

Developer Documentation

How to obtain the source code?

1. Download and install git (<http://git-scm.com/downloads>)
2. Download and install node.js (<https://nodejs.org/>)
3. Open a terminal/command prompt and navigate to a folder you would like to store the project
4. Run 'git clone <https://github.com/SharedMusic/SharedMusic>' to obtain a local copy of the repository
5. Navigate to the newly created SharedMusic directory
6. Run 'npm install' to download all project dependencies

Layout of directory structure?

bin

initializing the server

controllers

logic for handling requests

sends commands to the model to update the model's state

sends commands to its associated view to change view's presentation

documentation

user and developer documentation

node_modules

installed dependencies

models

internal data structures of the server

Room

User

RoomState

test

test related files

mocha iterates over each file in directory and runs its tests

views

view-able page related logic

Html pages

public

Javascript/CSS required for views

How to build the software?

Javascript is an interpreted languages so this is not required.

How to run test suite?

1. Navigate to root of SharedMusic project folder
2. Double check all dependencies are installed by running 'npm install'
3. If you are running a system that supports make:

Run 'make test'

Else:

Run 'chmod +x node_modules/.bin/mocha'

Run 'node node_modules/.bin/mocha'

4. Results of the SharedMusic test-suite should be output to console indicating the success/failure of individual tests.

How to run test version of website?

1. Navigate to root of SharedMusic project folder
2. Double check all dependencies are installed by running 'npm install'
3. If you are running a system that supports make:

Run 'make run'

Else:

Run 'npm start'

4. Console output should read 'listening on *:xxxx' where xxxx is a port number
5. Open your favorite browser and search for 'localhost:xxxx'
6. The home page of SharedMusic should be displayed and the site should be fully functional for any further user testing.

How to set up an automated daily build and test?

Instead of listing all of the steps of setting up the nightly build that triggers the tests to run since they are long and include code examples, they can be found here <https://circleci.com/docs/nightly-builds>. We have CircleCI setup to run tests each time a push gets made to the github repo.

How to release a new version of your software?

Updating revision numbers in your code and documentation.

1. Open package.json in SharedMusic root directory and update version field appropriately.
2. Update all pertinent documents in documentation to the correct version.

Creating the zip file that a user or developer will download to obtain your product.

Not applicable to our project because our product is a website. If the user wishes they can clone the website or 'download zip' (<https://github.com/SharedMusic/SharedMusic>).

Copying files to your website so that the new version is visible to the world.

A simple command: `git push heroku master` — they must first be added permissions as a collaborator on heroku.

In order to run that command, there is some setup involved:

1. The heroku app owner must add you as a collaborator via the command—
`heroku sharing:add joe@example.com`
2. Then follow the instructions here for downloading the heroku toolbelt and cloning the heroku app
<https://devcenter.heroku.com/articles/collab>

Sanity-checking the website

Team manager is responsible for running all tests and doing a quick run through of basic functionality of the site before deploying a new release.

How to access the list of outstanding bugs and the list of resolved bugs, and how to resolve a bug.

Any issues can be filed/handled via the project's github repository (<https://github.com/SharedMusic/SharedMusic/issues>).