

# React JS

Integration with Liferay

DXP Using Headless API



## Things To Note



The purpose of this documentation is to show how users can use a popular Javascript library called React.js with Liferay and Headless API to create stand-alone React applications.

---

1 Primary requirements include knowing a bit of JavaScript, how to set up/download software onto Terminal for MacOSX. Also, knowing how to initialize a Liferay instance locally will help with the set up portion of this walk through.

---

2 For this project, I developed on a MacOSX (Linux) system, using VS Code as a barebones code editor and GitHub to push my files onto the internet for public access.



---

3 This documentation is set up into separate parts, divided by column. These include Getting Started, Headless API's Into React, and Component Explanations.



# Getting Started



---

**<https://github.com/sasajallana/react-for-liferay-fe/tree/master>**  
For reference if you're ever stuck

---

# NPM

NPM is the orchestration tool used to create and manage React Apps. Make sure you have these latest versions (as of 2/27/2020), which work with Liferay DXP 7.2.

npm/npx

6.13.6

---

java sdk

openjdk version "1.8.0\_232"

---

node

v13.7.0

---

# Liferay and Blade

Liferay is being used as a platform management system in this project. It provides the Headless API that our stand-alone React application(s) call in order to retrieve data.

Blade is the CLI tool used to initialize a Liferay project with the correct files.

This tutorial uses 7.2 DXP with Tomcat.



## Setting Up



One thing that you should have already is **Homebrew**, a MacOSX package installer. Using Homebrew makes it easier to download software using Terminal:  
<https://docs.brew.sh/Installation>

Before beginning to setup ReactJS for Liferay Workspace, you will need a few things. As mentioned above, you should first download things like Npm/npx, node, Java. On MacOSX, the easiest way to download is by typing these commands in \$USER home on a Terminal window...

node

brew install node

npm/npx

(will be installed w node, check via)  
npm --version

And then, you're going to need to download Liferay Workspace:

<https://sourceforge.net/projects/lportal/files/Liferay%20IDE/3.8.0/>

LiferayWorkspace-202002250521-osx-installer.dmg

2020-02-26

22.6 MB

Click on this line from the above link and wait for the download. Once done, go through the process of setting up from this site:

[https://portal.liferay.dev/docs/7-1/tutorials/-/knowledge\\_base/t/liferay-workspace](https://portal.liferay.dev/docs/7-1/tutorials/-/knowledge_base/t/liferay-workspace)

**Note:** If you try to open the Installer directly, it might issue a 'Untrusted Developer' warning. The easiest work around is to right-click the Installer icon and from the Menu, click Open.



## Why We Use SDK from DMG



We are using the SDK installer to initialize Liferay and Blade CLI. From now on, we will use Blade and npm to create Liferay Workspaces to host our React App for development.



## Setting Up Workspace



Now with Node, Java, NPM installed and the Liferay installer done, let's start from anew.

Go ahead and open your Terminal to your \$USER home directory.  
Let's create a new folder where your Liferay Workspace will live and cd into it.

**cmd**      `mkdir react-for-liferay-fe`

**cmd**      `cd react-for-liferay-fe`

Now, in your new empty folder "**react-for-liferay-fe**", initialize a Liferay workspace through:

**cmd**      `blade init -v 7.2`

The **-v** tag indicates what version of Liferay we're using. Now if you list the files in the directory (**ls** command), you will see this:

### Terminal

```
Setups-MacBook-Pro:react-for-liferay-fe setup$ blade init -v 7.2
Setups-MacBook-Pro:react-for-liferay-fe setup$ ls
build.gradle           gradle           gradle.properties      gradlew.bat          settings.gradle      wars
configs                gradle-local.properties  gradlew             modules            themes
bundles               gradle           gradle.properties      gradlew.bat        modules           wars
```

These are all Liferay Workspace files. Next, we need Liferay Server files, in order to run and setup the rest of Liferay. Still in the **react-for-liferay-fe** directory, run command

**cmd**      `blade server init`

### Terminal

```
Setups-MacBook-Pro:react-for-liferay-fe setup$ blade server init
Executing gradle task initBundle...

server init completed successfully.
Setups-MacBook-Pro:react-for-liferay-fe setup$ ls
build.gradle           configs           gradle-local.properties gradlew           modules           themes
bundles               gradle           gradle.properties      gradlew.bat       settings.gradle   wars
```

This will create a **bundles** folder. In the new bundles folder, there will be a bunch of server files. To access the command to run Liferay Workspace, you will need to execute this command to move into the correct folder:

**cmd**      `cd bundles/tomcat-9.0.17/bin`

### Terminal

```
/Users/setup/react-for-liferay-fe/bundles/tomcat-9.0.17/bin
Setups-MacBook-Pro:bin setup$ ls
bootstrap.jar           configtest.bat      setclasspath.sh      tomcat-native.tar.gz
catalina-tasks.xml      configtest.sh       setenv.bat          tool-wrapper.bat
catalina.bat            daemon.sh          setenv.sh           tool-wrapper.sh
catalina.sh             digest.bat         shutdown.bat       version.bat
ciphers.bat            digest.sh          shutdown.sh         version.sh
ciphers.sh             makebase.bat      startup.bat
commons-daemon-native.tar.gz  makebase.sh      startup.sh
commons-daemon.jar      setclasspath.bat  tomcat-juli.jar
```

You will notice a bunch of .sh and .bat files. These are executionary files that can be launched or stopped by using commands like "**run**" and "**stop**". In this case, we will be setting up Liferay Workspace for the first time, so we need to *run* the server. You do this by

running the **catalina.sh** file with the command:

**cmd**      `./catalina.sh run`

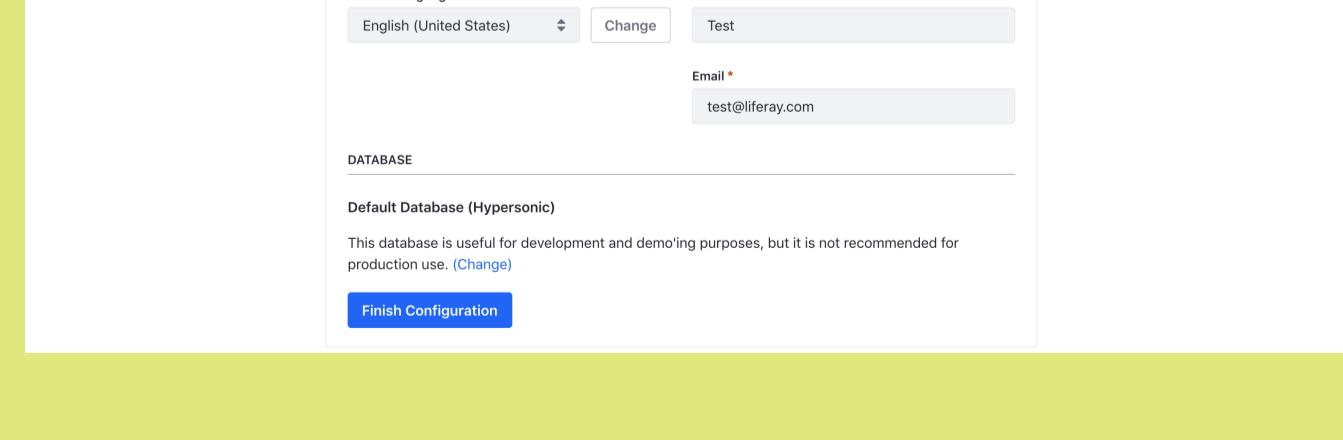
Use this command anytime you would like to run the Liferay Workspace. Since we are going to be consistently building off data from the Liferay Workspace, the server will need to be up. If nothing goes wrong, in the Terminal the below Liferay logo and extra lines of code should appear, and then a browser window will open connected to web address

**localhost:8080**

### Terminal

```
Starting Liferay Community Edition Portal 7.2.0 CE GA1 (Mueller / Build 7200 / June 4, 2019)
```

**localhost:8080**



Continue set-up as usual with **Default Configuration** and **Default Database** as Hypersonic database, and make sure to create the **Admin user as the initial test@liferay**, just incase you ever get locked out somehow. **Agree to the Terms of Service** and set your password.

You should have a fresh, working Liferay Instance working!



## Setting Up React



To start integrating React with Liferay, open up Terminal and redirect into the workspace directory created before that hosts all of the server and Liferay files (if following the tutorial, **react-for-liferay-fe**).

Assuming that you have npm/node installed, we will now be able to use **npm/npx** commands in order to build our ReactJS application.

One warning/issue you might bump into is the **mismatch of compatible Node.js version to work with Liferay**. As of **3/3/2020**, the latest version of Node.js that works with Liferay DXP 7.2 is **Dubnium**. You can change your Node version via **nvm**.

From the **react-for-liferay-fe** directory in your Terminal, type the following command:

```
npx create-react-app reactapp
```

**npx**: is like npm, but runs the following command temporarily, and is best in this case to just instantiate a new React application in the local **react-for-liferay-fe** directory

**create-react-app**: is the command to build React apps/generate fresh React application files

**reactapp**: is the name of our new React application

After the command runs and says it's creating and finishes installing a new React project called **reactapp**, you should get a few "Success" messages, and recommended commands to run in order to start the project. Go ahead and **cd** into **reactapp** and **ls**.

```
cd reactapp ; ls
```

```
[Setups-MacBook-Pro:reactapp setup$ pwd  
/Users/setup/react-for-liferay-fe/reactapp  
[Setups-MacBook-Pro:reactapp setup$ ls  
README.md          package-lock.json      public  
node_modules       package.json          src
```

**node\_modules**: where external dependencies/files needed for the project as you develop will be stored and referenced

**package.json**: metadata for the project, including dependency version numbers, script commands

**public**: on initialization, contains barebones files that renders the homepage (**index.html**) and other files

**src**: contains JavaScript and style files that helps to render the homepage programatically

You don't have to exactly know everything about React from the get-go; you will learn as you work with it continuously. Anyways, let's go ahead and start up the application with command:

```
npm start
```



If successful, the app will launch in browser on **localhost:3000**. You have successfully created a work React App in a Liferay Workspace directory!