UI Implementation of PunchOut2Go

For PunchOut2Go the UI implementation comes after the session setup step.

Once the session is set up between the Procurement system and the HCL B2B store, the user can shop the items and add to cart.

Once the user add the items to the cart and click on checkout Transfer cart button is displayed on the cart page.

Transfer Cart Component:

This component consists of a button which sends a POST api request to the procurement system along with the cart details. Below is the detailed description of the same.

For the api services we have created a puncout.service.ts file.

**Step 1: Auto login the user.**

The url will include the token and required information to login the user.

On UI side we will extract the required details from the url and auto login the user.

Below is the code for login.



If the trusted token is present in the URL then, auto login the user.

**Step 2: Add items to the cart and proceed to transfer cart.**

Once you login to storefront, user will have an option of going back to the procurement system if he/she doesn’t want to transfer the cart.

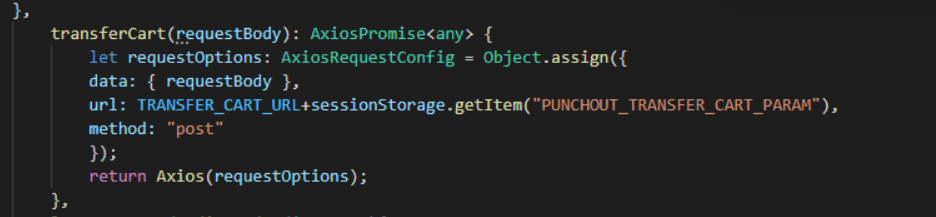
Or he might transfer the cart with all the item details.

The API to transfer the cart is shared by the punchout2GO.

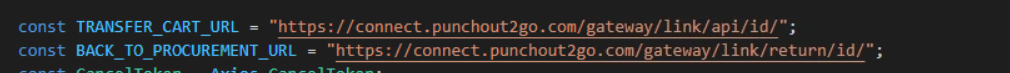
The transfer cart component consist of a button as below.



**Step 3: Create a punchout service file for making the API calls.**



The transfer cart URL is provided as below by the punchout team



The request body to the transfer cart is as below:



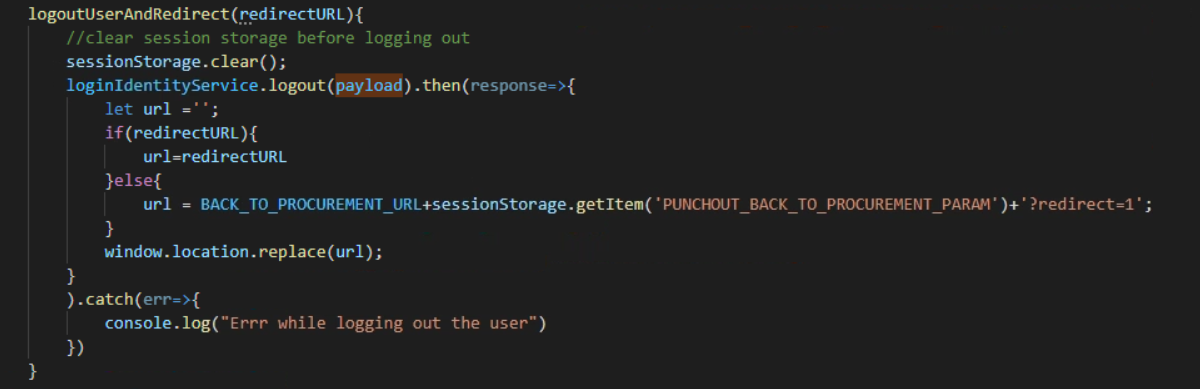
**Step 4: Call HCL commerce submit cart API when the transfer cart response is successful.**



We need to mark the order as complete in HCL Commerce server once the cart transfer is successful.

**Step 5: Logout the user once the transfer cart flow is complete.**

Logout the user from the current session once he/she clicks on back t procurement or transfer cart.



**Step 6: Create a setupProxy.js file and do the below changes.**

This will resolve any CORS issue while integrating with the punchout2go server.

