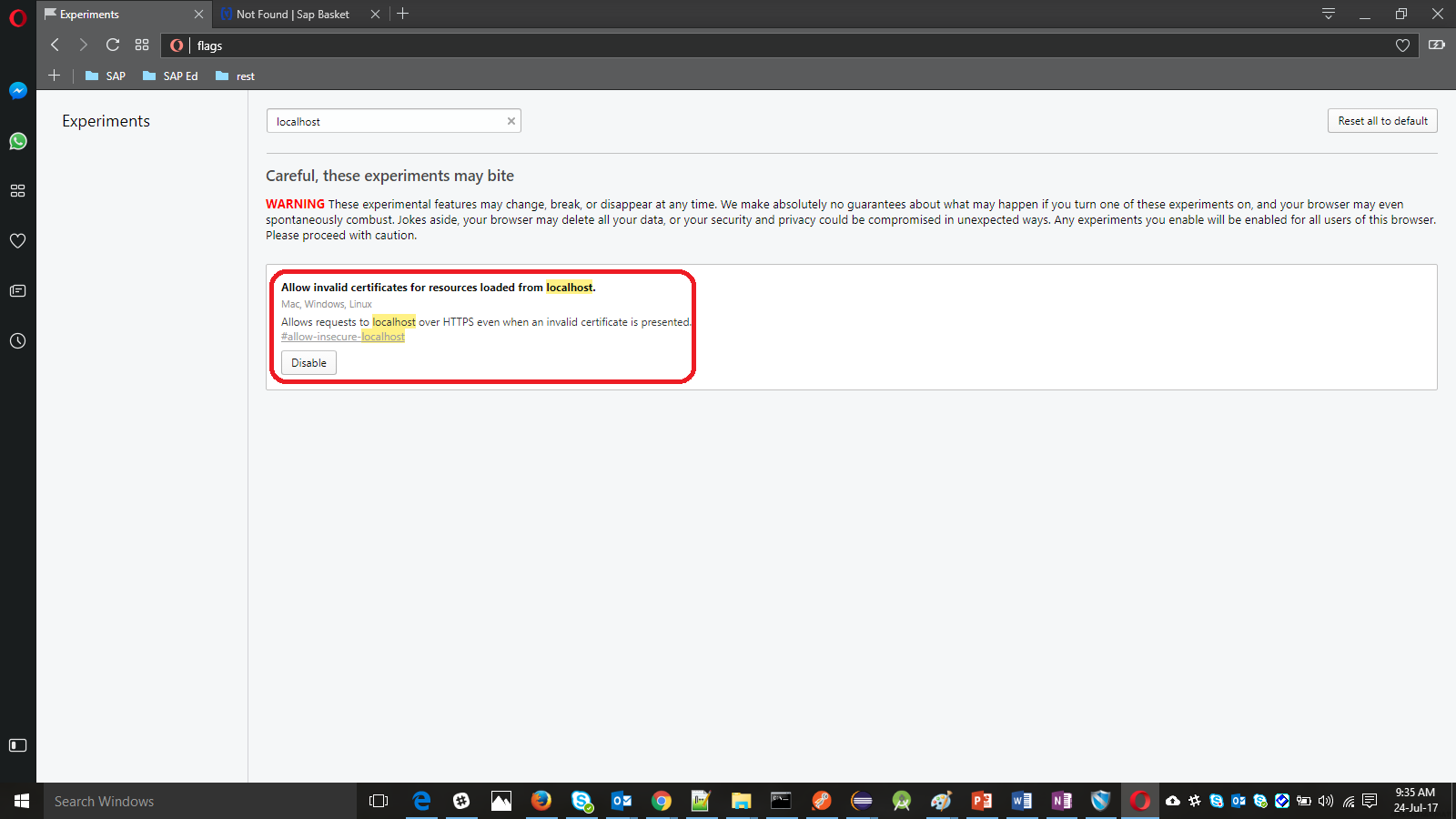
**Web Push Notifications for Hybris**

**Steps:**

1. Ensure that **pushnotificationaddon** AddOn, **webpushnotificationsbackoffice** & the **storefront** extensions, are all included as dependencies directly in the **/hybris/config/localextensions.xml** file.
2. **Optional (if not already done):** Before running the **addoninstall** command in the next step, ensure that the extension **addonsupport** is included as a dependency, **either directly in /hybris/config/localextensions.xml** **or indirectly as <requires-extension name="addonsupport"/>** in the **extensioninfo.xml** file of the **pushnotificationaddon** AddOn.
3. Install the **pushnotificationaddon**, for a specific storefront by running the following command

**ant addoninstall -Daddonnames=”pushnotificationaddon” -DaddonStorefront.yacceleratorstorefront=”<name of the storefront>”**

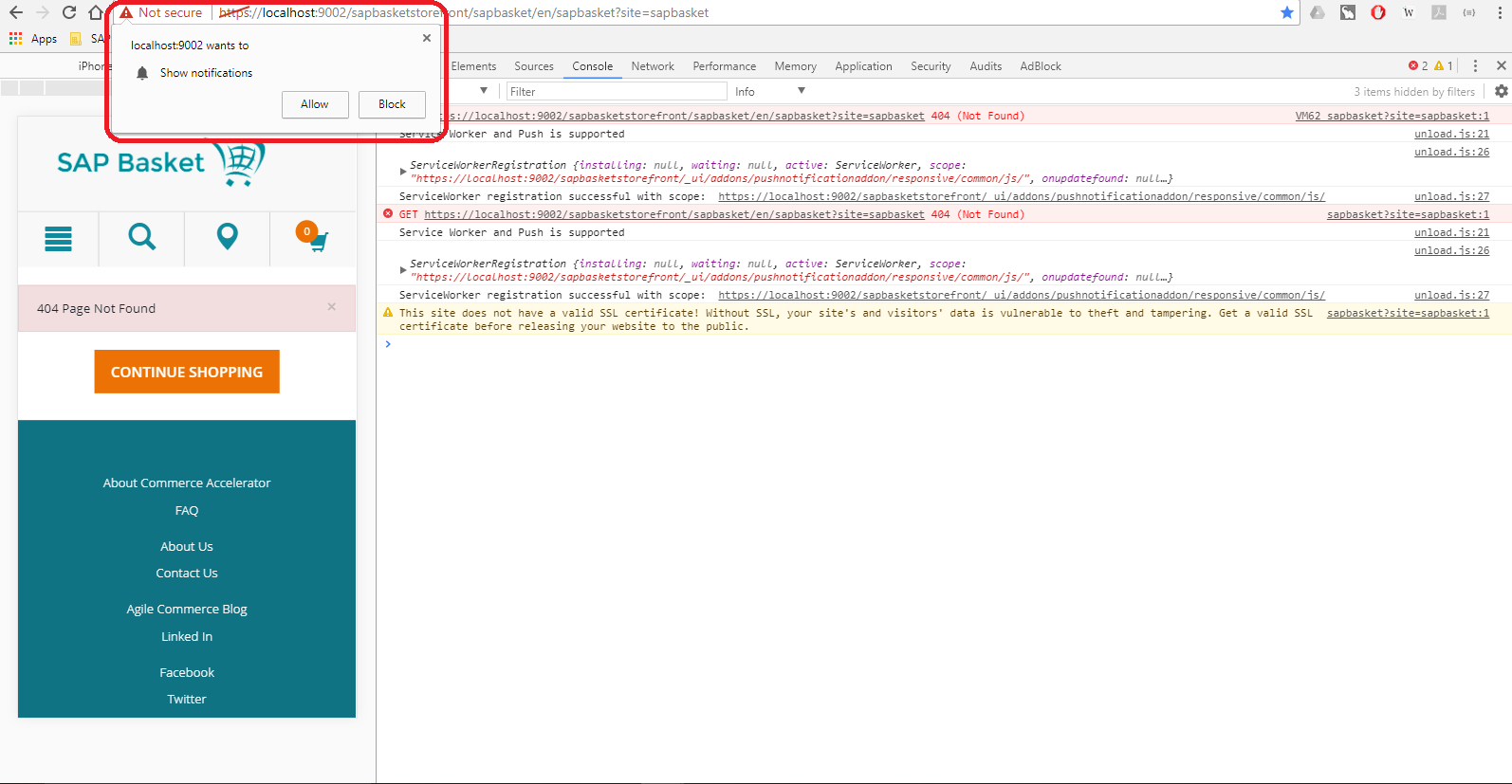
1. Run **ant clean all, followed by hybrisserver.bat [debug]** in the **/hybris/bin/platform/** directory.
2. **Update running system** from the **HAC console**, with the **webpushnotificationsbackoffice** extension **checked.**
3. After **updating** in the previous step, login into backoffice [**https://localhost:9002/backoffice/login.zul**](https://localhost:9002/backoffice/login.zul)(using the login credentials **admin** & **nimda** as username & password, respectively), for hybris server running on localhost port no. 9002(always, by default, unless changed)  
   Then scroll down to find the **Web Push Notifications** **node** on the left hand side explorer tree.
4. Press F4 after clicking on the **Web Push Notifications** **node** in the previous step. Click on the **‘+’** icon in the backoffice perspective. And in the available widgets pop up, select **webpushnotificationbackoffice**.Check **webpushnotificationsbackoffice Sample Widget** option. Click on **Add & Close**.
5. For **hybris server running on localhost**, ensure that the following **chrome** **flag** is **enabled**(by clicking on the **Enable** button, after hitting the url **chrome://flags/#allow-insecure-localhost** in **Google Chrome &/or Opera**), on getting any Security exceptions(SSL Certificate error, resulting in unable to register the Service Worker file **firebase-messaging-sw.js**), on launching the storefront, & then **relaunch the browser(Chrome or Opera** here in this case).Also find the screenshots below for reference purposes.

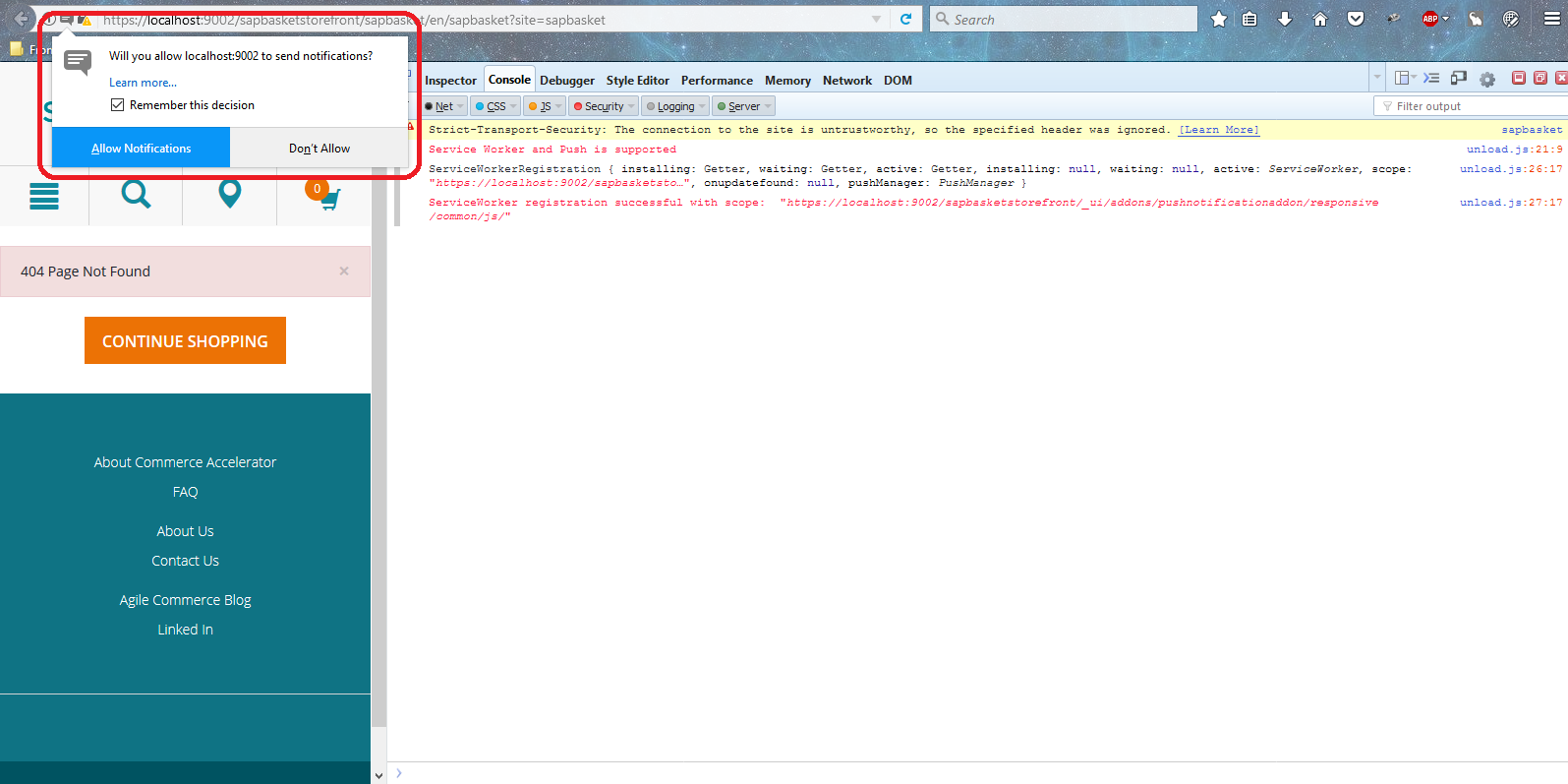


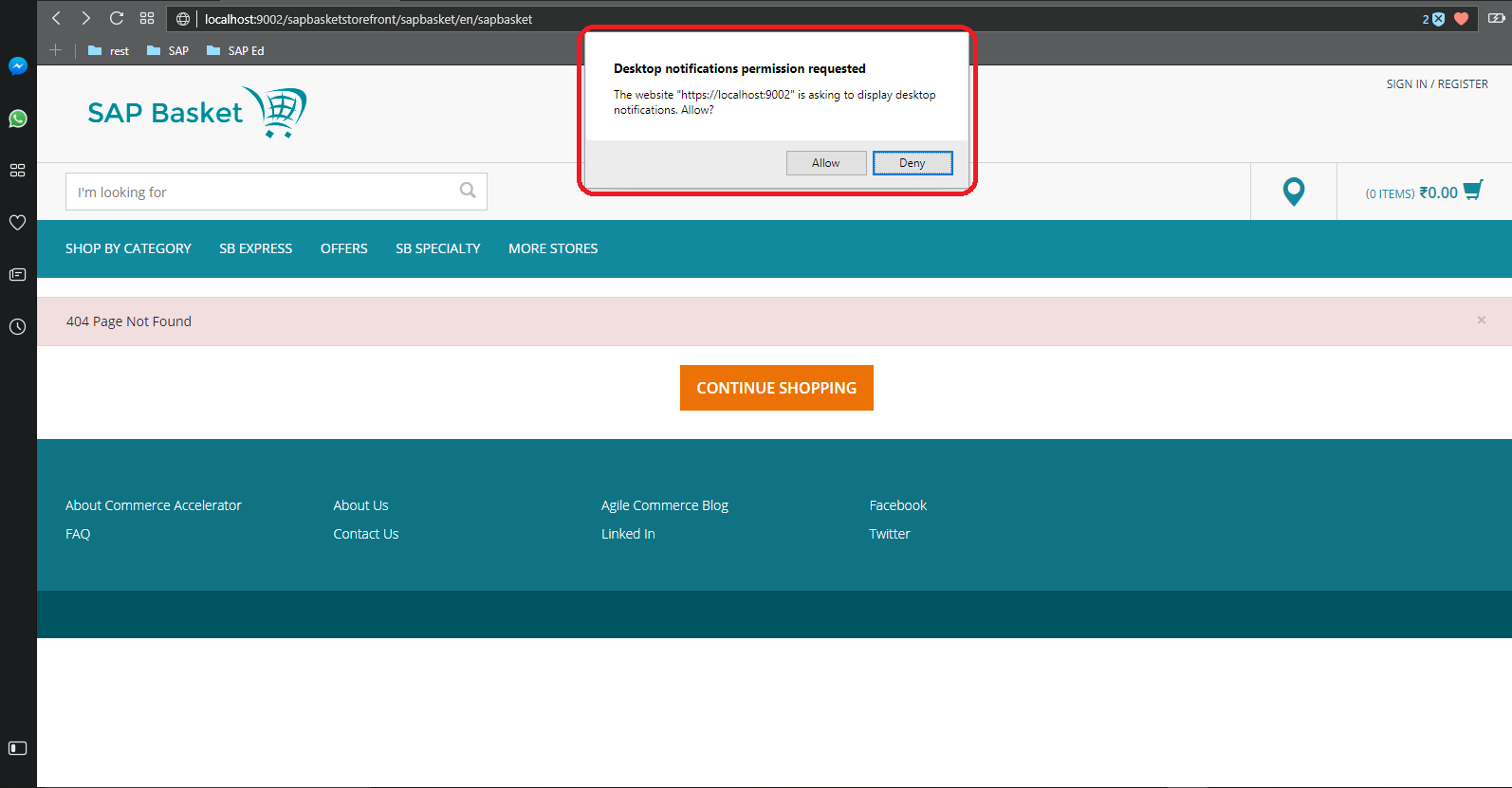
Similarly, for **Mozilla Firefox**, here is the screenshot. So, **uncheck the checkbox** **highlighted in black.**

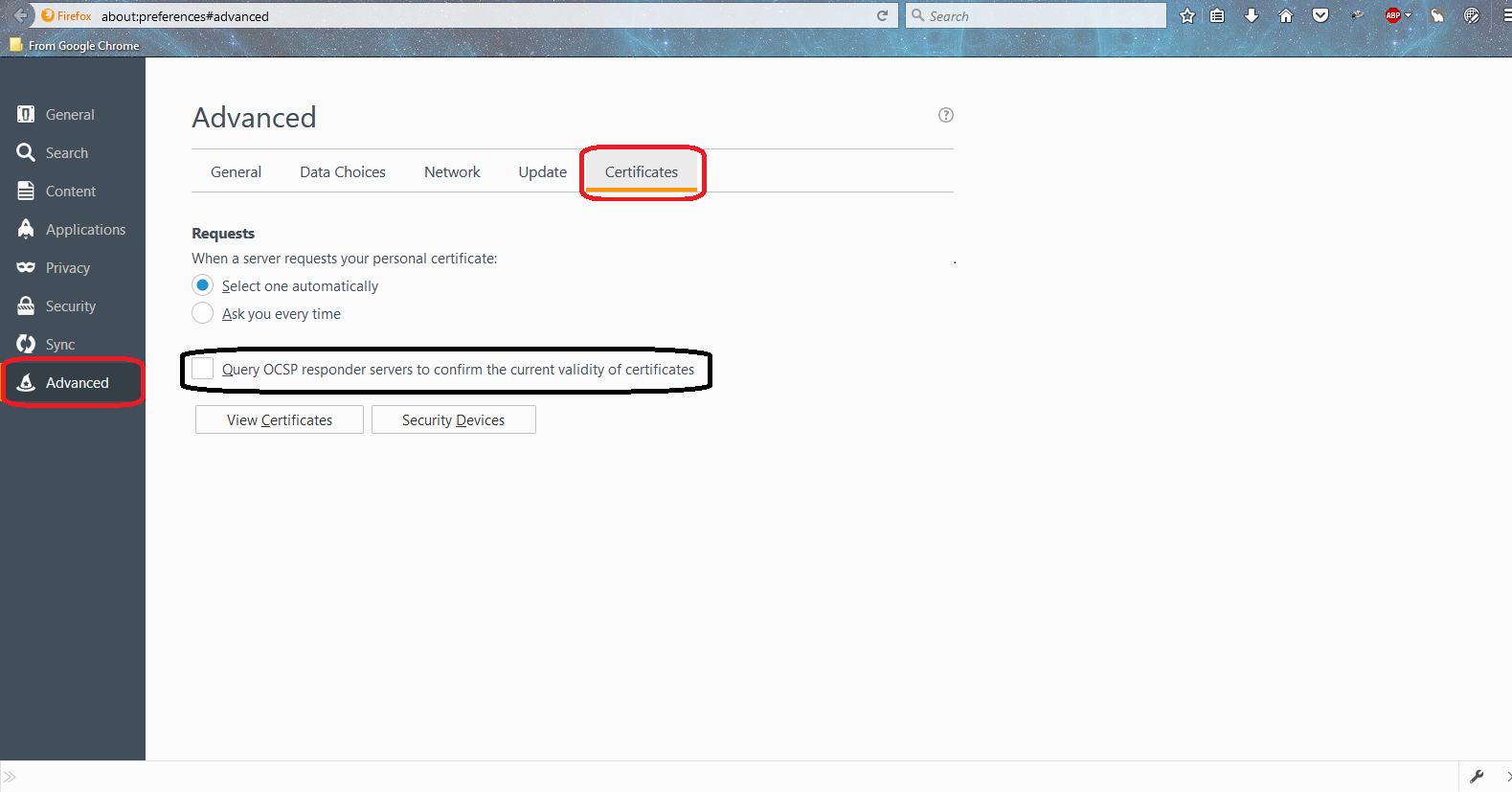
**N.B:** For a storefront **not running on localhost & able to produce a certificate** (for making HTTPS connection requests), this step won’t be needed for any of the 3 browsers supporting Service Workers, as it was only during the development & testing phases.It won’t be required in the production environment, provided a **valid SSL** certificate is produced to the browser, on making HTTPs calls to the server hosting the storefront & also the certificate.

1. Before clicking the SEND button in the next step, the storefront needs to be loaded once in any of the 3 browsers supporting, to **subscribe/opt-in** to receive any future notifications, by clicking on Allow in the **Allow/Notifications - Don't Allow/Block/Deny(depending upon the browser)** popup setting.(Screenshots of all the 3 browsers here).

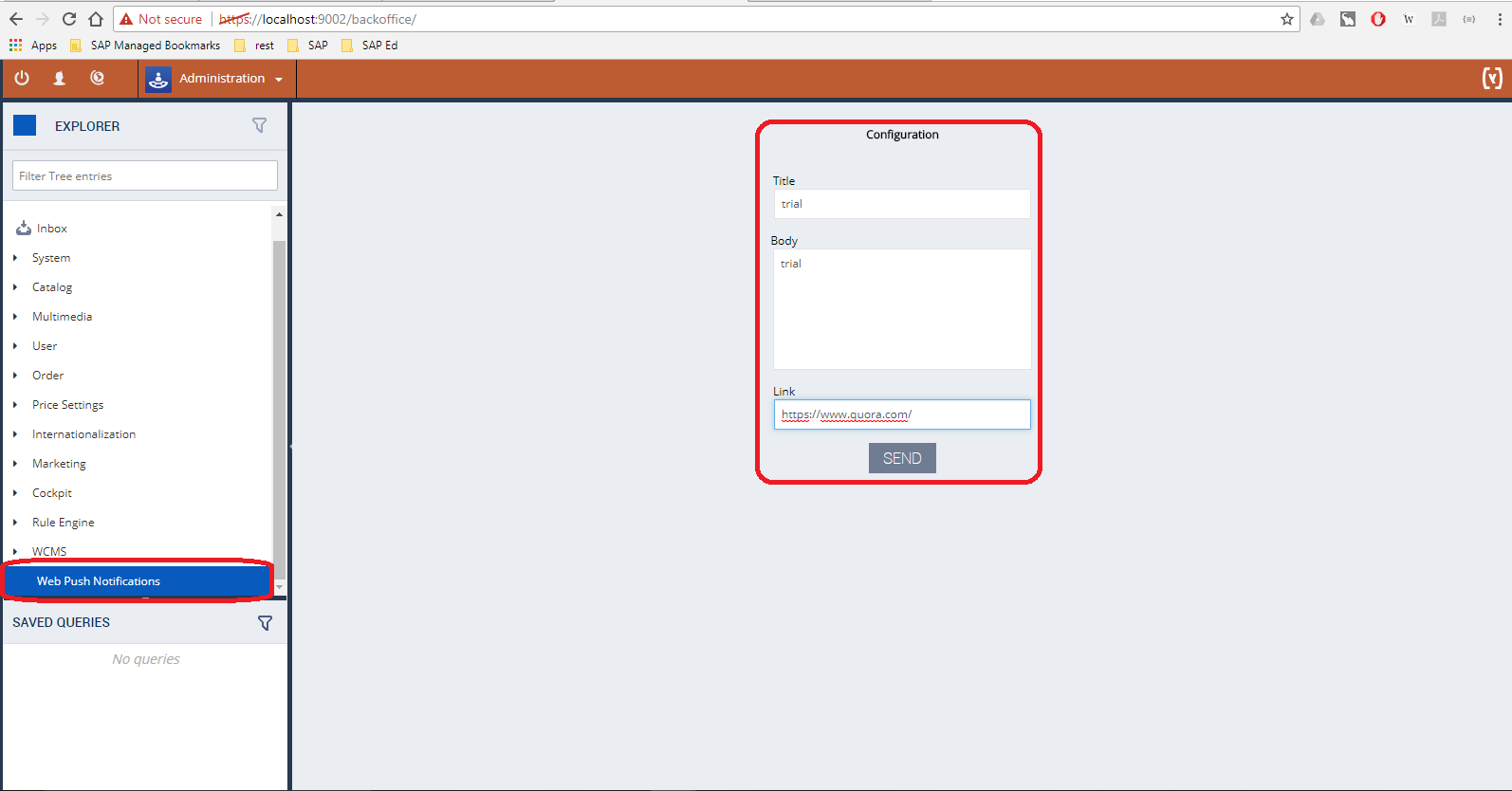






1. **(N.B:** Should be on **SAP Internet** network to **trigger** a push notification from the **webpushnotificationsbackoffice** backoffice extension’s **Configuration**, as the SAP Corporate network doesn’t allow to do so, due to some security reasons.)

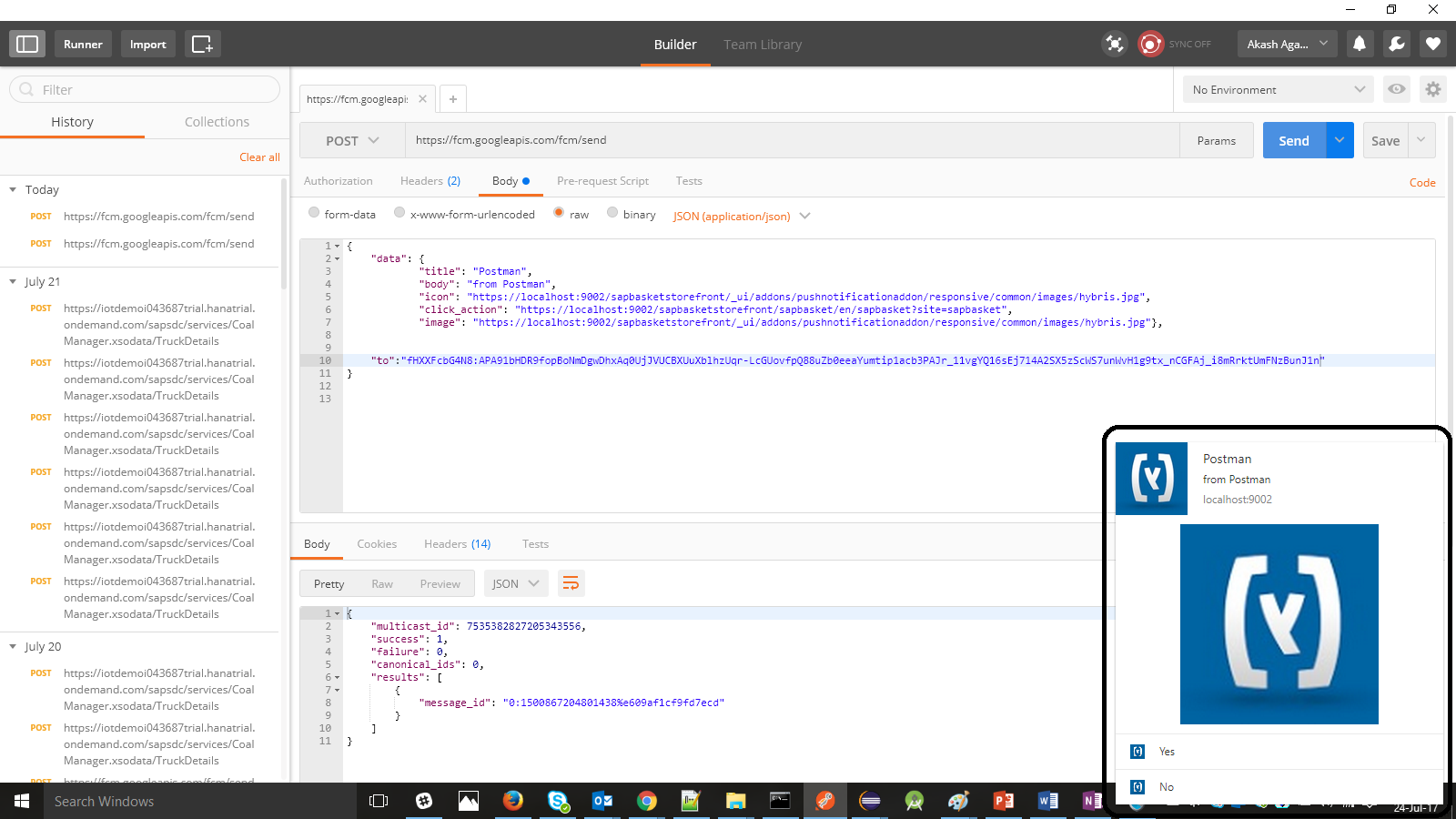
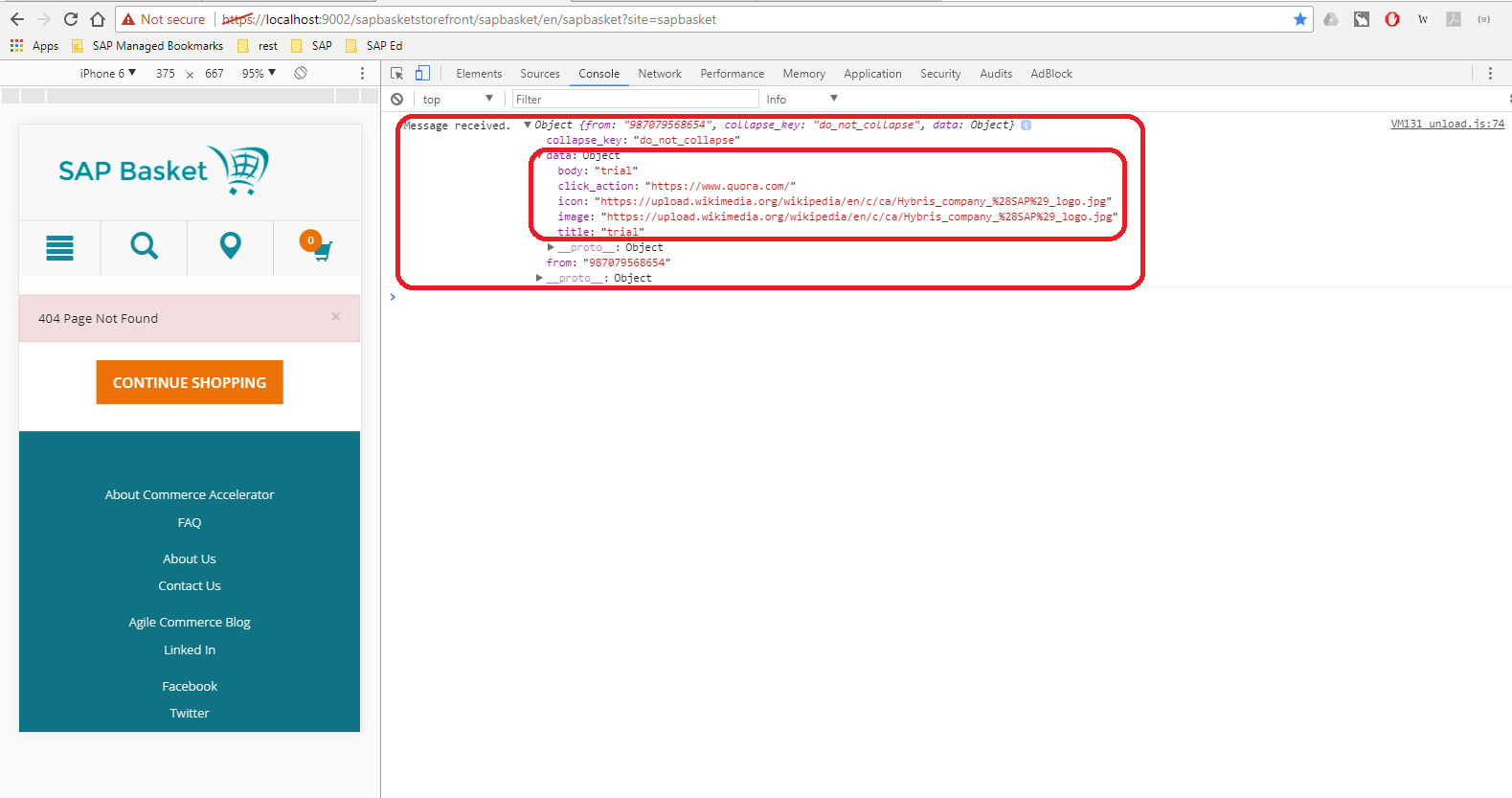
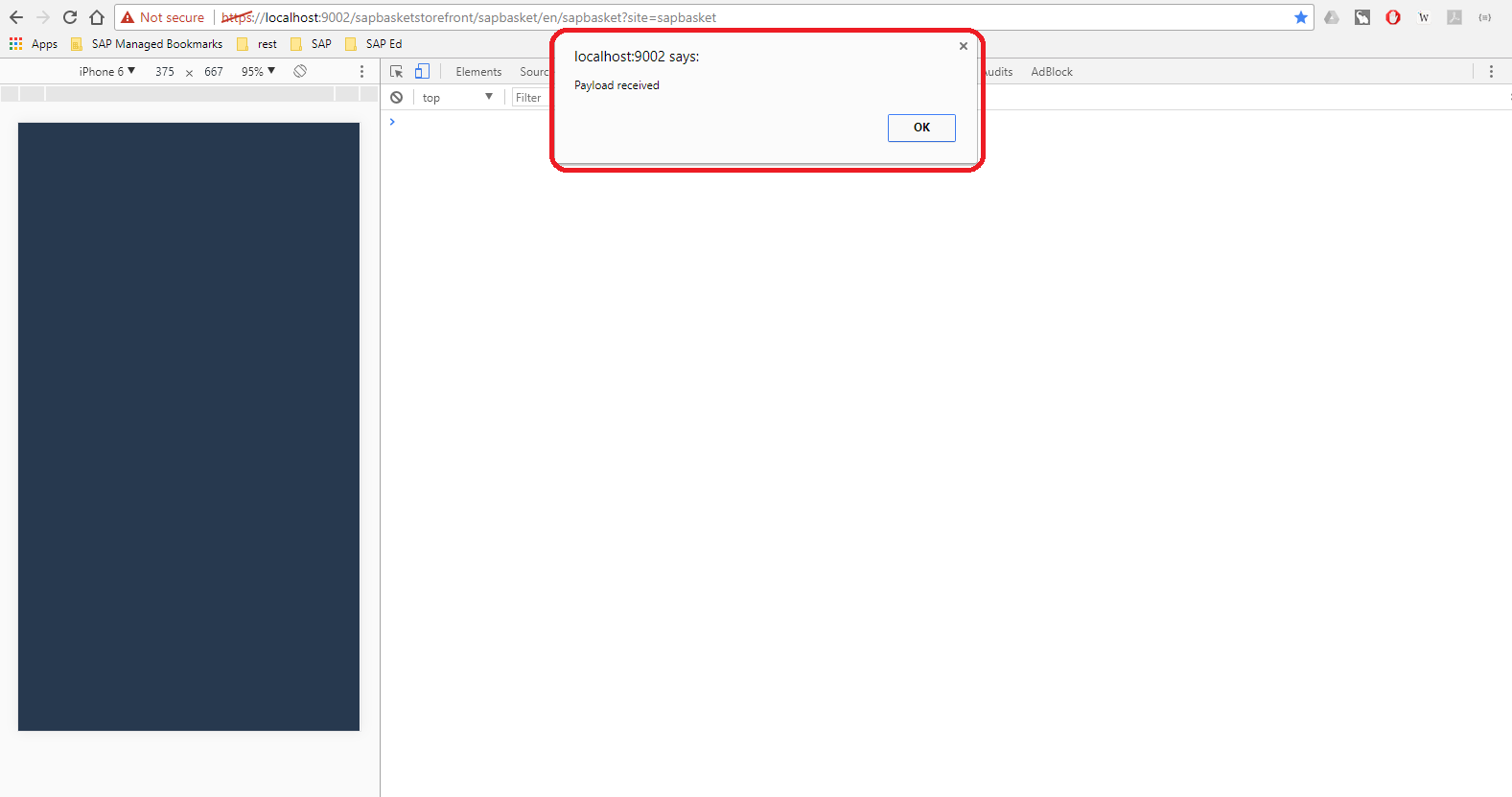
The push notification **Configuration** widget now appears, after pressing F4 again. Enter a **Title,** a message **Body** & a **Link** & then click on the **SEND** button to send a notification message to all the subscribed users.



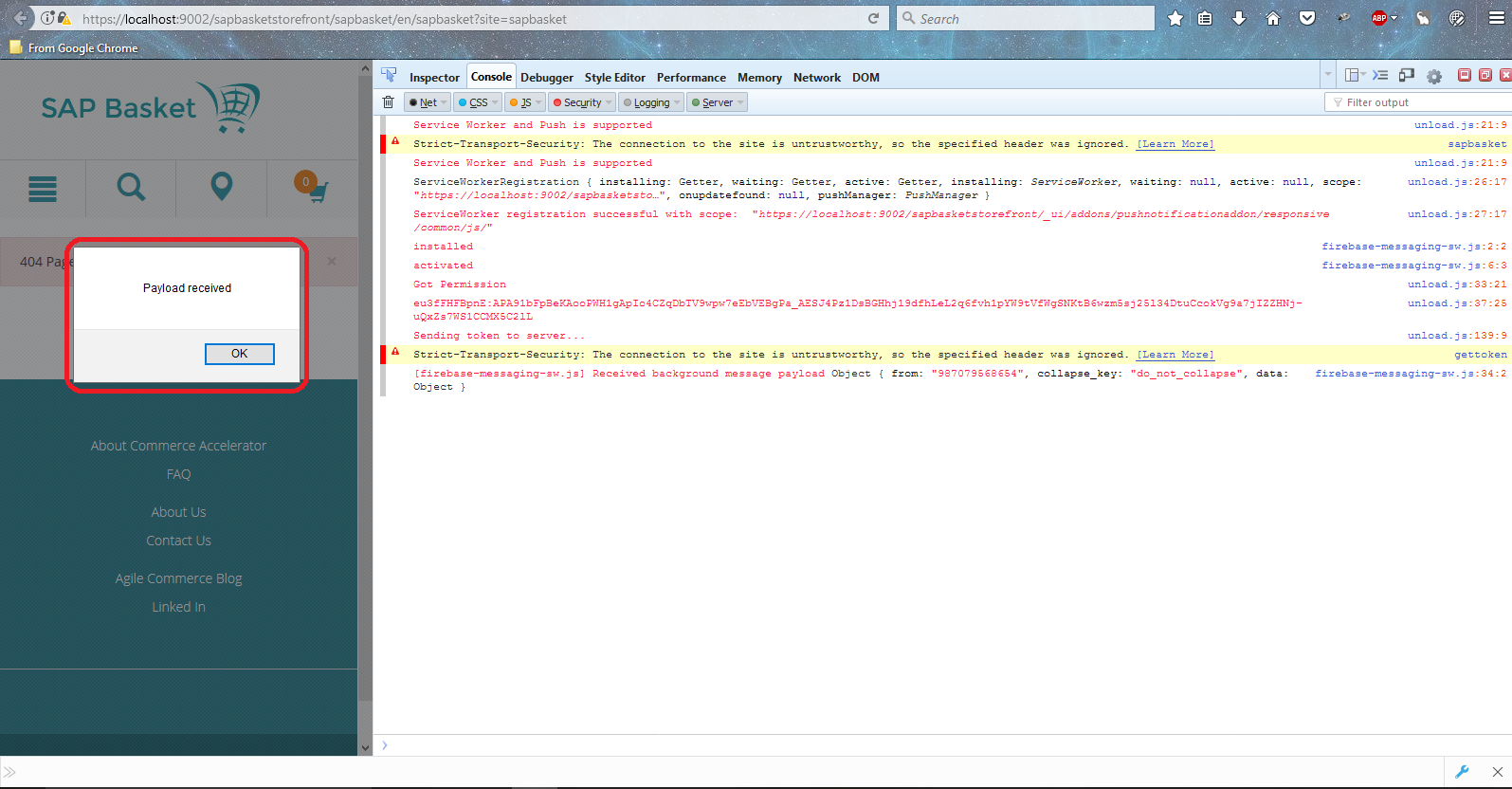
1. If the **storefront’s tab** is **in focus**, then the user will receive an **alert popup** message, or else s/he will receive a **notification popup**.
2. On clicking the **notification popup**, the user is redirected to the **Link** mentioned in the backoffice widget.

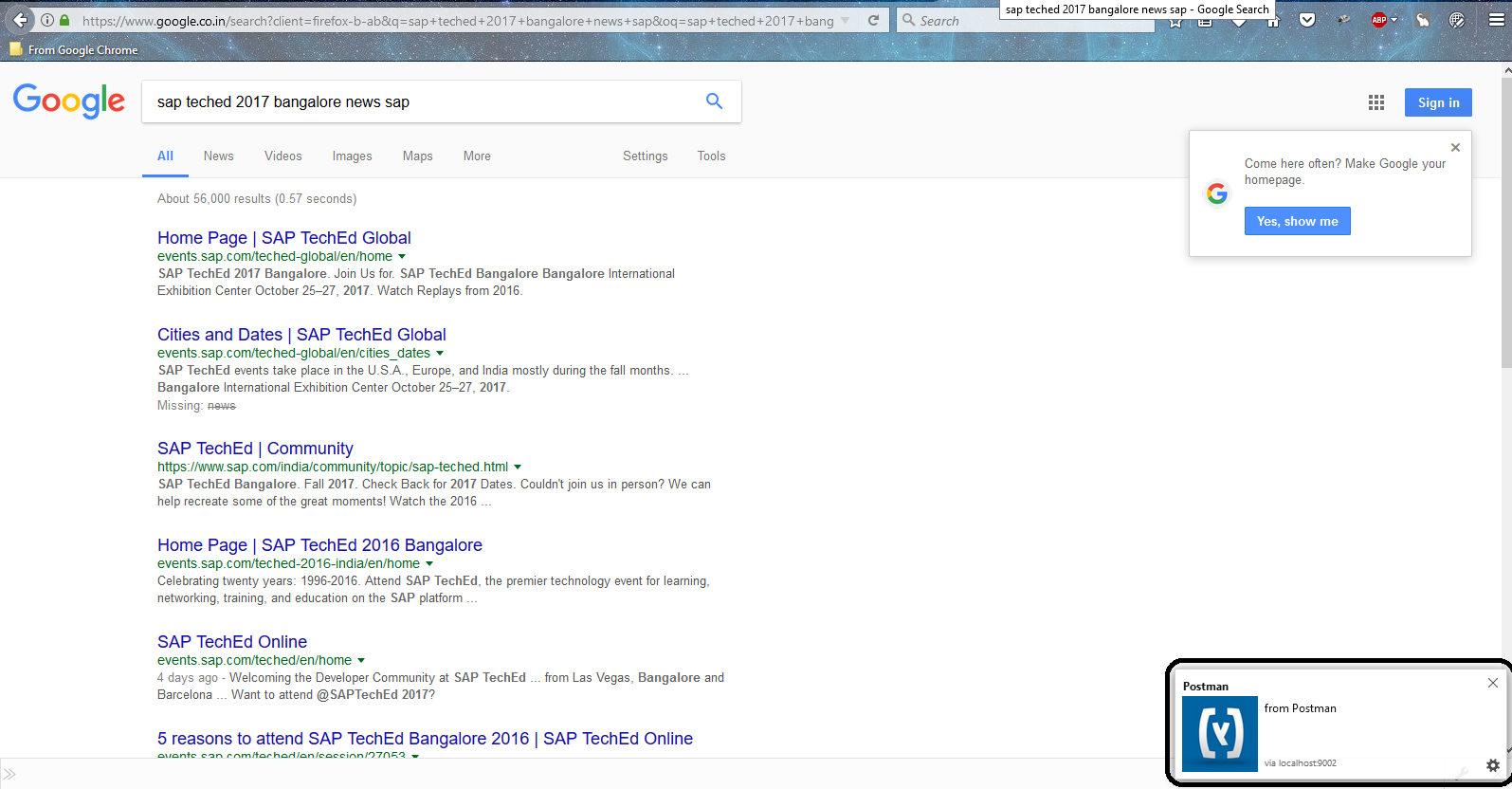
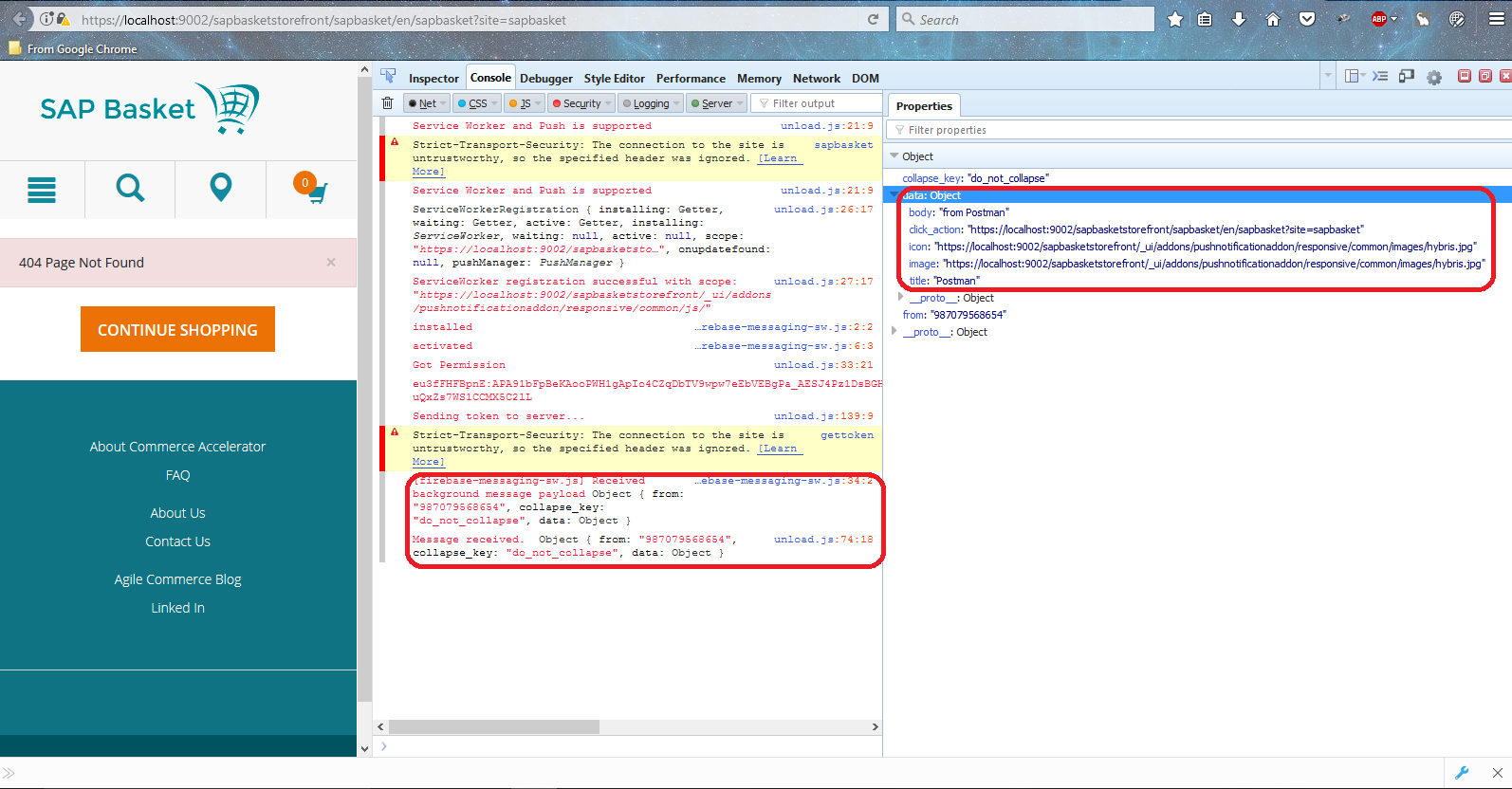
**Screenshots for the previous 2 steps:**

**Google Chrome:**

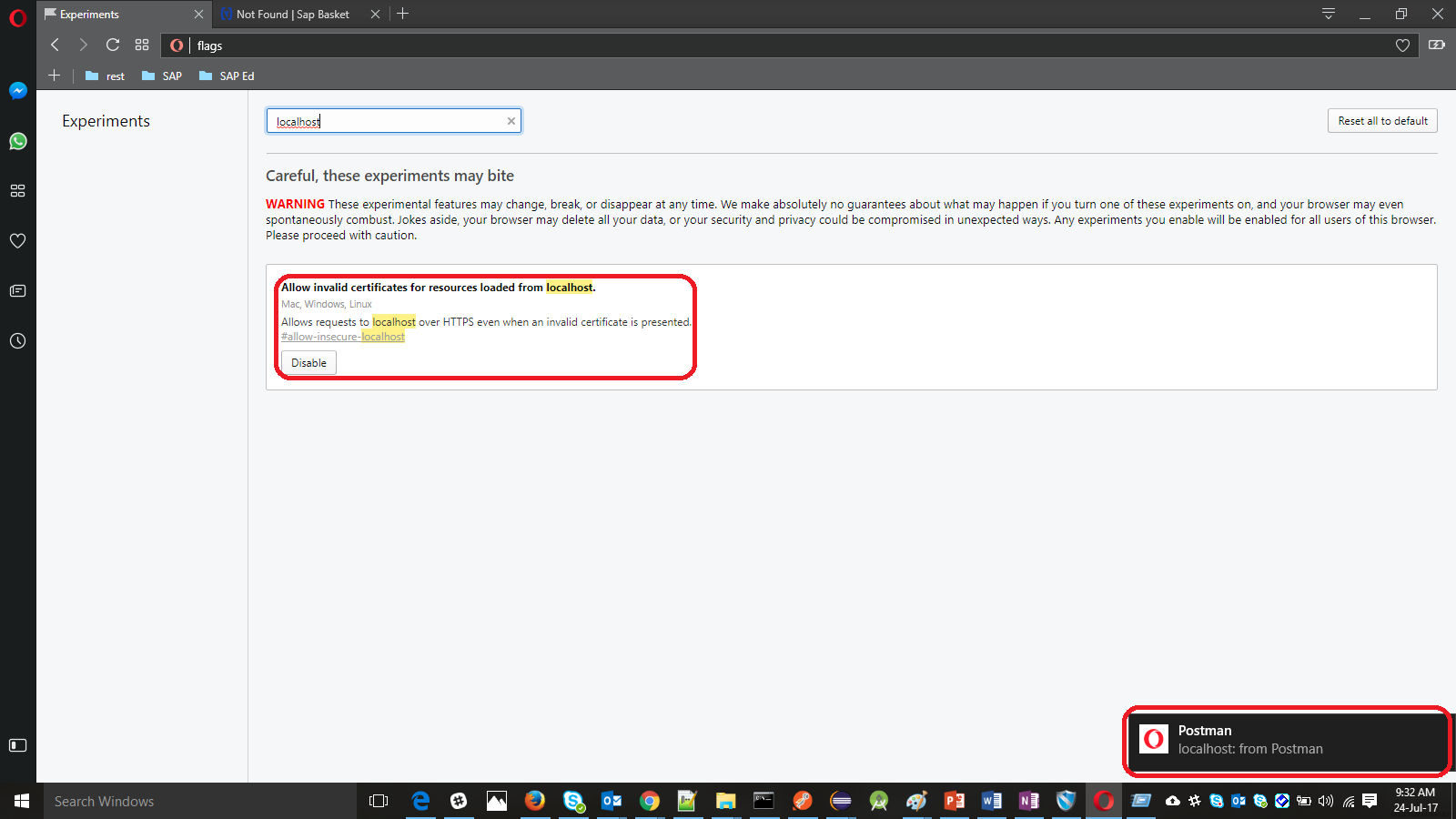
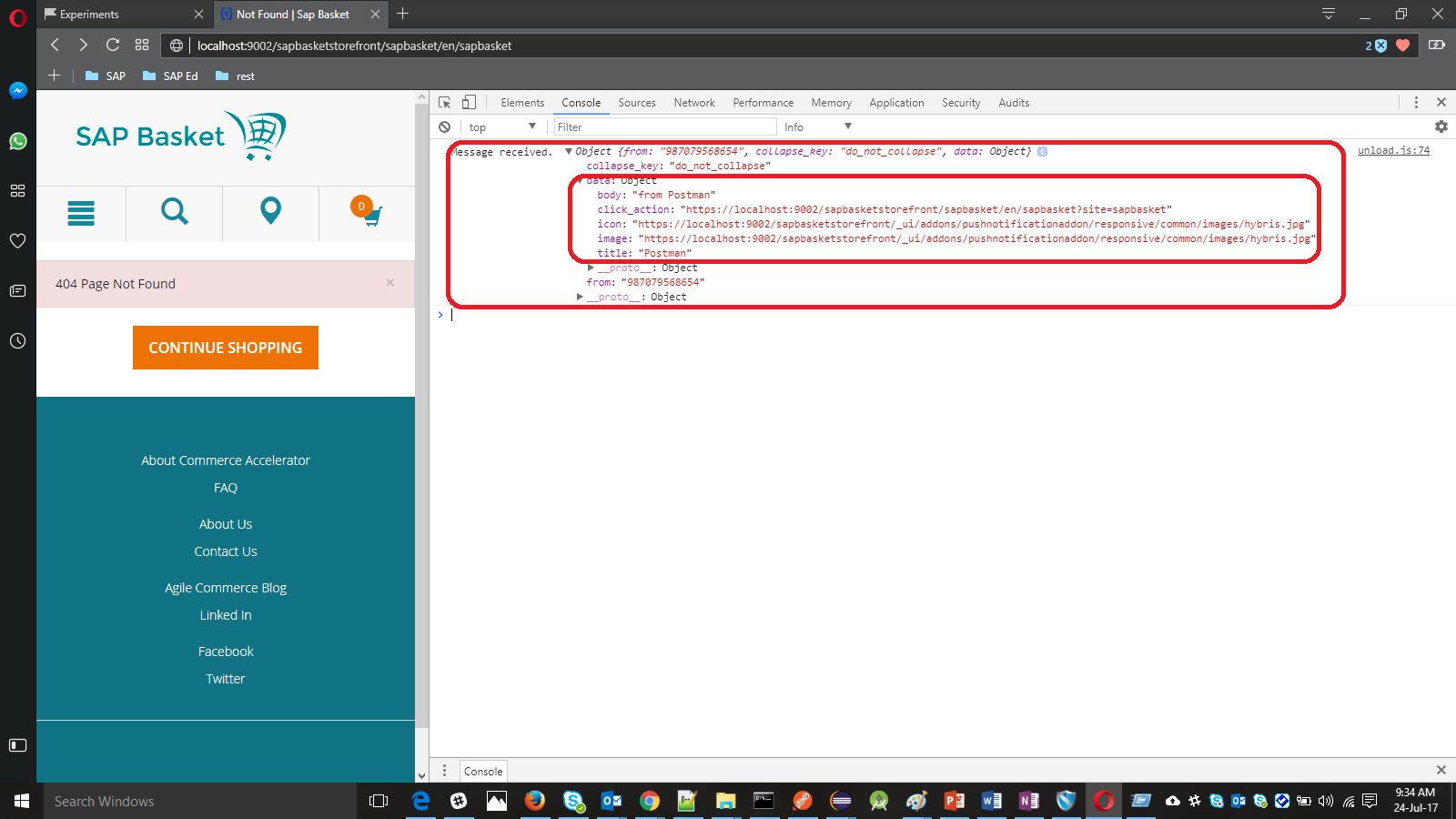
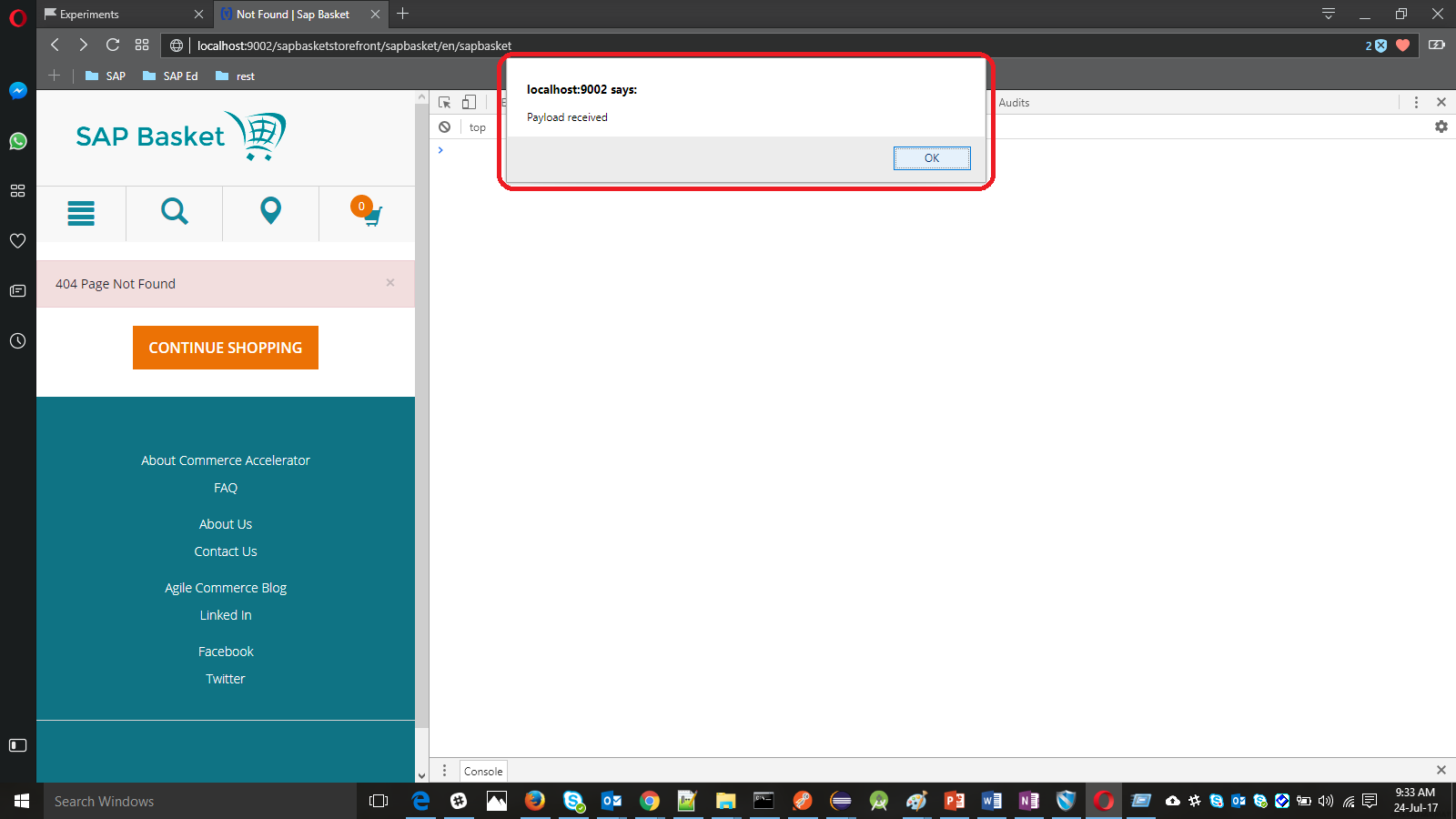
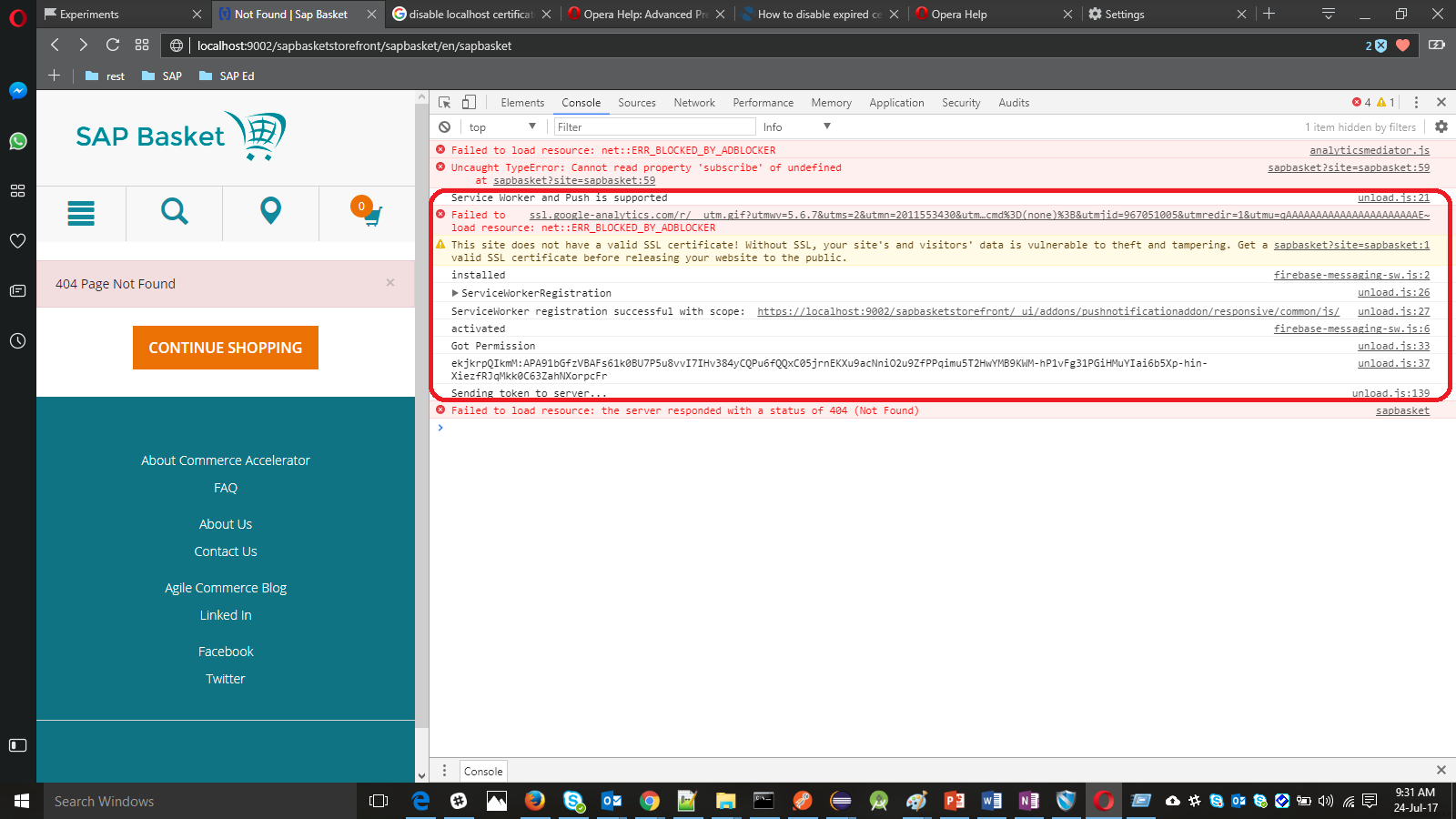
****

**Mozilla Firefox:**

****

****

**Opera:**

****

**Customer Specific Customizations:**

* Create a **new Firebase project** altogether, by clicking on **+ Add Project,** & then follow the steps as required,here

<https://console.firebase.google.com/>

**N.B:** A firebase project is specific to a **Google account** only.

* Then copythe **config** object**,** after clicking **Add Firebase to your web** **app** section of the firebase project just created in the previous step.

Replace the same in **pushnotificationaddon->unload.js** and **pushnotificationaddon->firebase-messaging-sw.js** files.

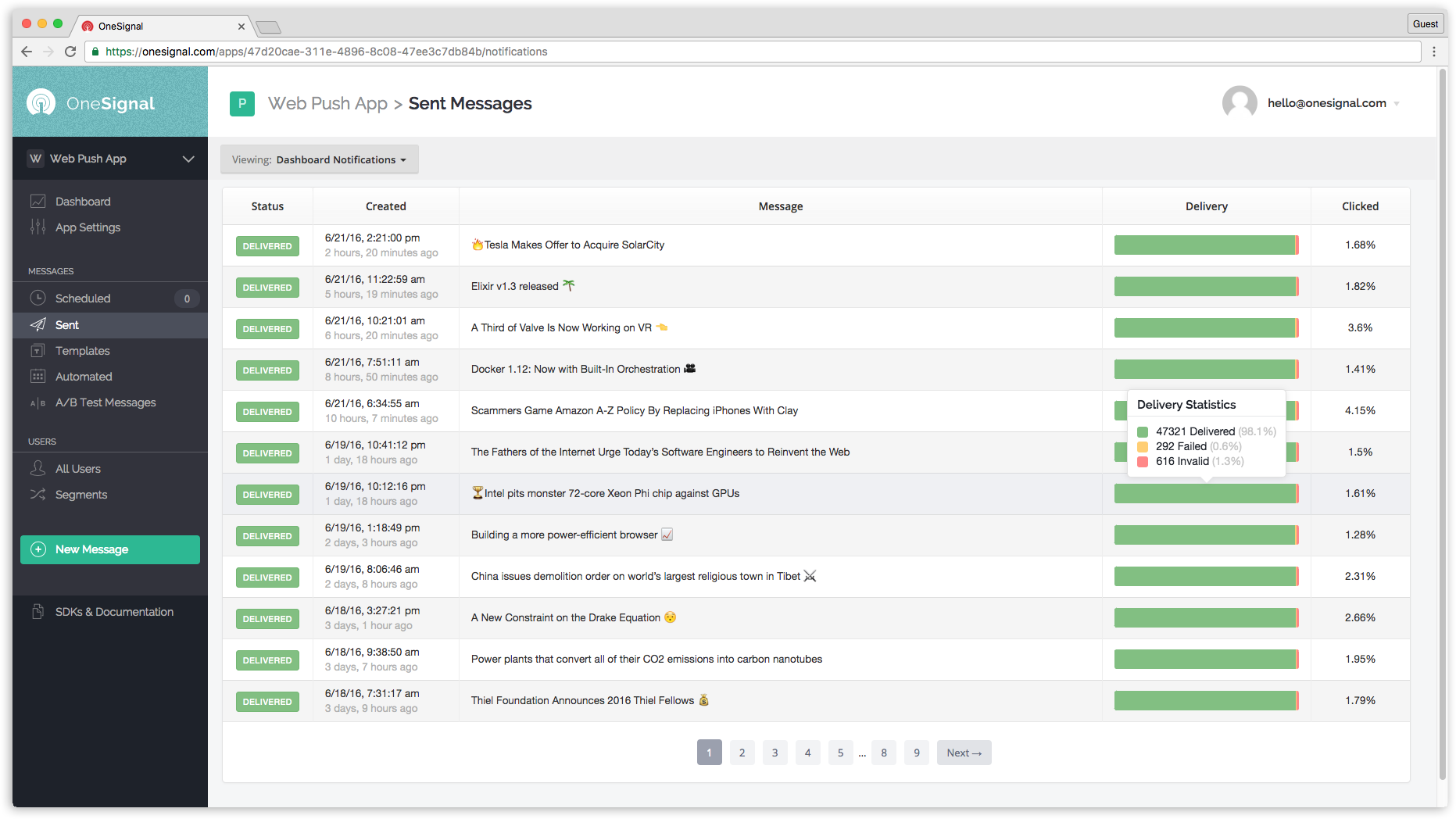
Run **ant all** now to reflect all the changes made **from the pushnotificationaddon to the sapbasketstorefront.**

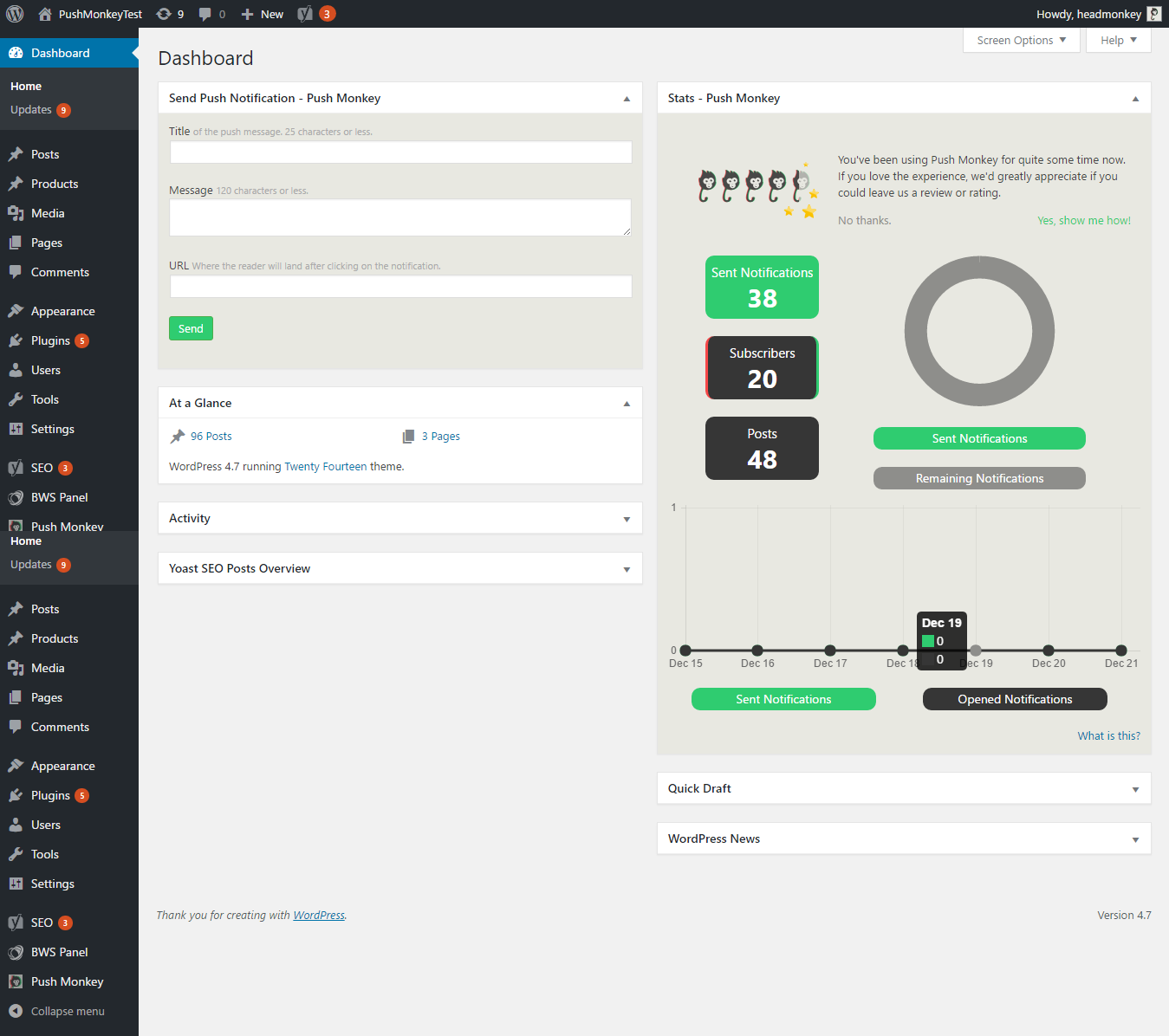
* Replace **conn.setRequestProperty("AUTHORIZATION",<key>)** in **WebpushnotificationsbackofficeService.java** file with the **server key or the legacy server key** of the newly created firebase project (You can get it from **Settings>>CloudMessaging** in your dashboard for your firebase project at <https://console.firebase.google.com/>)

**Extensibility Options:**

* Customer-specific notifications(**User Segmentation**).
* **CronJob(s) to automatically send notifications** at regular time intervals or trigger at certain events.
* Can be used alongside **analytics** systems and correlate with business performance indicators trend
* Few **samples** are as follows:







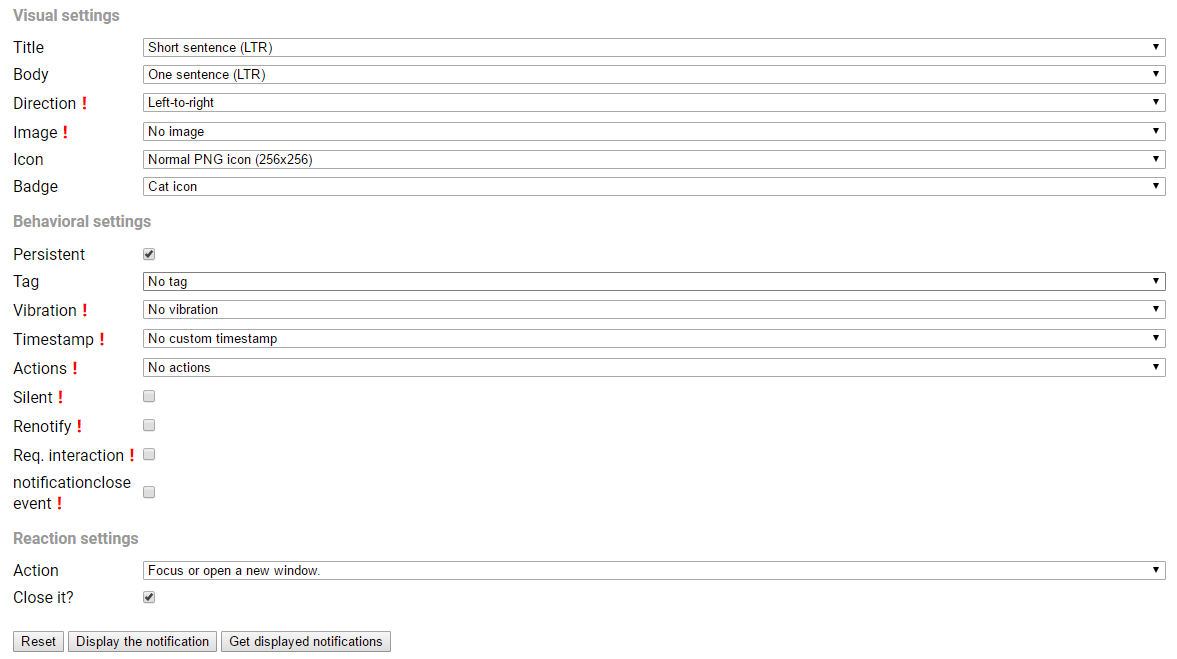
* Notifications can be customized to tweak the behaviour in different devices

Follow the links below for more customization options available

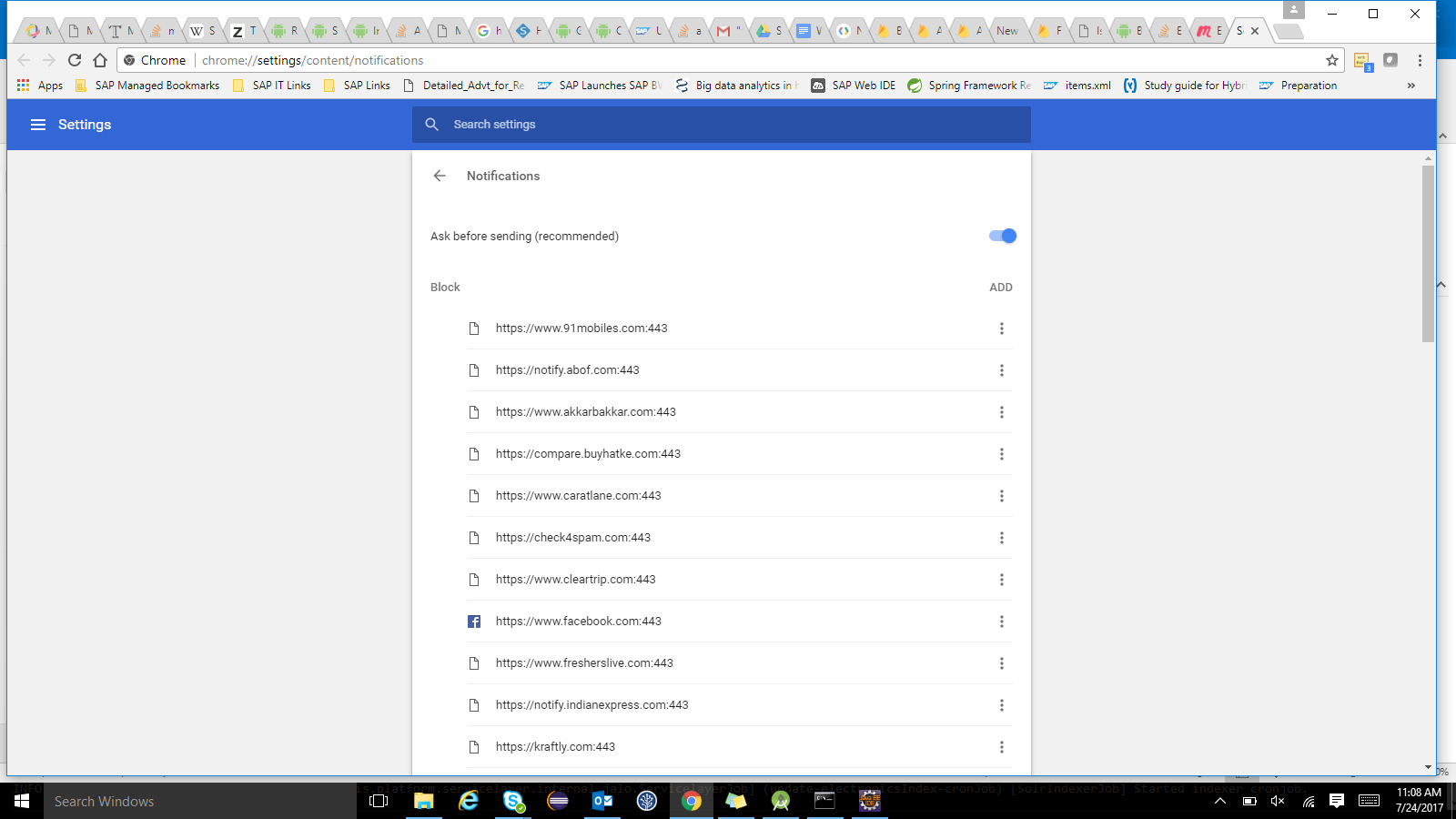
[**https://firebase.google.com/docs/cloud-messaging/http-server-ref#notification-payload-support**](https://firebase.google.com/docs/cloud-messaging/http-server-ref#notification-payload-support) and

[**https://firebase.google.com/docs/cloud-messaging/concept-options**](https://firebase.google.com/docs/cloud-messaging/concept-options)

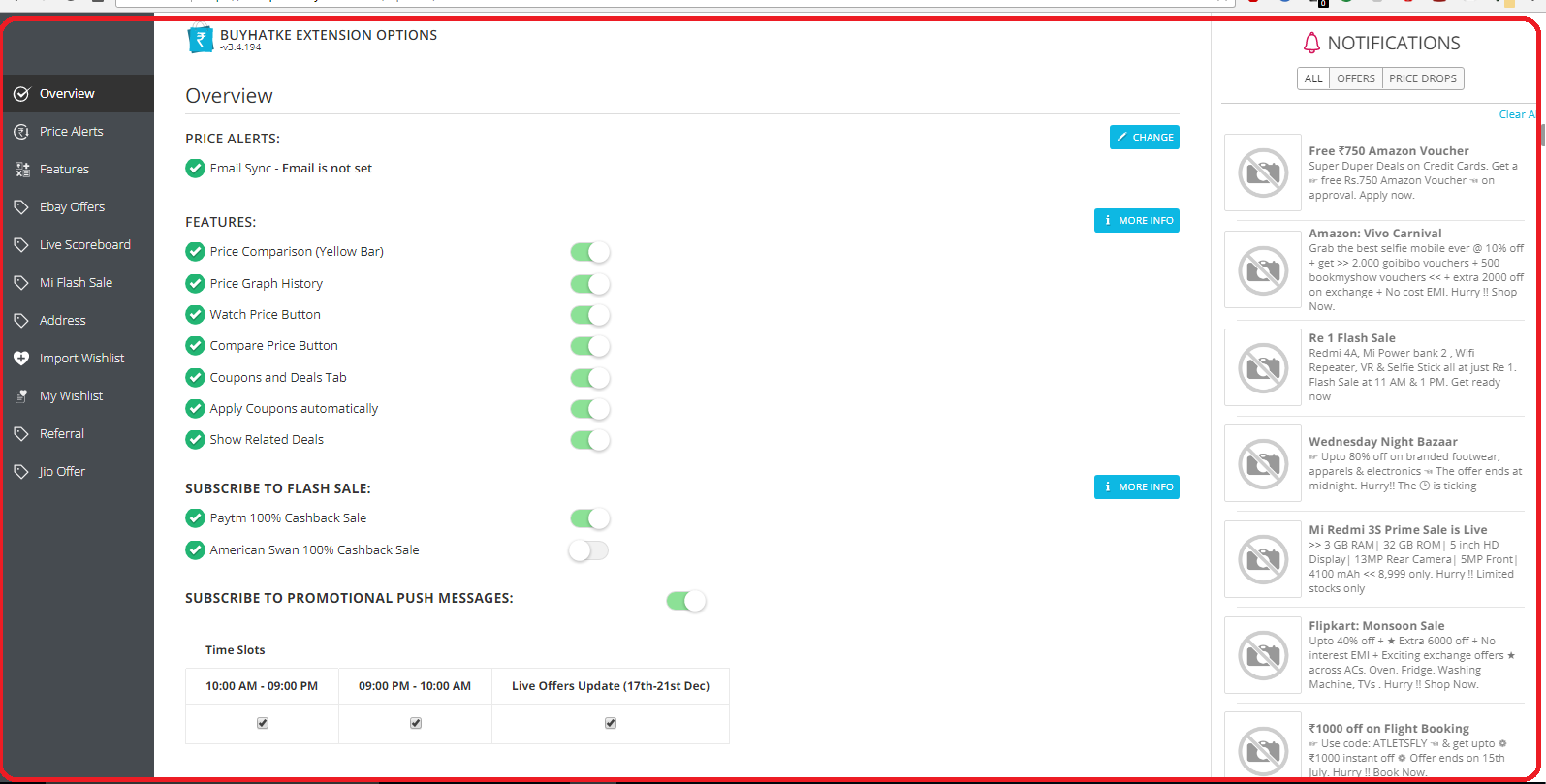
Here’s a sample screenshot...



* Further, on clicking the “settings” gear icon on the notification pop up, we get a default page for push settings, like in chrome



This can be further customized to provide more user specific options like subscribing to a specific sale, time slots to display notifications etc.. For example,



**N.B:**

* All the above steps are fully functional, as tested on Hybris installation zip version **6.2.0.2**
* The end-user can **opt-out/unsubscribe** **any time,** of the subscription(through many ways).
* In order to display all the **tokens** currently persisted in our hybris hsql db, navigate to **hac>>console>>flexible search** & then execute the following query

**SELECT \* FROM {NotificationTokens}**

& then switch to the **Search Result** **tab** to view them.



* **Current state of support for Service Workers(SW) in different browsers(tested on latest versions):**

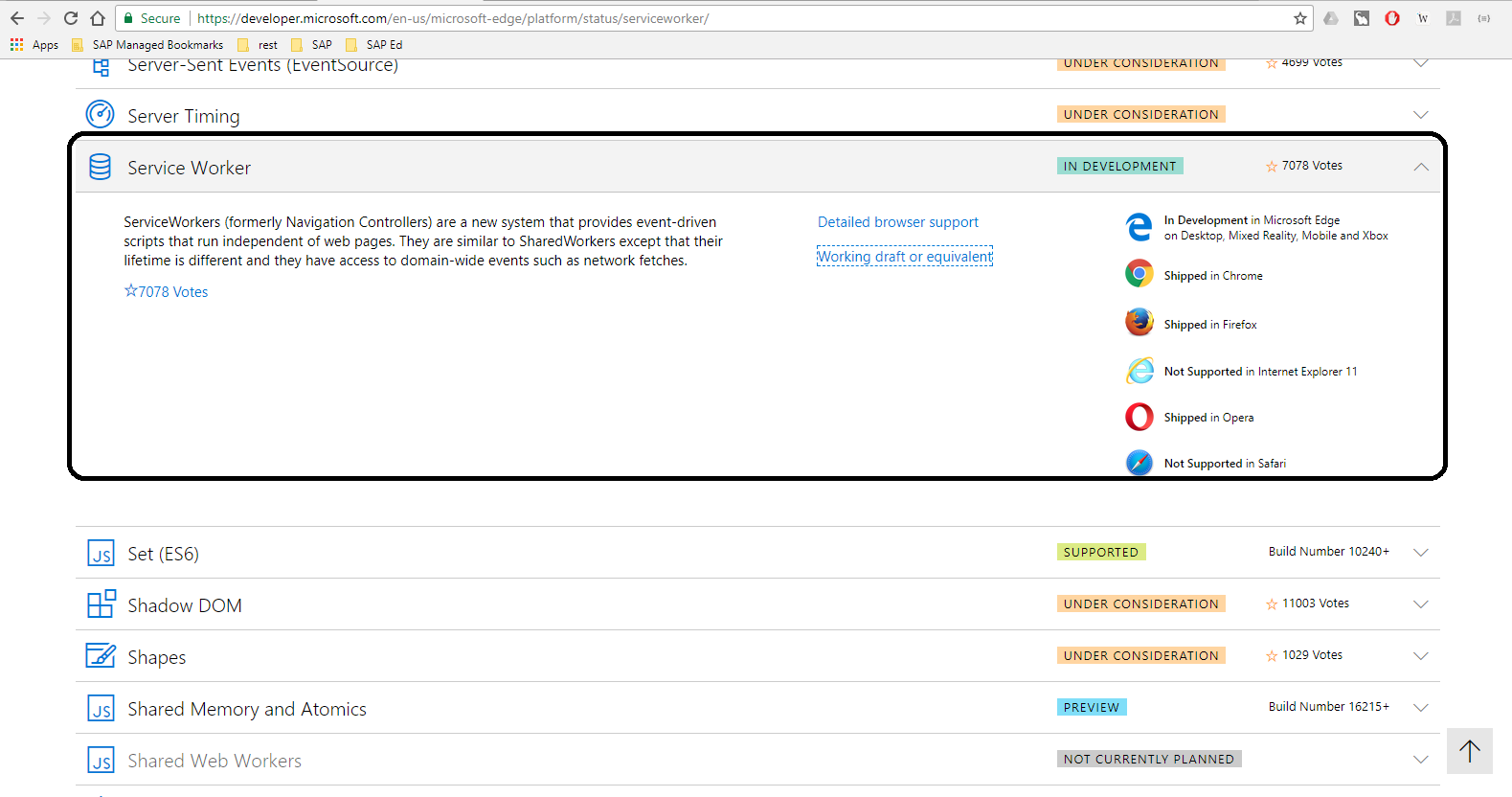
**(Courtesy: h**[**ttps://jakearchibald.github.io/isserviceworkerready**](https://jakearchibald.github.io/isserviceworkerready)**)**

**Google Chrome, Mozilla Firefox & Opera**->Supported

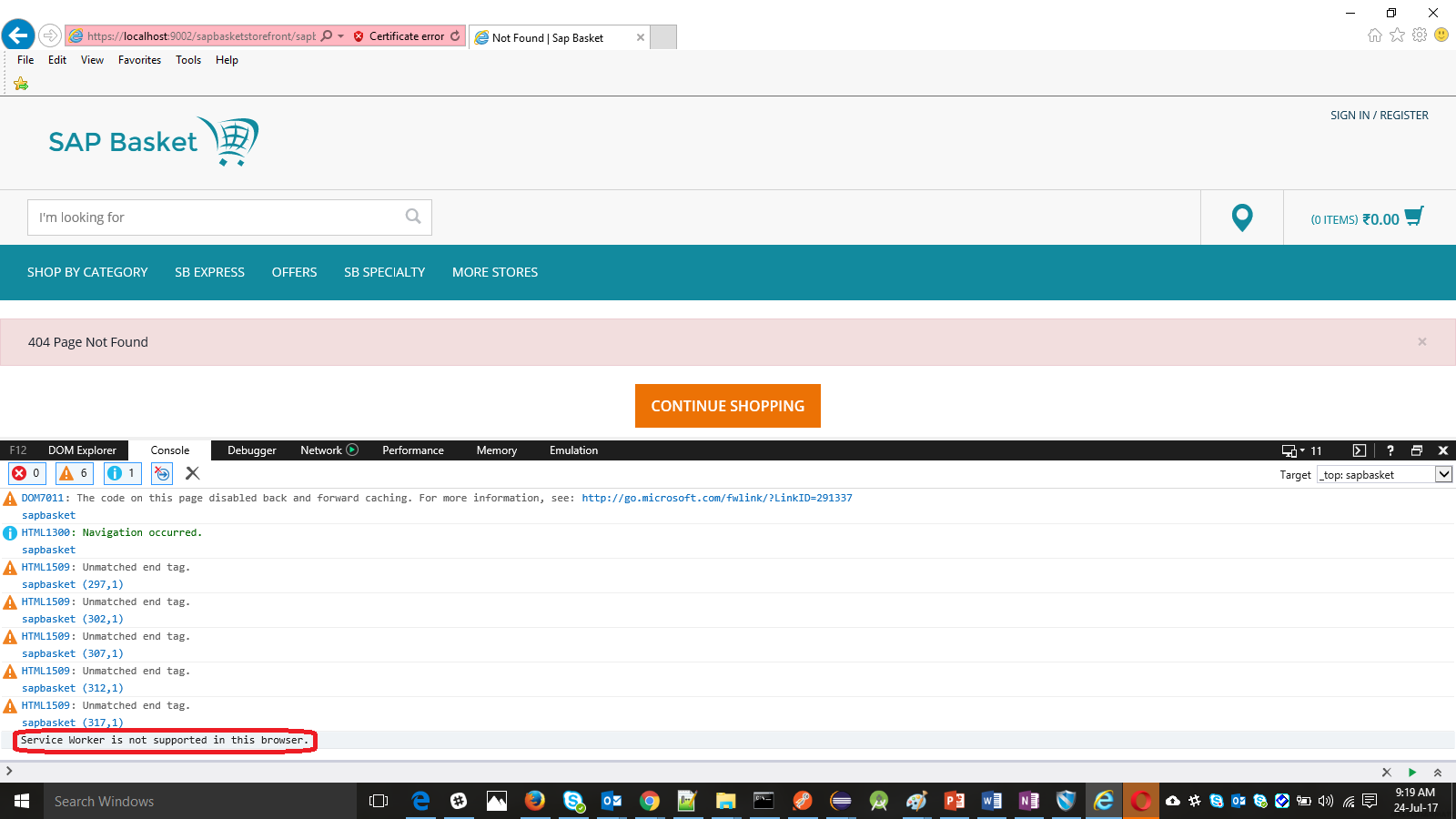
**MS Edge** -> Still under development, to be supported in near future

**MS IE v 11 & Safari** -> Not supported

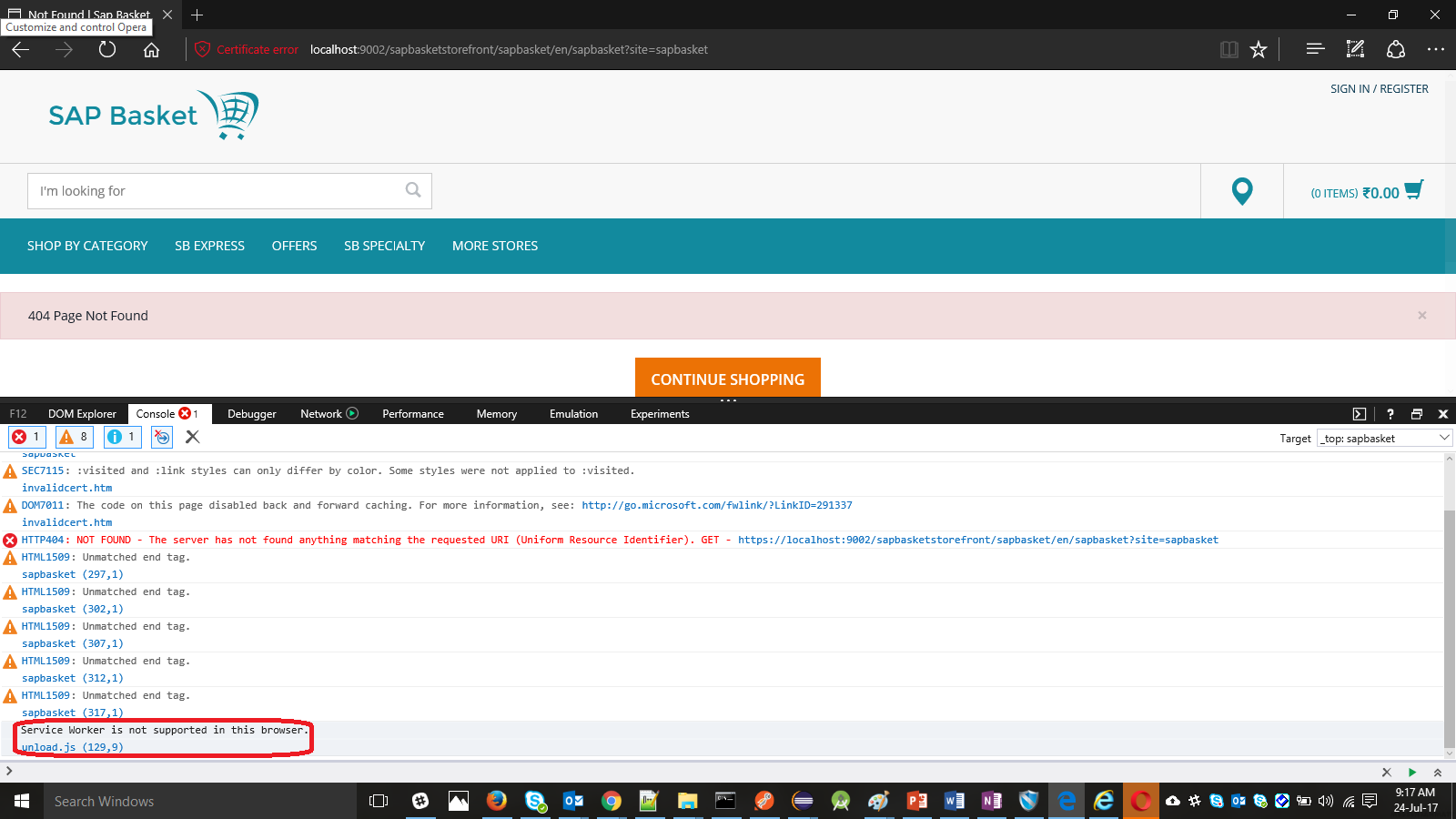
**Screenshots:**

****

**MS IE v 11**

****

**MS Edge**

****