pipeline {

agent {

docker { image 'zenika/alpine-chrome:with-node' }

}

stages {

stage('check npm version') {

steps {

sh 'npm -v'

}

}

stage('Git pull') {

steps {

git branch: 'main', url: 'https://github.com/EmmyHermans/Jenkins-SIG'

}

}

stage('Install frontend dependencies') {

steps {

sh 'cd koa-angular-app/angular-frontend && npm ci'

echo "installed frontend end"

}

}

stage('Test frontend') {

steps {

sh 'cd koa-angular-app/angular-frontend && npm run test:ci'

echo "Tested frontend end"

}

}

stage('Install backend dependencies') {

steps {

dir('koa-angular-app/koa-backend') {

sh 'npm ci'

echo "installed backend end"

}

}

}

stage('Test backend') {

steps {

dir('koa-angular-app/koa-backend') {

sh 'npm test'

echo "Tested backend end"

}

}

}

stage('clean workspace') {

steps {

cleanWs()

}

}

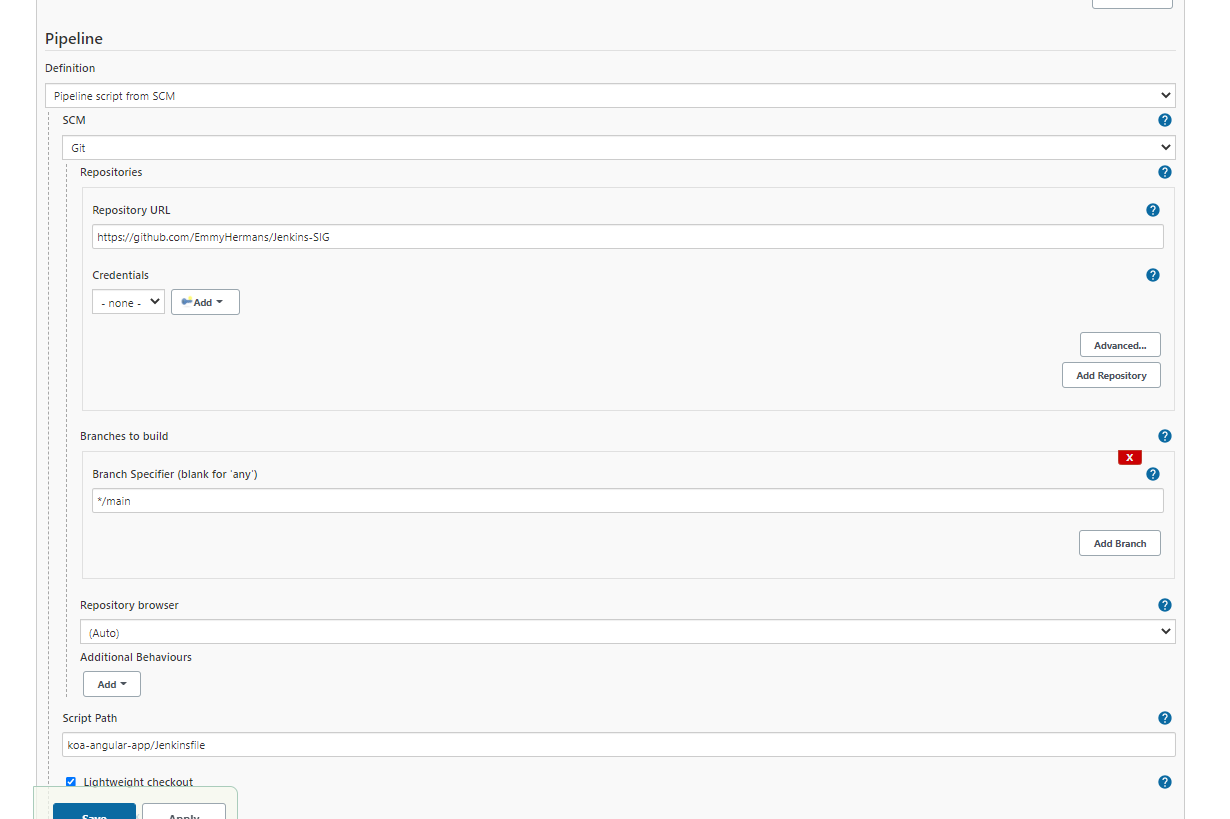
}

}

### Jenkinsfile

#### Assignment 3

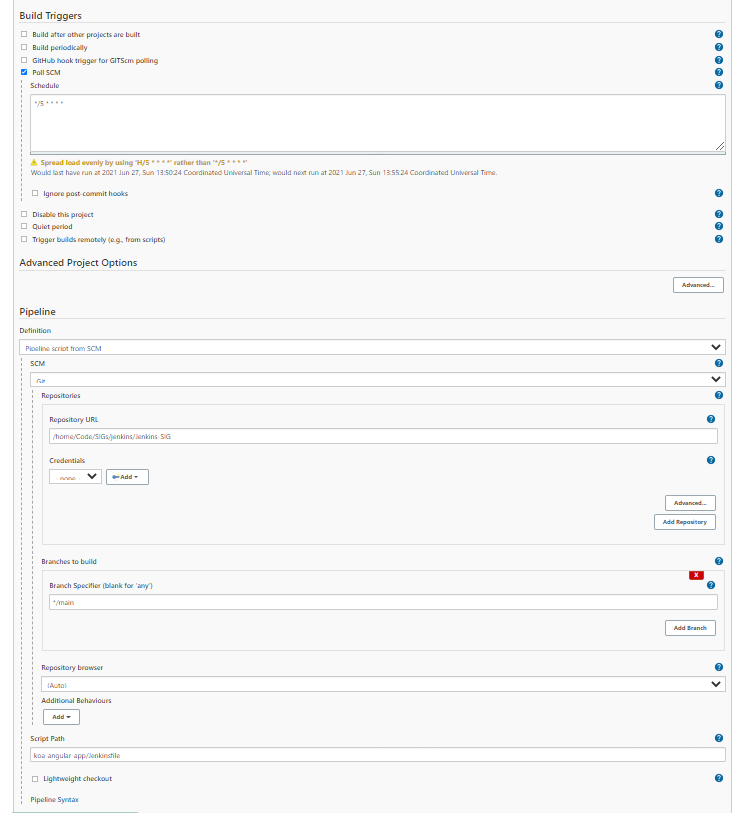
The screenshot below shows the options that need to be filled in order to run the pipeline from the Jenkinsfile on github.



#### Assignment 4

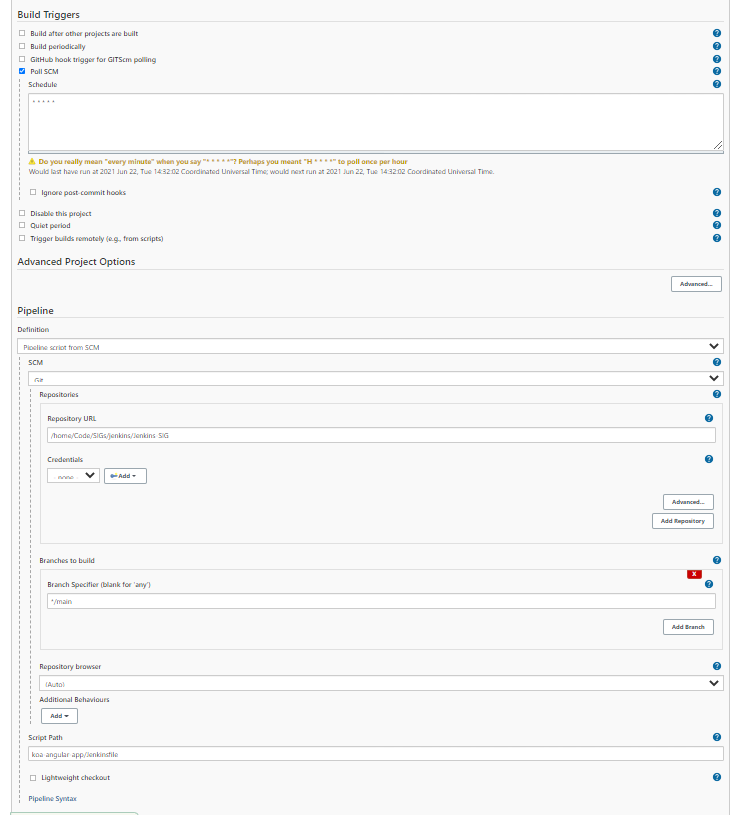
##### Answer 4a – local repository:

The screenshot below shows an example how to poll for changes in a local repository. For “Repository URL” you need to put the url to your git repository, relative from the home folder you specified in the docker compose.yml.



##### Answer 4b - github:

The screenshot below shows an example how to poll for changes in a github repository.



#### Assignment 5

See the assignment-5 folder for a possible solution. This includes the solution for the bonus assignment. If you didn’t do the bonus assignment, you can ignore the “Build” and “Deliver” stages shown in the Jenkinsfile.