**Creating a Custom Plugin**

1. **Command to Generate the Plugin :-**
   1. The first step is to create the custom plugin using a command. This command generates the necessary files and folder structure for your plugin.
   2. Run the following command in your terminal:

node script/generate\_plugin --name Asma -y

**Explanation of the command:**

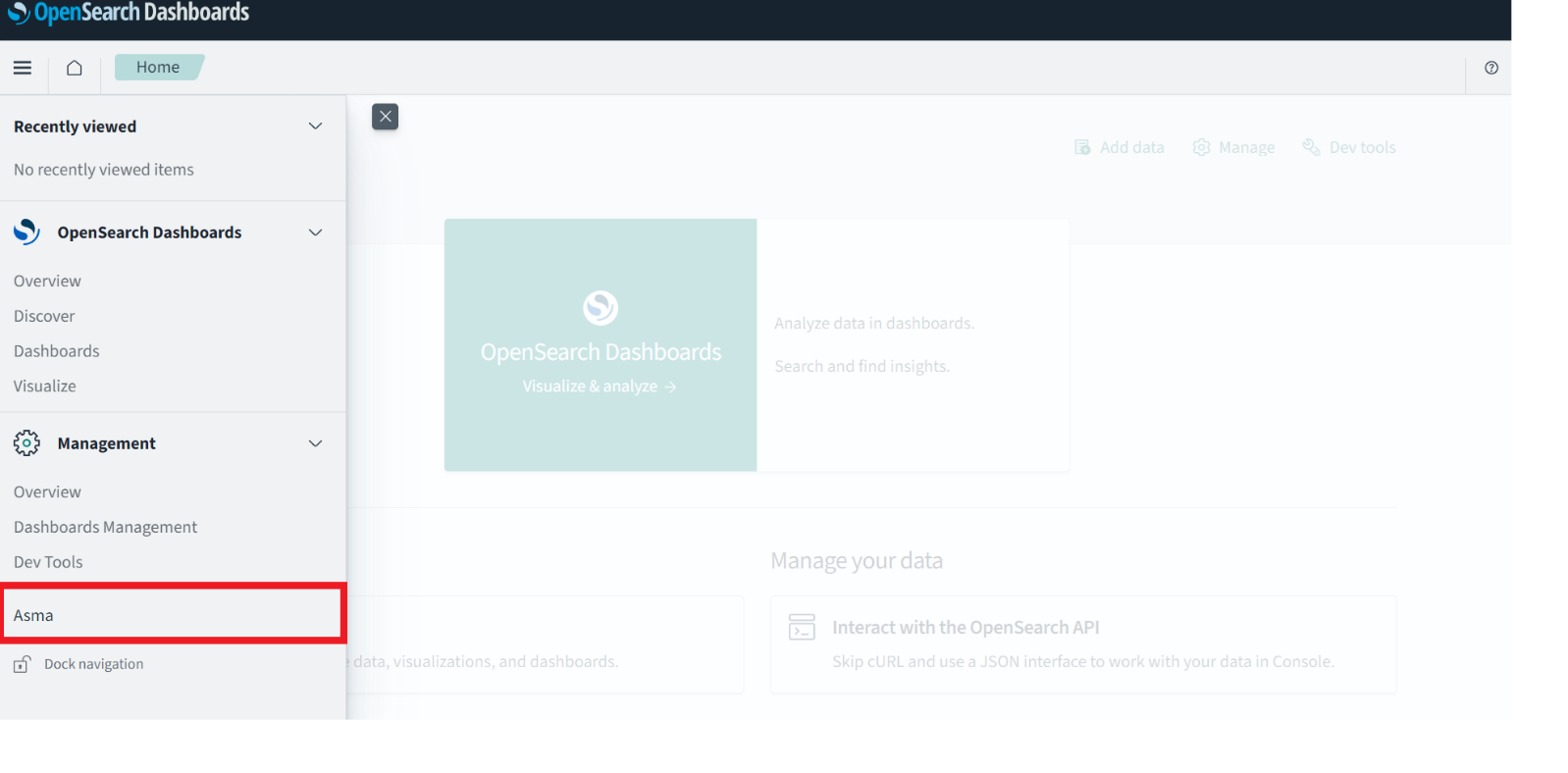
* **node scripts/generate\_plugin:** This runs a Node.js script to generate the plugin.
* **--name Asma:** The name of your plugin will be "Asma."
* **-y**: Automatically accepts any prompts and runs the script without asking for confirmation.

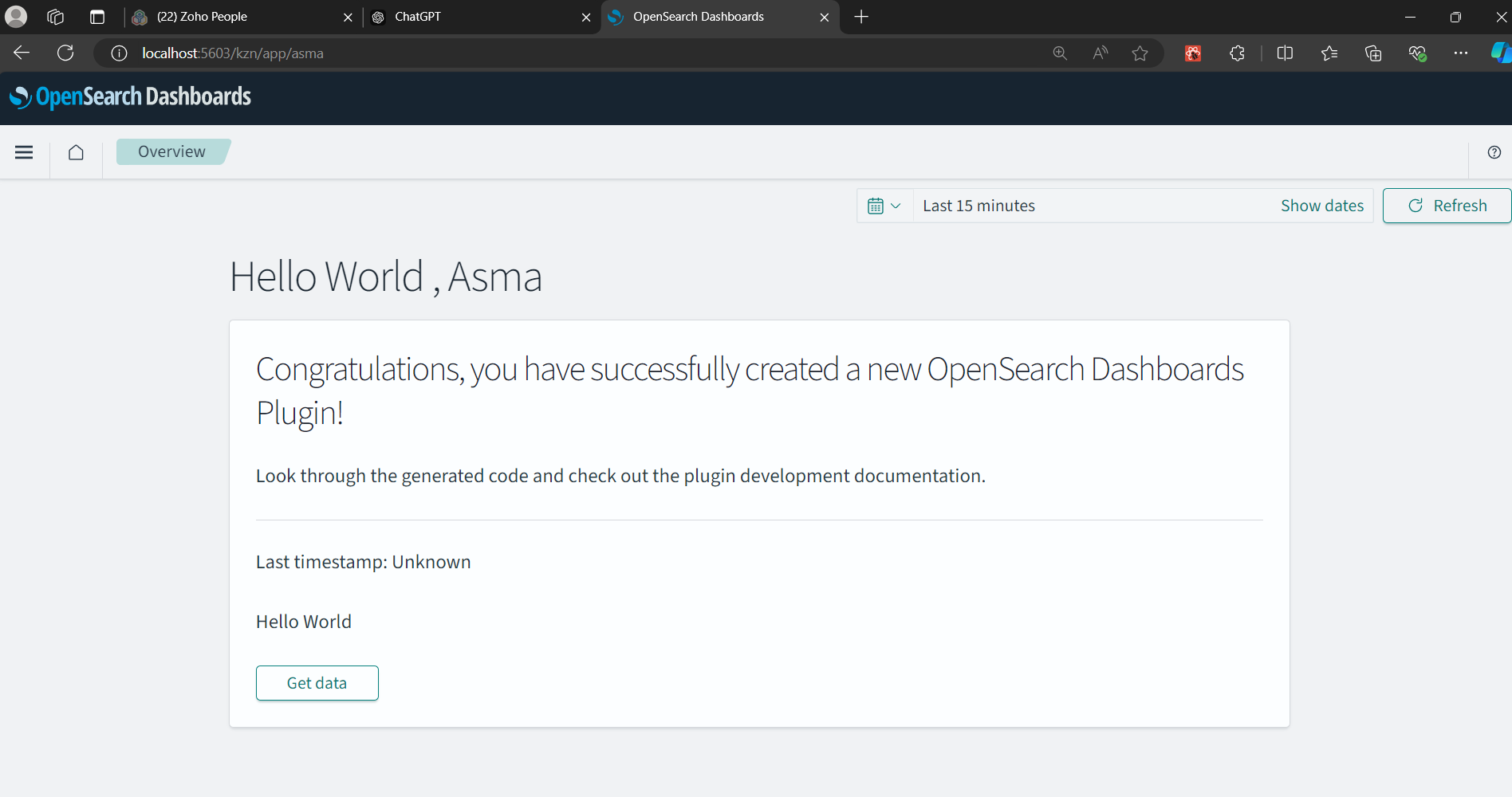
This command will generate a folder structure, including the necessary files to get started with the plugin.

**2.do the code in app.tsx**

****

**3.output**

****

****

**Main code inside app.tsx (add hello wolrd inside this app).**

**File structure**

**1.application.txs**

**Purpose**: This file handles the rendering of the React application when the plugin is mounted within the OpenSearch Dashboards environment. The renderApp function takes in several parameters (such as notifications, HTTP, and navigation) and mounts the AsmaApp component into the DOM.

**Key Points**:

* It uses ReactDOM.render to render the main React component AsmaApp to the DOM.
* It passes various services (like notifications, http, navigation) and app-related parameters (like basename and element) to the AsmaApp component.
* The return value of renderApp is a cleanup function (unmountComponentAtNode) to unmount the component when the app is no longer needed.

### 2. index.ts

* **Purpose**: This file sets up the plugin and exports necessary static code and types. It initializes the AsmaPlugin class, which is the core of the plugin.
* **Key Points**:
  + The plugin function returns an instance of AsmaPlugin.
  + This file also exports types such as AsmaPluginSetup and AsmaPluginStart for use in other parts of the application.

### 3. plugin.ts

* **Purpose**: This is the main plugin file that defines the behavior of the AsmaPlugin class. It implements the OpenSearch Dashboards Plugin interface and provides setup and start methods, which are crucial for plugin initialization and lifecycle management.
* **Key Points**:
  + **setup method**: Registers the plugin in the OpenSearch Dashboards system. It also configures the application to be added to the navigation and renders the React app using renderApp.
    - The app is loaded asynchronously when the mount function is called.
    - The getStartServices method is used to obtain core services that are needed during the plugin’s runtime.
  + **start method**: Defines the behavior when the plugin starts. In this case, it returns an empty object.
  + **stop method**: Placeholder for any necessary cleanup when the plugin is stopped.

### 4. types.ts

* **Purpose**: This file defines the TypeScript interfaces used throughout the plugin, especially for interacting with OpenSearch Dashboards services and other plugins.
* **Key Points**:
  + **AsmaPluginSetup interface**: Defines methods or properties that will be available during the plugin setup phase (e.g., getGreeting).
  + **AsmaPluginStart interface**: Currently an empty interface, which could later be used to define properties or methods available when the plugin starts.
  + **AppPluginStartDependencies interface**: Specifies dependencies required for the plugin to run, such as the NavigationPublicPluginStart type, which is likely related to OpenSearch Dashboards' navigation system.