Frontend: JavaScript

Description

Ochsnersport.ch is one of the clients of behamics. The integration is through the injected script behamics.js. In the source code of the shop you find integrated the behamic.js script. This script tracks the user through events, collects information and sends them to the server through *getmessage* requests. Please have a look on the script and understand the flow of actions in order to answer the corresponding questions.

Hint: To view content of the behamics.js script, Right click -> Inspect -> Network -> JS -> Filter: behamics.js, Refresh the page and wait till the behamics.js is loaded.

Questions

1. List down all information the script stores in browser cookies, storage and temporal storage.

On the **Local Storage**

***bhmuser* -** on the variable userStorageKey. It is an object which contains the following data: Cart, globalControl, guest, sessionID, userID

tempSession - it is an object which contains the following data: Behpageid, behtempid, creationDate, language

It also gets the value of bhmfrom

On the Session storage

Earlycode

It gets earlycodeWishListCode, earlycodeWishListData

No cookies set by this script

2. List down the conditions needed to be met for the script to be executed.

For the script to be executed the _getHostObject function should return true.

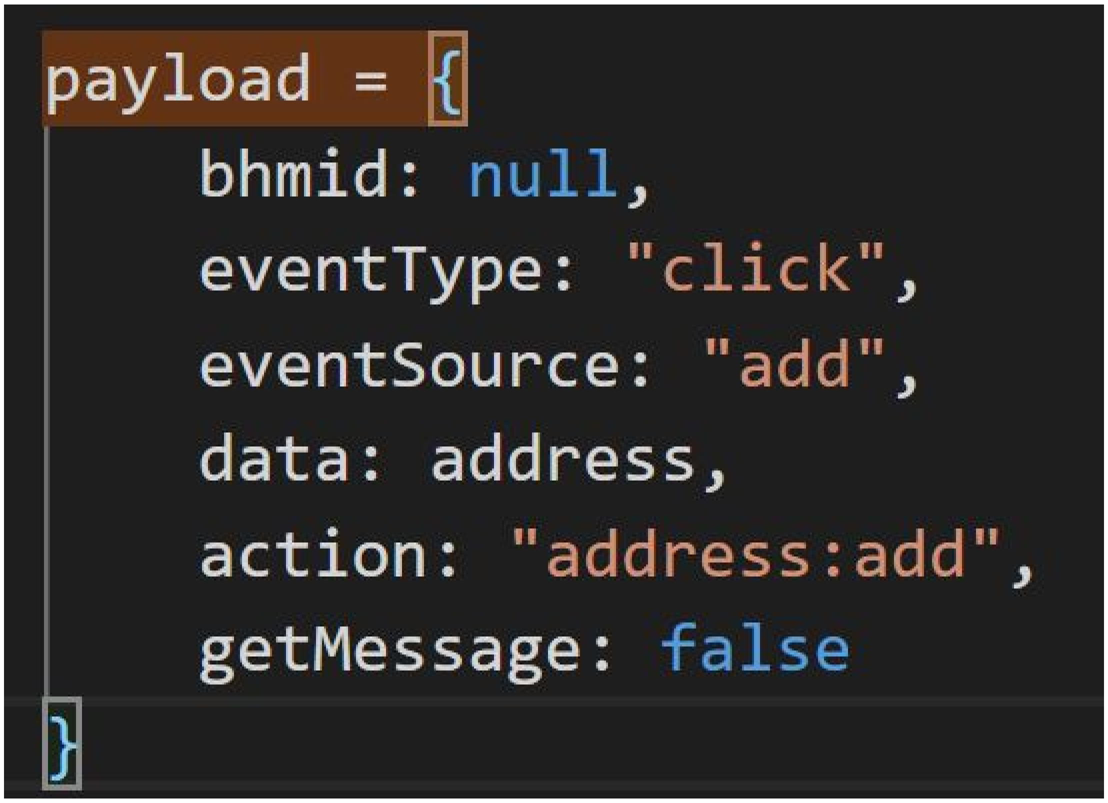
The _getHostObject function returns true if the type of the digitalData variable is not ***undefined*** and, if either one of the following conditions returns true:

- there exists the *digitialData.page* object and the *digitalData.page.pageInfo.pageID* has a value
- there exists the window.dataLayer[0].page object and the window.dataLayer[0].page.pageInfo.pageID has a value

3. List down the attributes of the payload (object) that are sent to the server with each request.

The payload object contains of the following attributes:

- bhmid
- eventType
- eventSource
- data
- action
- getMessage



```
payload = {
    bhmid: null,
    eventType: "click",
    eventSource: "add",
    data: address,
    action: "address:add",
    getMessage: false
}
```

5. Describe in detail what is taking place at codel line 1202.

The line 1202 is executed if there is an error and it executes the _reportErrorToServer function.

The function accepts 2 parameters: the payload object and the errorType.

It creates an object ***finalFormat*** (which contains data for the occurring error) that will be used as a body to the request at the end of the function.

To fill the object with data it checks multiple conditions if variables are set then it sets the variables on the finalFormat object to the existing data.

After checking all the conditions it creates an XMLHttpRequest instance and it checks if the error type was sent.

If there is an error type sent to the function it opens a **post** request including the *type* query param on the endpoint.

If not, it opens the request on an endpoint without the error type.

After that it sets a request header of “Content-type” to “text/plain”

It creates an empty function to be called *onload* of the request.

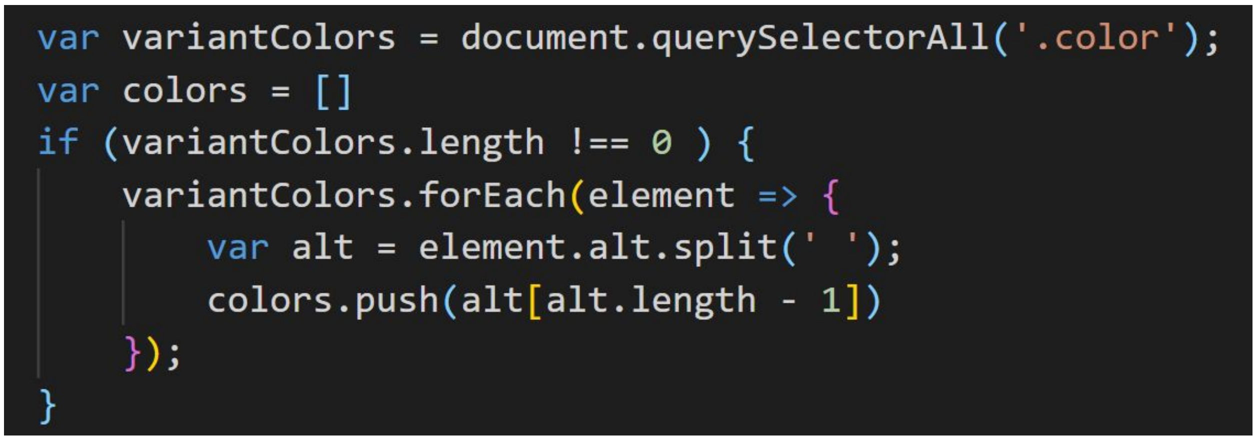
It sends the stringified format of the *finalFormat* object.

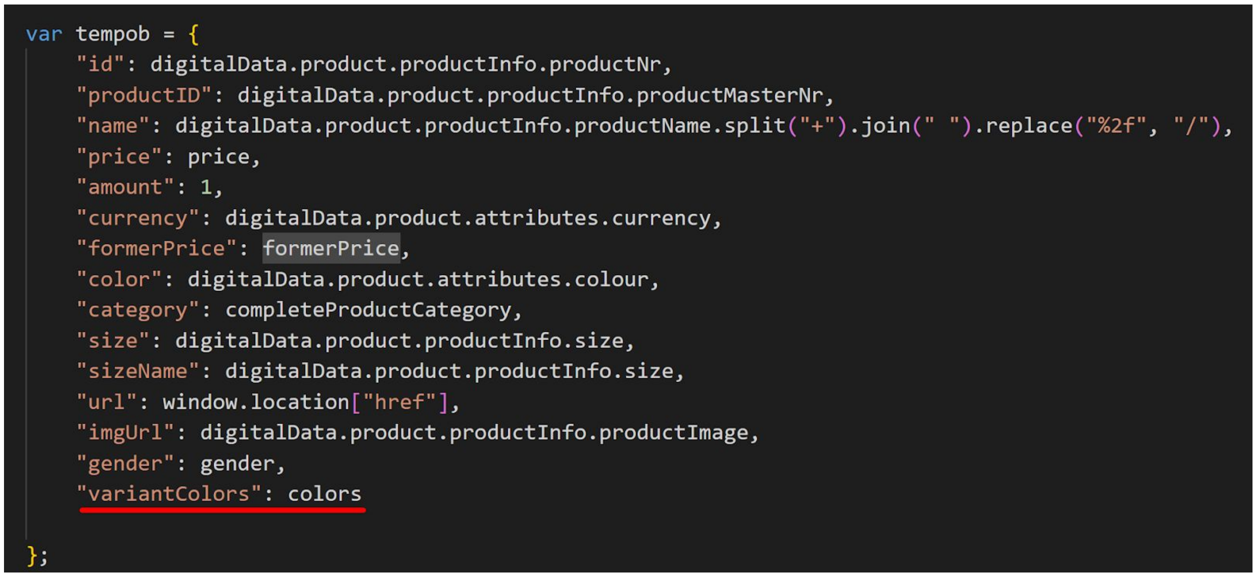
6. Describe in detail what is taking place at codel line 539.

tmpProductList[i] contains an HTML element that holds some product attributes such as product-tile which contains the id of the product tile.

On line 539 it first executes the function ***_onScreen*** which checks to see if the selected element is at the moment shown on the window. If yes we push the ***id*** of the product tile to the *ProductList[“resolvedIDs”]* array and also to the *newIDsToBeResolved* array declared at the beginning of the *_getListOfProductsIDs* function.

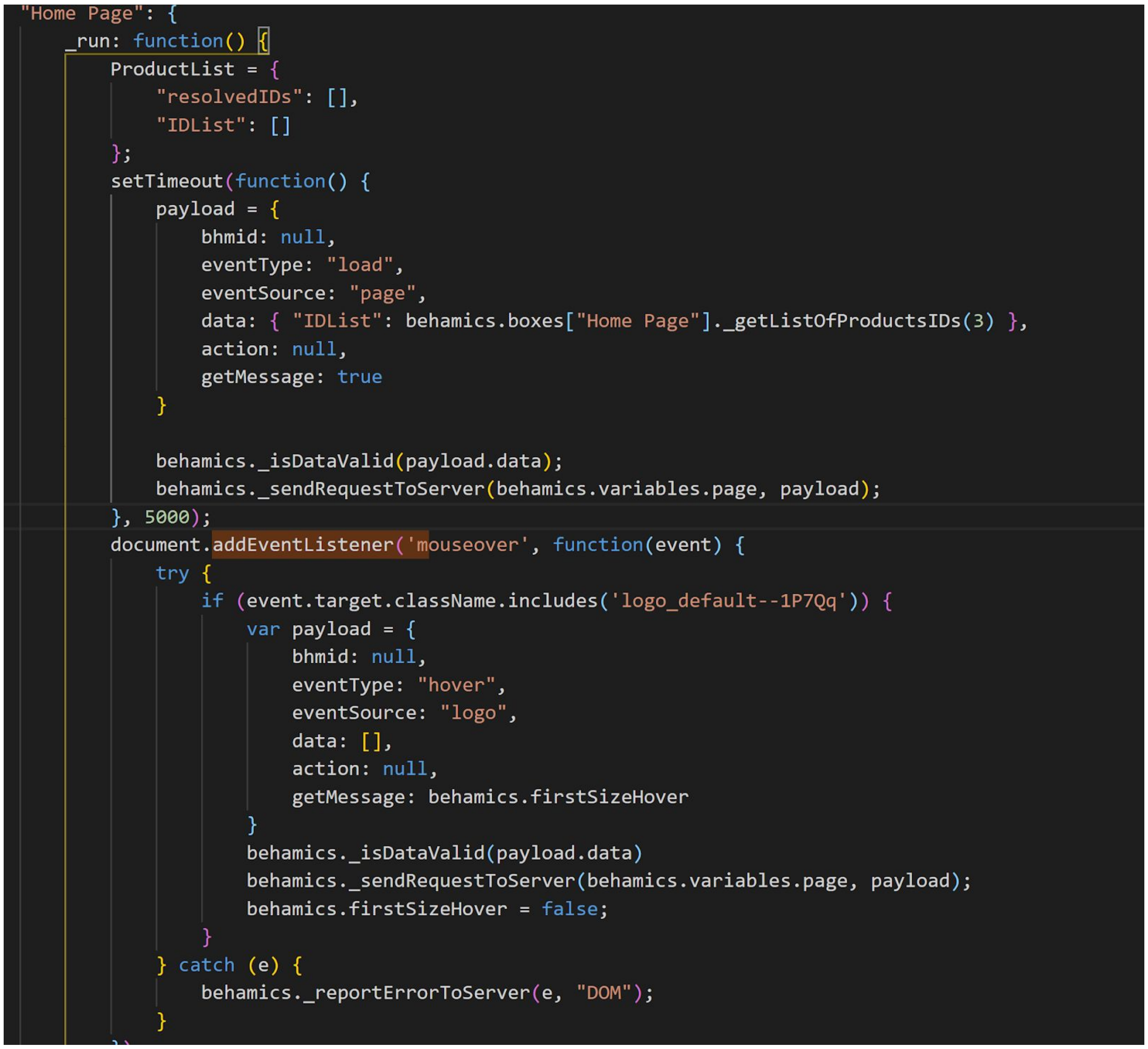
7. Modify the “Product Page”, (url: https://www.ochsnersport.ch/de/shop/46-nord-damen-softshelljacke-coral-00002001793549-p.html) method, *_getProduct*, so that it sends the list of available colors for a specific product as well. Hint: You need to find the list of available colors from HTML.


```
var variantColors = document.querySelectorAll('.color');
var colors = []
if (variantColors.length !== 0 ) {
    variantColors.forEach(element => {
        var alt = element.alt.split(' ');
        colors.push(alt[alt.length - 1])
    });
}
```


```
var tempob = {
    "id": digitalData.product.productInfo.productNr,
    "productID": digitalData.product.productInfo.productMasterNr,
    "name": digitalData.product.productInfo.productName.split("+").join(" ").replace("%2f", "/"),
    "price": price,
    "amount": 1,
    "currency": digitalData.product.attributes.currency,
    "formerPrice": formerPrice,
    "color": digitalData.product.attributes.colour,
    "category": completeProductCategory,
    "size": digitalData.product.productInfo.size,
    "sizeName": digitalData.product.productInfo.size,
    "url": window.location["href"],
    "imgUrl": digitalData.product.productInfo.productImage,
    "gender": gender,
    "variantColors": colors

};
```

8. Write the code through *addEventListener* that when in the “Home Page” (home page url: Ochsnersport.ch), when the user hovers with mouse on the logo of the shop which is placed on the menu bar, the codes sends a request to the server with standard payload format with the *eventType: “hover”*, *eventSource: “logo”*, and *data: []*.
We advise you to understand the script as in there you will find plenty of examples where the request is sent to the server with the right payload when an event takes place.



```
"Home Page": {
    _run: function() {
        ProductList = {
            "resolvedIDs": [],
            "IDList": []
        };
        setTimeout(function() {
            payload = {
                bhmid: null,
                eventType: "load",
                eventSource: "page",
                data: { "IDList": behamics.boxes["Home Page"]._getListOfProductsIDs(3) },
                action: null,
                getMessage: true
            }

            behamics._isDataValid(payload.data);
            behamics._sendRequestToServer(behamics.variables.page, payload);
        }, 5000);
        document.addEventListener('mouseover', function(event) {
            try {
                if (event.target.className.includes('logo_default--1P7Qq')) {
                    var payload = {
                        bhmid: null,
                        eventType: "hover",
                        eventSource: "logo",
                        data: [],
                        action: null,
                        getMessage: behamics.firstSizeHover
                    }
                    behamics._isDataValid(payload.data)
                    behamics._sendRequestToServer(behamics.variables.page, payload);
                    behamics.firstSizeHover = false;
                }
            } catch (e) {
                behamics._reportErrorToServer(e, "DOM");
            }
```