**ILP-003 – Output and Code Submission details**

**Always refer to Microsoft Teams for the latest version of the document.**

**Please read the complete document properly and follow the instructions exactly. Read all the 4 pages of the document.**

The client side application written in Angular will interface to the server using HTTP protocol to obtain data from the Express server or save the data on to the Express server. The data required for Issues should be managed in a JSON memory object (OR) in a JSON file on Express server.

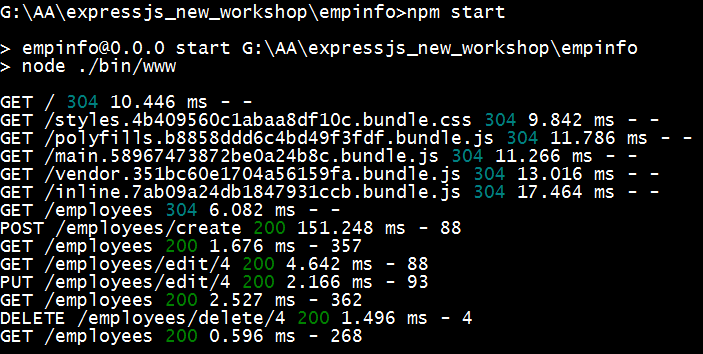
The implementation will be similar to 16-httpservice and empinfo-app demo code in Angular. (Refer 16-httpservice demo and empinfo-app demo in Angular Webex session).

**Output Screenshot**

After you have implemented the Integrated Project ILP-003, run the application and take a screenshot image of the following screens.

1. View Issues (Main Screen) displaying all columns
2. Add Issue Form Screen
3. Edit Issue Form Screen
4. Delete Issue Screen - A screen after deleting the issue
5. After taking screenshots of the above 4 screens, go to Command Prompt window where "npm start" was executed and take a screenshot 🡸 **VERY IMPORTANT TO INCLUDE THIS SCREENSHOT**
6. View Issues displaying a subset of the columns (not all)
7. View Issues providing some search filter and displaying the filtered rows

Your screenshot for **“npm start”** should be similar to below screenshot. It should show all the operations done – View, Add, Edit and Delete and **the Angular bundles getting transferred**.

****

**Code Submission to Gitlab**

**Important Note:**

1. Take a **backup** of the source code of ILP-001 and ILP-002 project and all the subsequent ILP projects and store it safely in your local system.

**Reason:** Some employees are still not comfortable using Git and making lot of mistakes in using it, hence take a backup so that your work is not lost.

1. In **ILP-001**, some of the employees **had not followed the instructions** and checked in code in many different ways as mentioned below.
2. Added the zip file of the source code.
3. Added only the app folder and sub-folders and not the entire Angular project
4. Added the Angular project directly under “fullstack-modernweb-ilp-capstone-projects” instead of adding under “fullstack-modernweb-ilp-capstone-projects/integrated-learning-projects/ilp-001”.

**For ILP-003, please follow the below instructions exactly as mentioned below.**

**Source Code structure:**

**Server Side Implementation:**

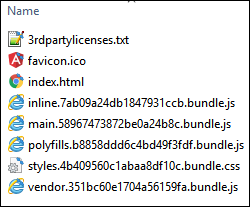
If you have generated the project using “**express issuetracker**”, a folder “issuetracker” will be the top level source folder for your Express application.

**Client Side Implementation:**

If you have generated the project using **“ng new issuetracker”,** a folder “issuetracker” will be the top level source folder for your Angular application.

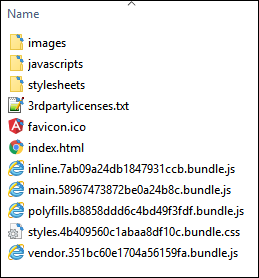
A production version of the Angular application need to be generated by executing the command **“ng build –prod”** in the root directory of the Angular application [ i.e. in the “issuetracker” folder which contains package.json ]. A folder “**dist”** will get generated in which files similar to the screenshot shown below will get generated.

**“dist” folder of Angular application**



Copy all the files in the **“dist”** folder of Angular application (**only the files and not the folder “dist”**) to the **“issuetracker/public”** folder of Express application. After copying, your public folder of Express application should appear similar to the screenshot as shown below.

**“public” folder of Express app after copying files from “dist” folder of Angular app**



Before checking in to Gitlab, copy the entire source structure **of Express Application** to your local Git repository starting from the top level folder **“issuetracker”** (including “issuetracker”) **excluding “node\_modules”** under **“fullstack-modernweb-ilp-capstone-projects/integrated-learning-projects/ilp-003”.**

Create a folder **“source\_angular”** under **“ilp-003”** and copy the entire source structure of Angular project starting from the top level folder **“issuetracker”** (including “issuetracker”) **excluding “node\_modules” and “dist” folders. 🡸** This step is only to transfer your source code of client side implementation into Gitlab repository. The integrated application [ Express + Angular ] which can be executed will be present under **“ilp-003/issuetracker”**.

**VERY IMPORTANT:** Do not check in “node\_modules” folder, “build” folder, “dist” folder, or any build generated folders. Specify all these folders or files in **.gitignore** in the project root directory of your repository.

**Output screenshots:**

Screenshots of the output should be placed under the **“output”** folder under **“fullstack-modernweb-ilp-capstone-projects/ integrated-learning-projects/ilp-003”**.

**How your folder structure should be in your local Git repository before checking in:**

**Note:** For instructions about how to check in to Gitlab, please refer the document **GitlabCodeSubmission\_Instructions\_v?.?.docx** uploaded in Microsoft Teams under **"??\_ModernWebFullStack\_Group/General/Files"**

In your local Git repository, your ILP-003 issuetracker project source [ **Express app along with files of “dist” folder of Angular app**], the output screenshots and the source code of Angular project should be organized as shown below before checking in.

**Note:** The screenshot below shows just a high level folder structure but in your ILP project, you may also have sub folders under some of the folders. Also, the **“public”** folder shown below will contain the files copied from the **“dist”** folder of Angular application as explained earlier.

