**Component Name**: Push Notification

**Component Functionality**:

Uses service worker Angular service and a .Net Core backend service to send-receive the push notifications. The notification channels are subscribed to from the front-end Angular application. The given channels are then broadcasted with required notifications (controlled via any user defined custom logic) from the backend .Net Core server.

**Usage Areas:**

Can be distributed as shared Angular module for apps needing push notification service where notifications are broadcasted from backend .Net Core server. The broadcast logic and contents of notifications themselves can be controlled from any custom logic written in the backend server side.

**Component Technical details:**

* Angular cli version: 9.0.7
* Typescript version: 3.7.5
* rxjs version: 6.5.4
* DotNet Core: 2.1

There are 2 components in the current app.

One is Notification Component. It uses Notification Service of service workers, which is something that is going to run in the background even if you do not have the website open. Tested on Chrome & Edge browser.

Second is the Broadcast Component from which we can send notification, which include Title of Notification, message of notification and the URL of website.

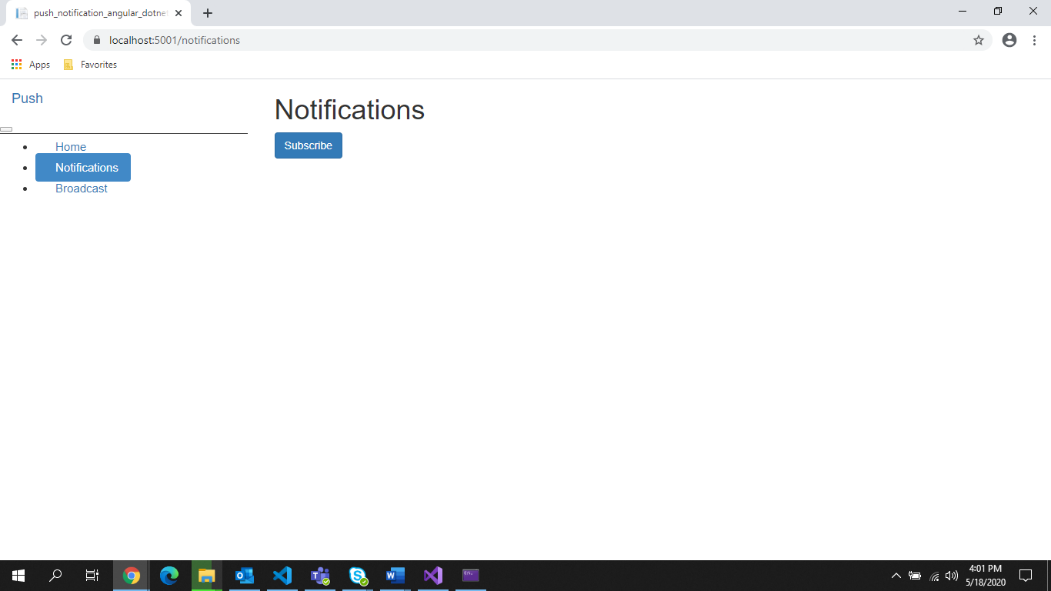
There is also a sw.js file which is a service worker file in JavaScript through which our app sends a notification.

**How this component can be integrated with other Applications:**

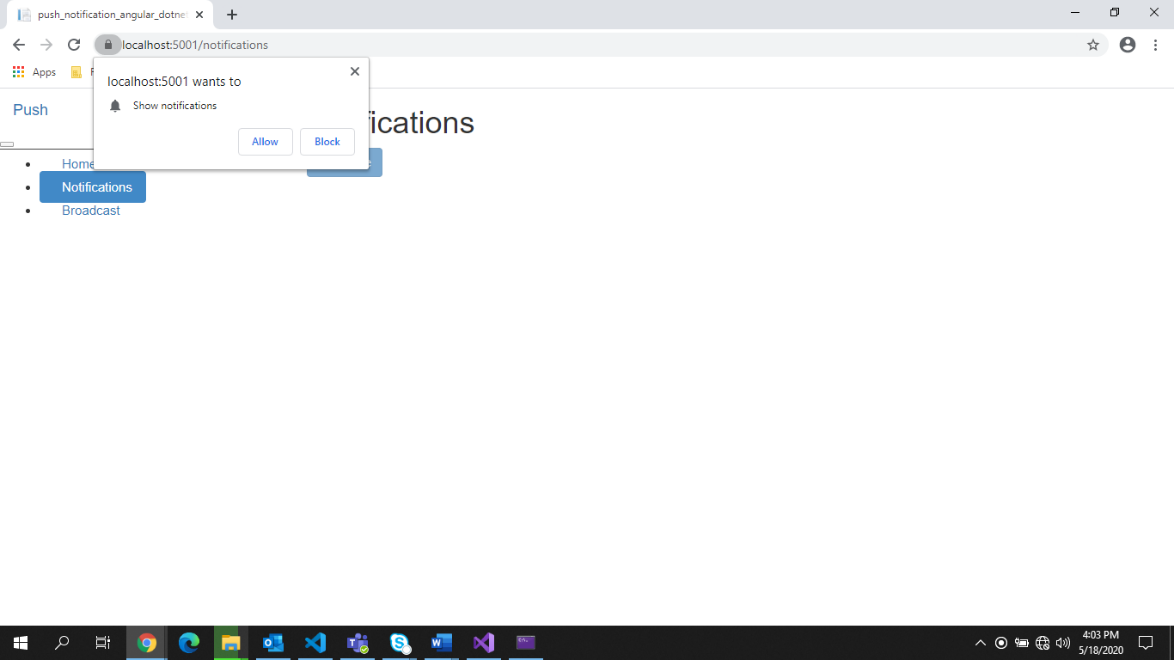
* We need to have a service file through which we can connect to our backend server. In this file we are going to create an object to hold the state of push notification of the browser instance. The app.component will call the init method as soon as the app starts which is going to check if the current browser supports service workers and push notifications, if it does then we're going to automatically register the service worker (sw.js) we added to our assets folder.
* We need a public and private key so all the communication between the sender and the receiver is encrypted. We are going to add our Public key to the environment.ts file of angular code & on the server side, we are going to add our Public and Private keys to the appsettings.json
* Also, we need to copy the generated.ts to the same level, where service file is.
* To send the push notifications, we are going to use a nuget package called WebPush. We have also configured in the startup to inject the VapidDetails which are the details we need to send the push notification: Subject (Origin URL), Public and Private keys.
* Sw.js should be copied to assets folder of our angular project. In this file we are going to add an event listener for the push event.

**Screenshots:**

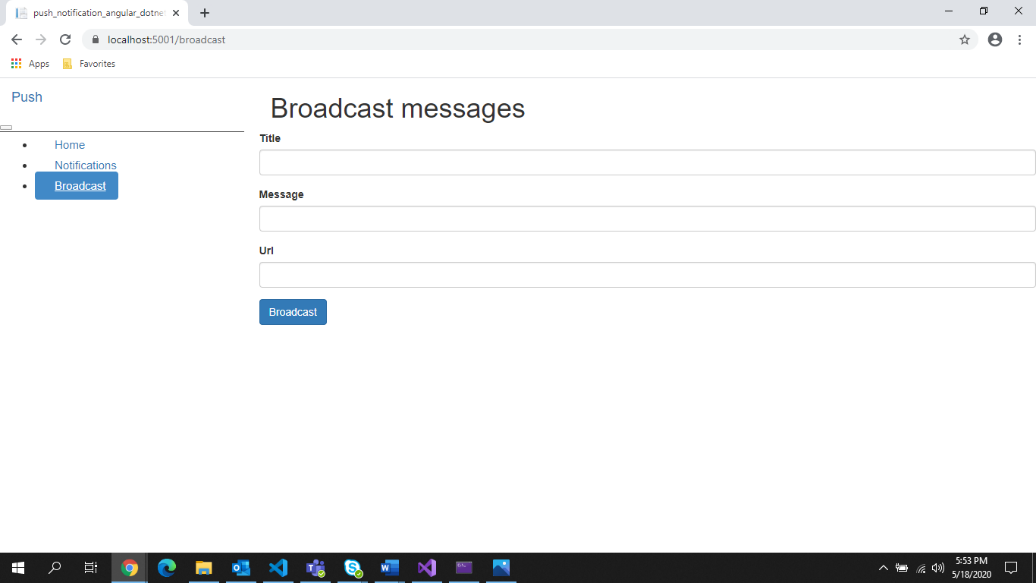
1. Initial screen:



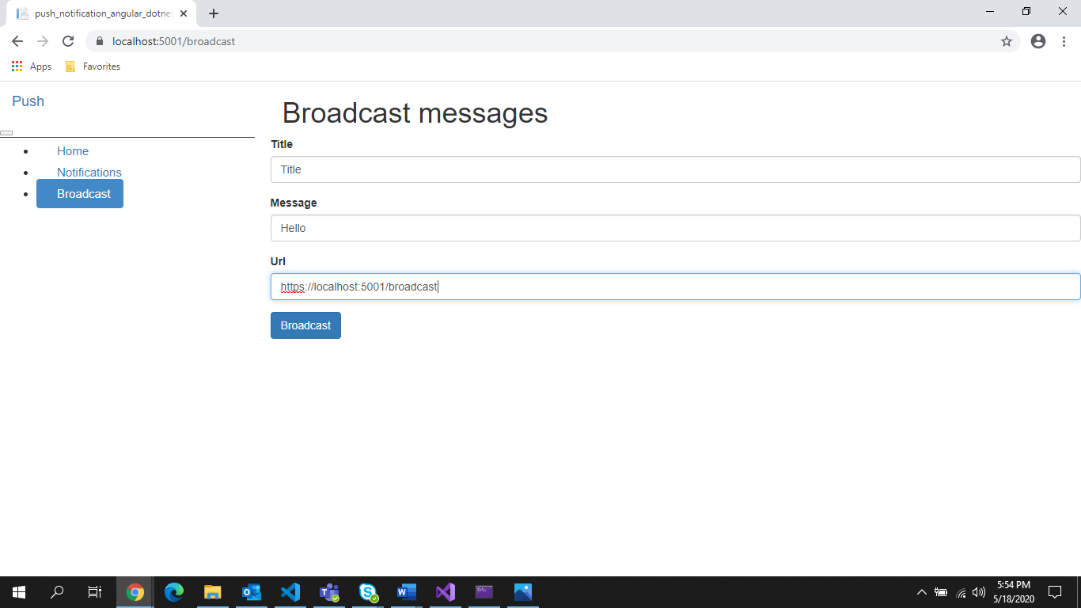
1. User Click on subscribe button, they get pop up from there browser to allow or block notification.



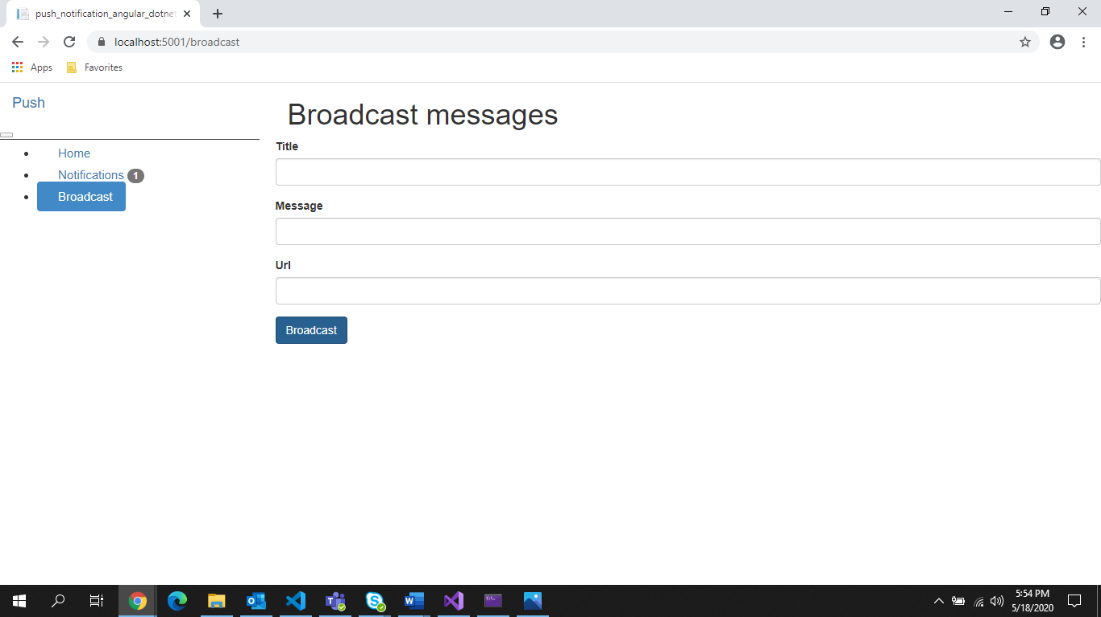
1. To broadcast a notification



1. Type the title, message, and URL then click broadcast



1. We will receive a notification like this



1. We can view our notification in notification tab

