

OR2018 DSpace Angular Workshop

- Slides : <https://tinyurl.com/or2018-dspace-ui>
- Learning Objectives
- Preparing for the workshop
 - Optional Installation
 - Installing the Angular UI locally
 - Creating a branch for the exercises
 - Optional Development Tools
- Solution tags
- Additional Resources

This page is for attendees of the OR2018 workshop "Getting Started with the DSpace 7 Angular UI". It provides links to resources, workshop preparation, etc. 

(This workshop was based on materials from the [OR2017 DSpace Angular Workshop](#), but slides/materials/exercises have been updated.)

Slides : <https://tinyurl.com/or2018-dspace-ui>

Learning Objectives

- Learn about the Angular framework, and its main elements (templates, components, services, modules)
- Learn about the TypeScript language and its syntax
- Learn how code is structured in the DSpace Angular UI codebase & current project status
- Learn the basics of how to currently "brand" or "theme" the Angular UI
- Learn about coding in the DSpace Angular UI (creating new pages / adding functionality, etc)

Preparing for the workshop

There are no installation requirements for the workshop. However, as the second half of the workshop is a code walkthrough (of some live development / theme customizations), you can optionally choose to install the Angular UI locally (and perform "hands-on" changes on your laptop).

Please be aware that, as we currently have over 40 workshop attendees, we unfortunately cannot promise any "hands-on" support on an individual basis. But, we still welcome you to turn use this as an opportunity to try some "hands-on" changes.

Optional Installation

During the code walkthrough, we will be working directly with the [dspace-angular](#) project (this is the in-development Angular UI for DSpace 7). However, don't worry. It's rather easy to install, and you **do not need to install DSpace itself** (as we will be using the [public, demo REST API](#) as our backend).

Installing the Angular UI locally

(These instructions are based on the [dspace-angular README](#))

- **Git** : You'll want to have Git to simply clone and install our [dspace-angular](#) project (this is the in-development Angular UI for DSpace 7). If you don't have Git, you can choose to install either [GitHub Desktop](#) (which includes a GUI) or [just plain Git](#). Any version of Git is fine.
- **Node.js** (version 6.x or 8.x recommended) : Required to build/run the Angular UI.
 - Note: Node.js will also automatically install [npm \(Node Package Manager\)](#). You shouldn't need to install that separately
- **Yarn** (version 1.3.x or above) : This is an easier to use, speedier package manager that works similar to npm (Node's package manager). It's currently required to build Angular UI.
- **dspace-angular** : You can now install [dspace-angular](#) itself using the command line.
 1. `git clone https://github.com/DSpace/dspace-angular.git` (Pulls down the code into a dspace-angular folder on your local computer)
 2. `cd dspace-angular` (Move into the newly created dspace-angular folder)
 3. `yarn install` (Install all local dependencies via yarn)
 4. `yarn run watch` (Build and startup the DSpace Angular UI, while watching code for new changes.)
 - a. Alternatively, you can also run "`yarn start`", but that command requires you restart to pick up local code changes.
 5. Assuming all went well, you should be able to go to <http://localhost:3000> in your web browser and see the Angular UI homepage.
- Now, to easily follow along with the workshop code changes step-by-step, you may wish to [create a branch for the workshop exercises](#)

That's it! Again, there is *no need* to have DSpace running or install a database, Java, Tomcat or any of the other usual DSpace prerequisites.

Creating a branch for the exercises

To follow along with the exercises run the following git commands in the project directory to get the correct starting point

1. `git remote add workshop https://github.com/DSpace-Labs/dspace-angular-workshops.git`
2. `git fetch --tags workshop` (Pulls down the dspace-angular-workshops code locally, including tags)
3. `git checkout or2018-start` (Checks out the starting point of the workshop, the "or2018-start" tag)

4. `git checkout -b or2018-hands-on` (Creates a local branch named "or2018-hands-on" for you, based on that "or2018-start" tag)

At certain points during the hands-on part we'll show a git tag that corresponds to the solution so far. If you'd like to sync up to that point, run the following:

1. Revert all existing files to their state at the start:
 - `git reset or2018-start --hard`
2. Remove any newly created files
 - `git clean -f -d`
3. Sync with the solution
 - `git merge $TAG_NAME`

Optional Development Tools

While not required, the following tools may make it easier to do basic development with the Angular UI.

- **Chrome web browser** : When developing an Angular application, all web browsers are *not* created equal. While you obviously should use other browsers to help test your application, Chrome has built in debugging/visualization tools that come in handy when doing Angular development. **Make sure you disable cache in your network tab and keep the inspect element open.**
- **An IDE** : If you wish to do much Angular development, you'll likely want an IDE (Integrated Development Environment) or an editor that understands the [TypeScript language](#). Here's a few we prefer. **Just choose ONE.** *If you have no strong preference, you may want to start with Visual Studio Code.*
 - **Visual Studio Code (Free)** : This is a TypeScript IDE from Microsoft (the makers of TypeScript). It's frequently used for Angular development, even by some of the main Angular developers, and provides an excellent [debugger for Chrome](#) (which you may also wish to install for easier debugging)
 - **IntelliJ (Requires a license)** : If you use IntelliJ already for other development work, [IntelliJ IDEA Ultimate](#) has TypeScript support. You could also use [WebStorm](#), IntelliJ's Javascript IDE
 - **Atom.io (Free)** : This is an advanced text editor from the makers of GitHub. If you install the [Typescript language plugin](#), it acts a little more like a basic IDE for TypeScript / Angular (providing autocomplete and highlighting). However, be forewarned that it doesn't have the debugging capabilities of Visual Studio Code or IntelliJ.
 - Any other editor or IDE that provides either TypeScript or Angular 2 support.

Solution tags

Click on the commit hashes for each of the tags in the "releases" to see all code changes for each of the steps of the workshop:

<https://github.com/DSpace-Labs/dspace-angular-workshops/releases>

Additional Resources

- [DSpace 7 UI Technology Stack](#) : Technology overview with link to tutorials on Angular, etc
- [DSpace 7 - Angular UI Development](#) : How to contribute to Angular UI