

Hostbridge HBJS Workshop III

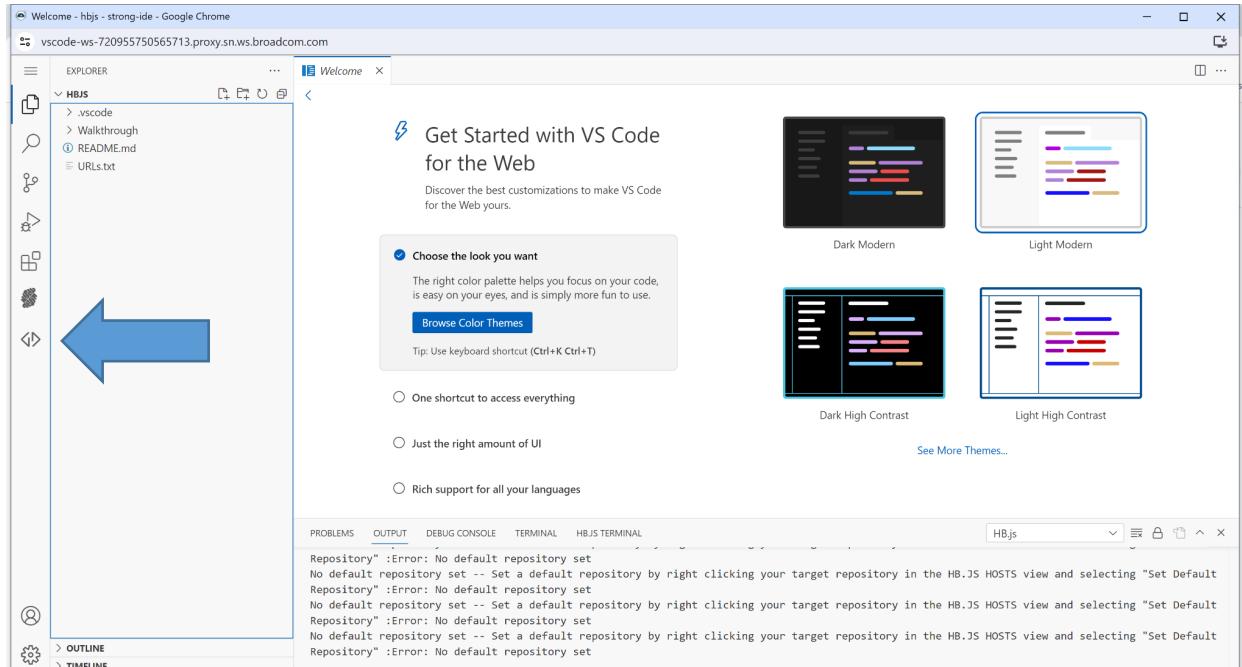


In this workshop you will create a script and modify it to have the Mainframe's CICS region to call an outside web service that will return weather reports.

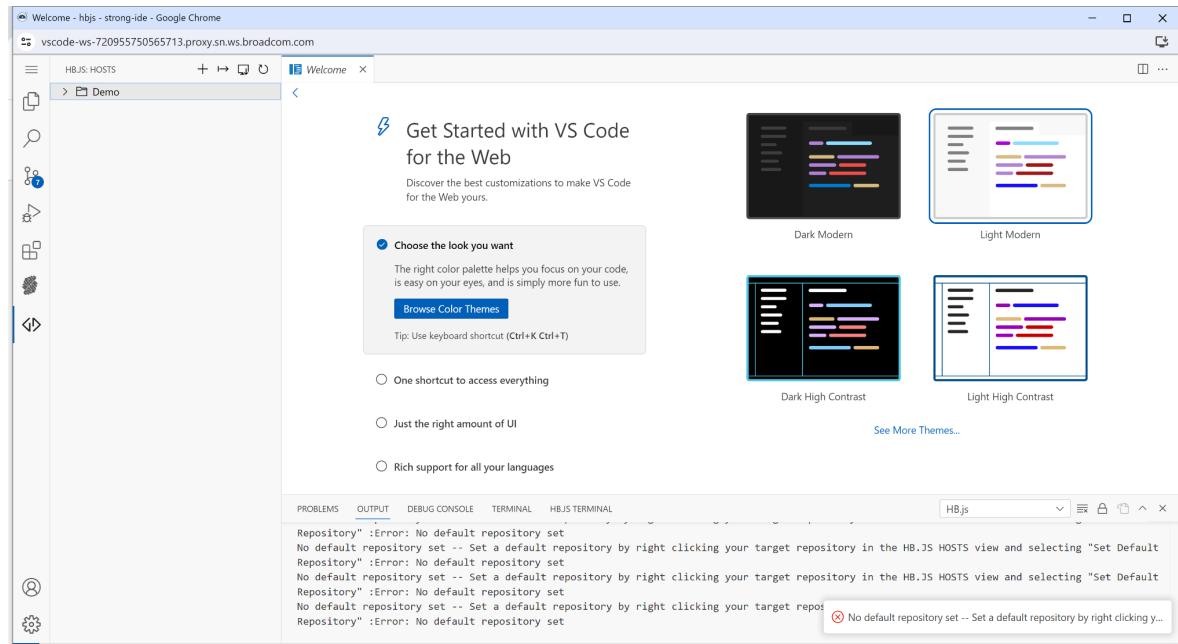
Configure:

The first part will be setting up the Hostbridge extension which has already been installed for you.

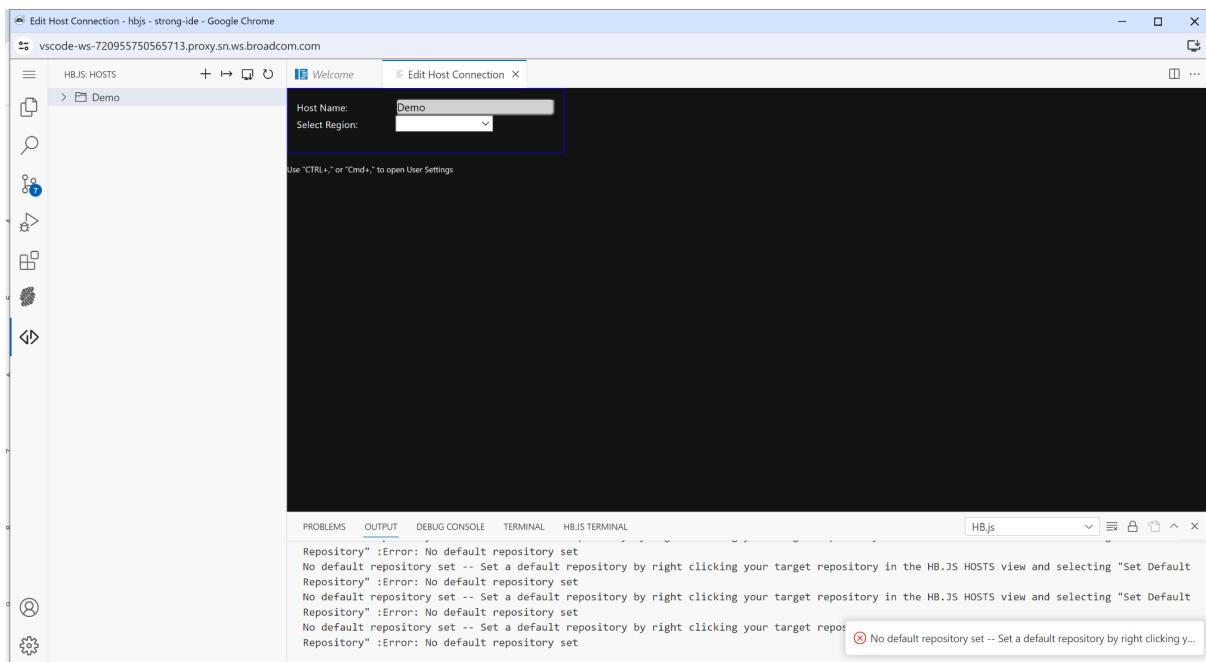
Click on the HB.js icon.



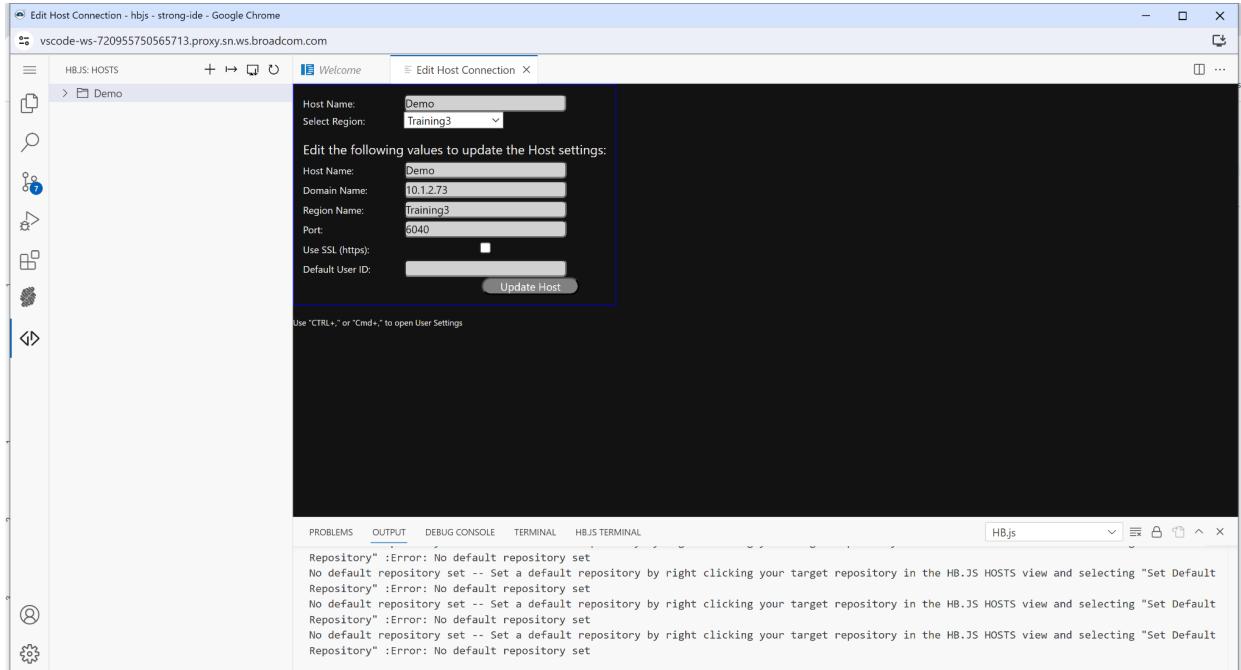
The following screen is returned.



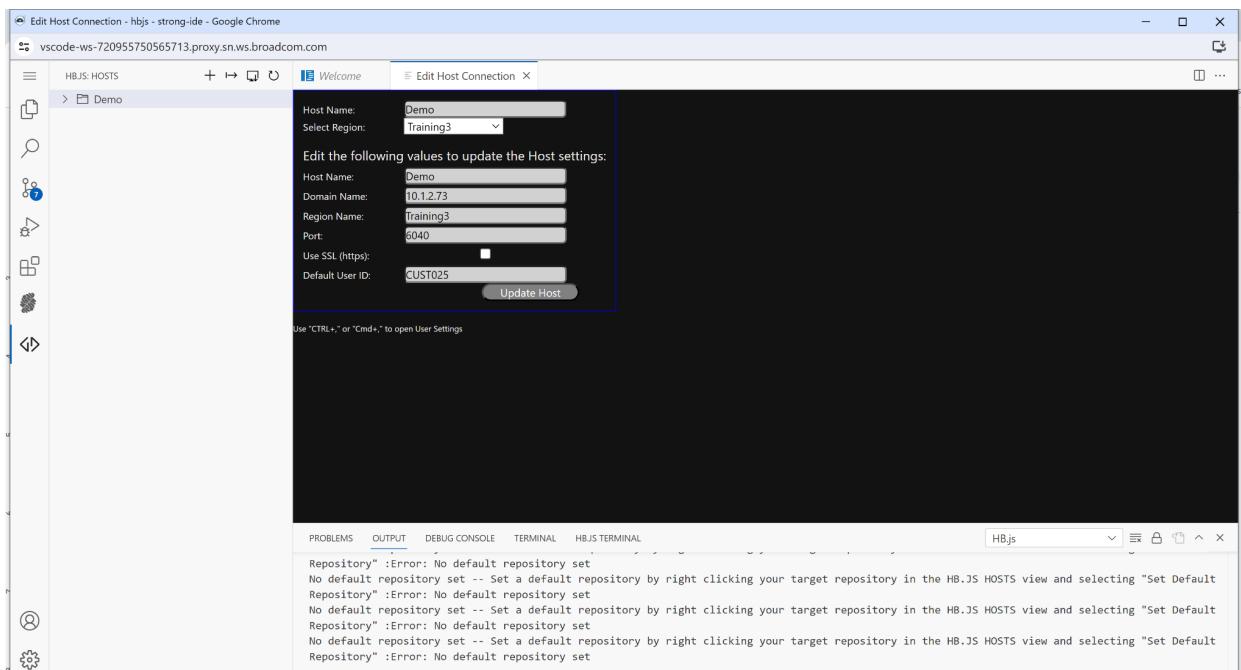
At this point we have already configured HB.JS to point to the mainframe and the CICS region. Right click on Demo > Edit Host so that we can add a default user ID. Afterwards, we will address the “No Default Repo... “ error.



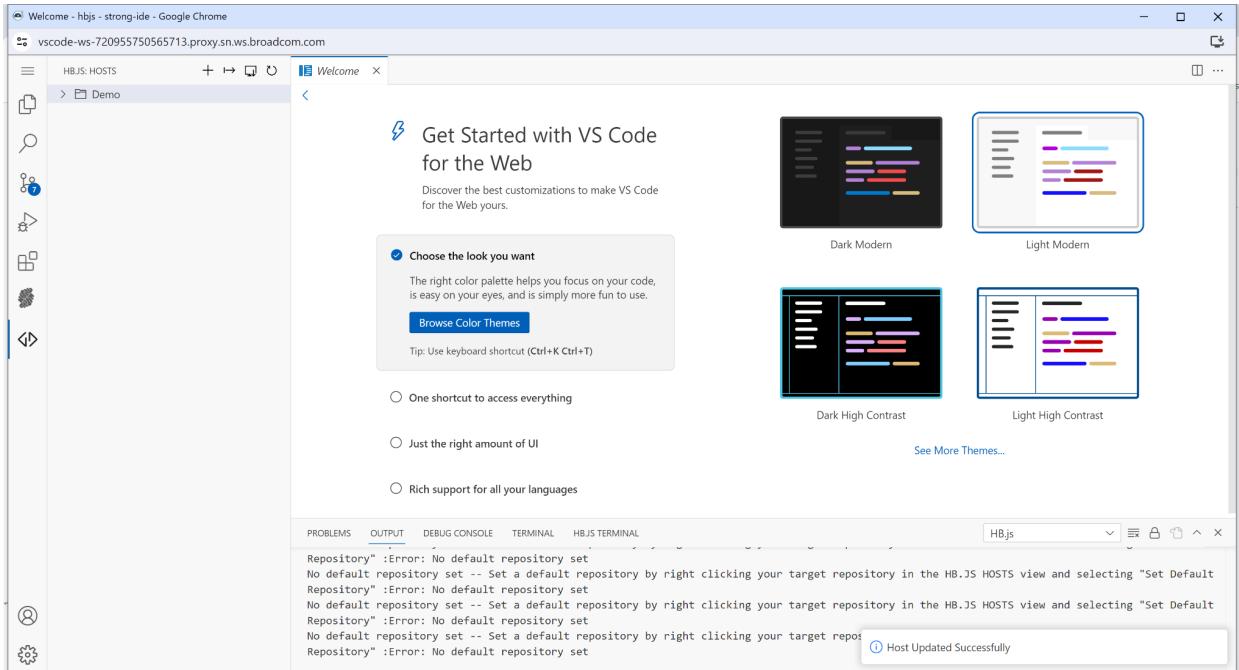
Select Training3 for the region.



Enter CUST0## for the Default User ID and then click Update Host. Please replace ## with your assigned UserID number.



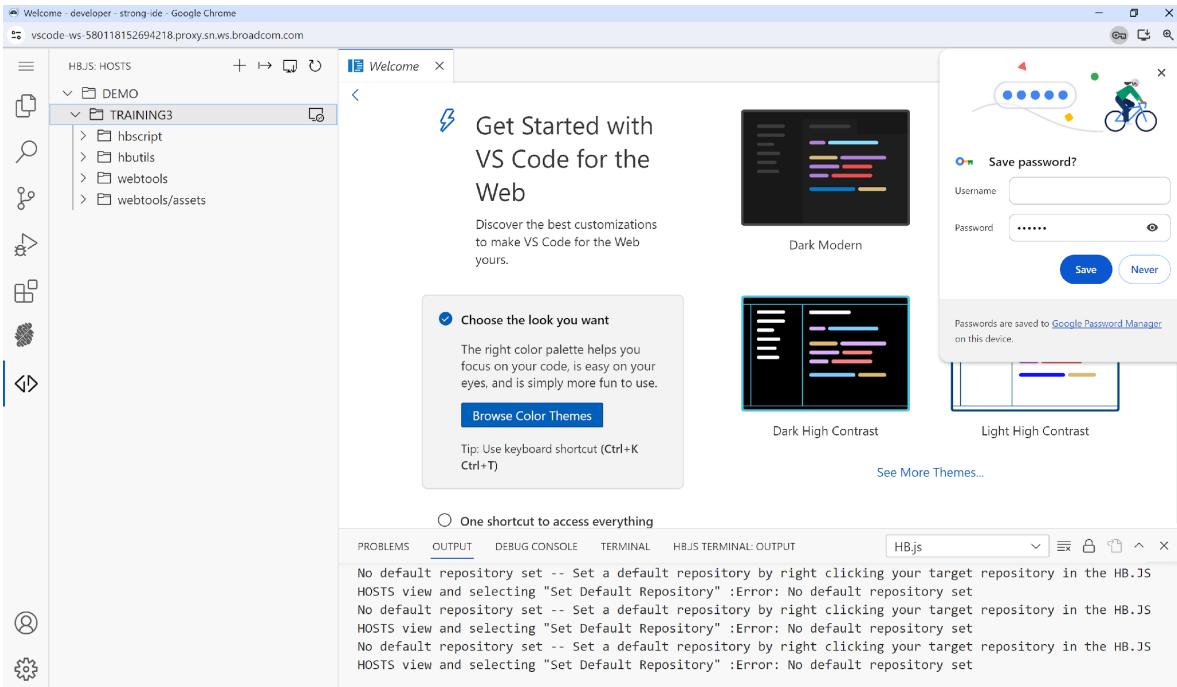
This will return the following screen. You should see the Host Updated Successfully message on the bottom right.



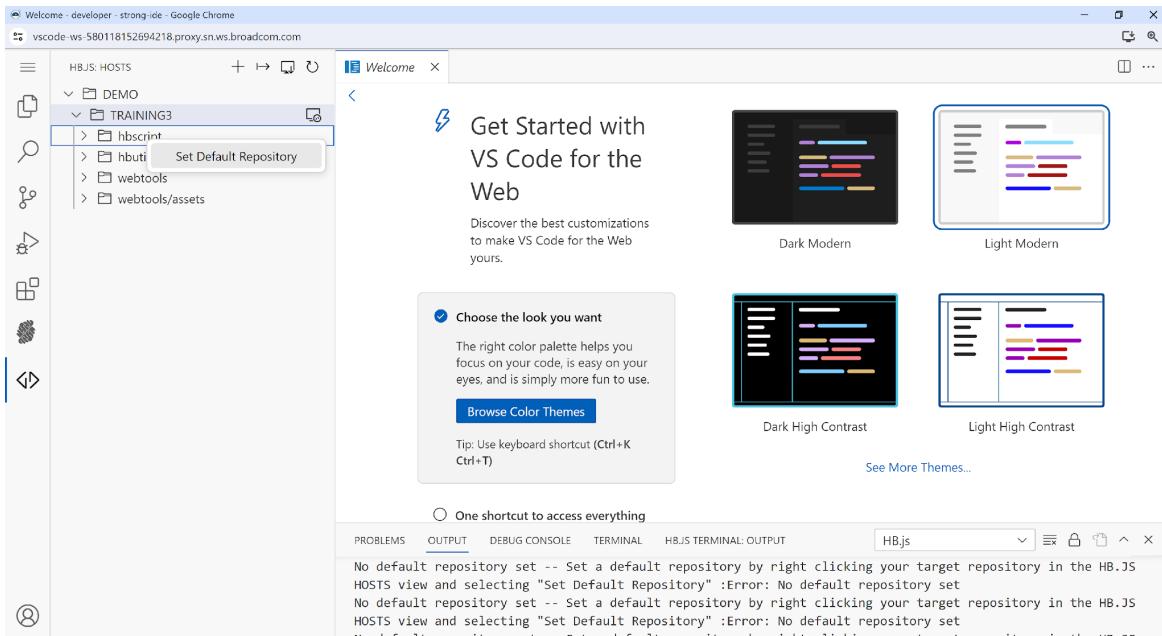
Now to deal with the error, “No default repository set ...”.

Expand the DEMO and TRAINING3 folders causing a pop-up box to appear. Here is where you enter your mainframe password CUST0##, where the ## is the number assigned to the workshop user. Press enter.

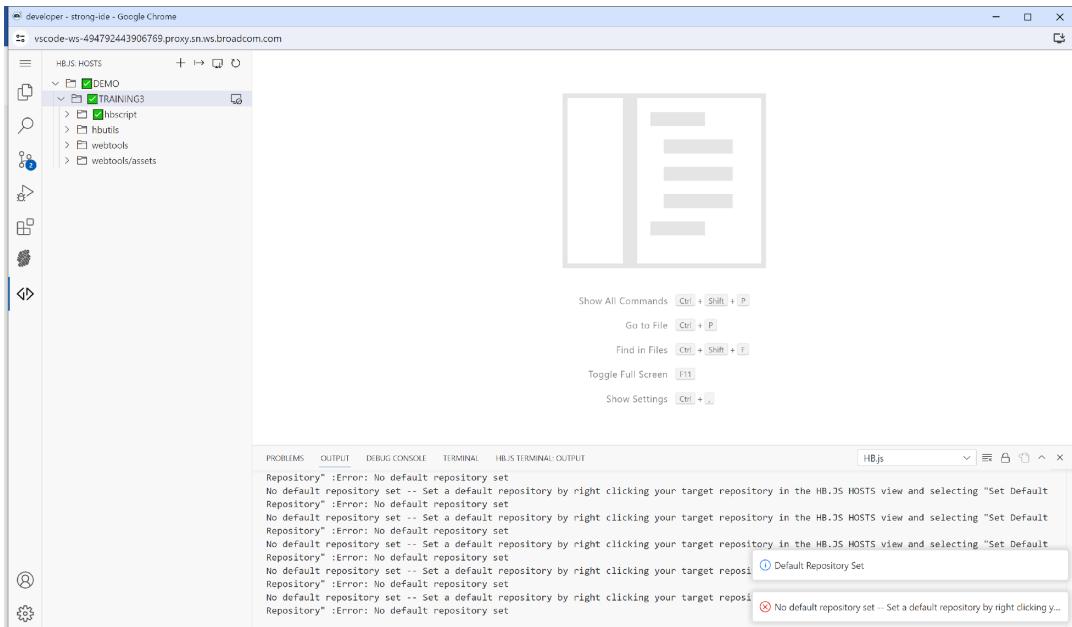
The following screen will be returned.



Right click the hbscript folder. Then click Set Default Repository

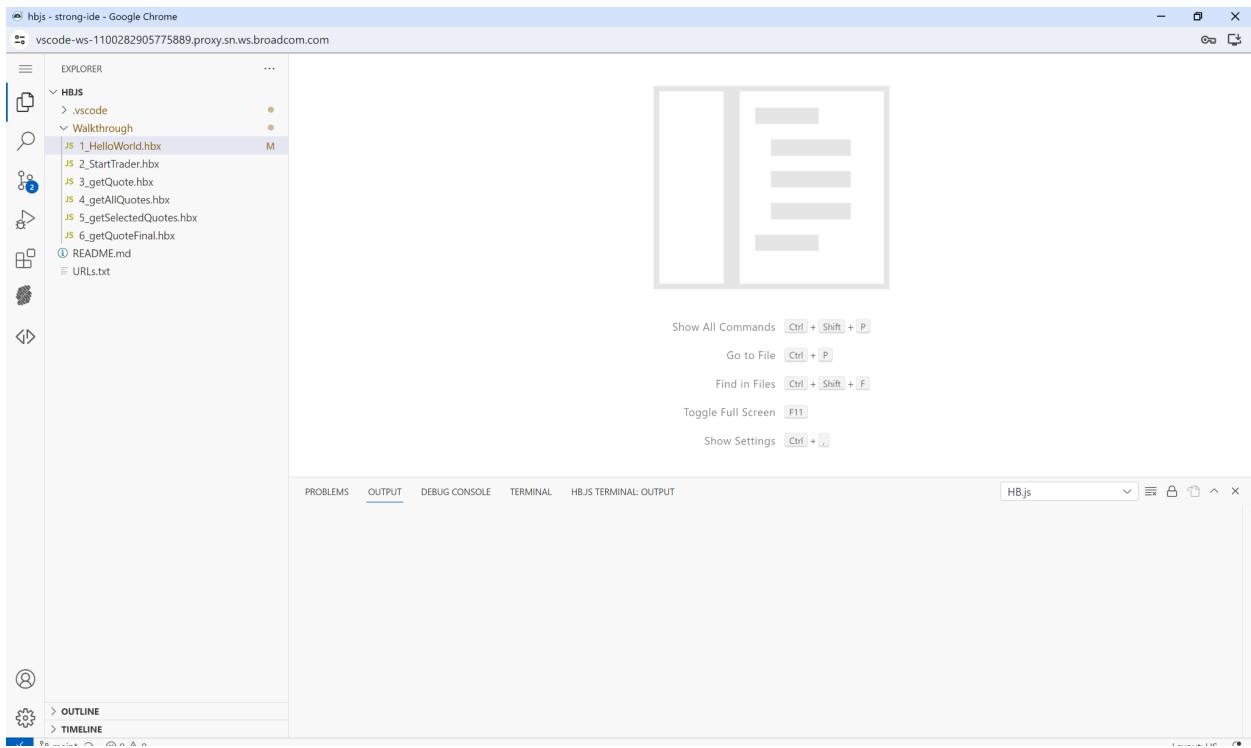


The following screen will be returned. Notice you now have a message stating, “Default Repository Set”.



Get exercises:

- 1 At this point you are ready to access the exercises for the workshop. Navigate to the explorer tab and expand the Walkthrough folder. The following screen is returned.



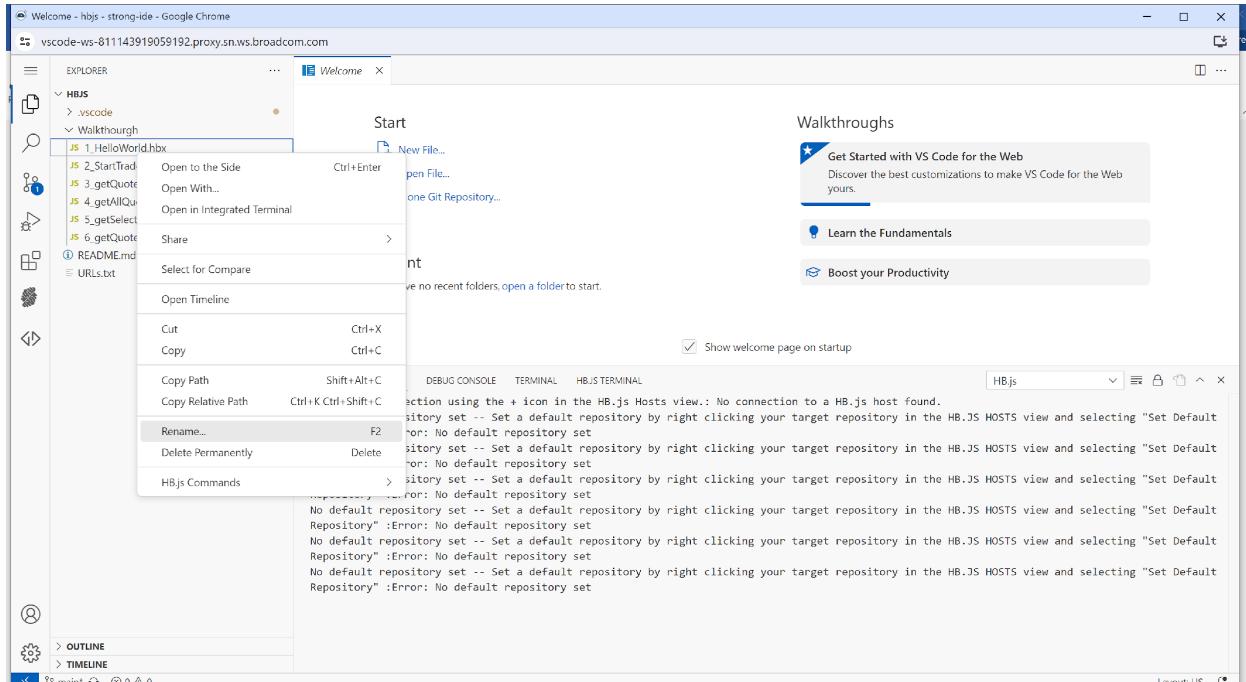
Workshop #1

The first workshop exercise is to change a line of Java Script. In the folder Walkthrough you will notice two files.

1-HelloWorld.hbx

2-StartTrader.hbx

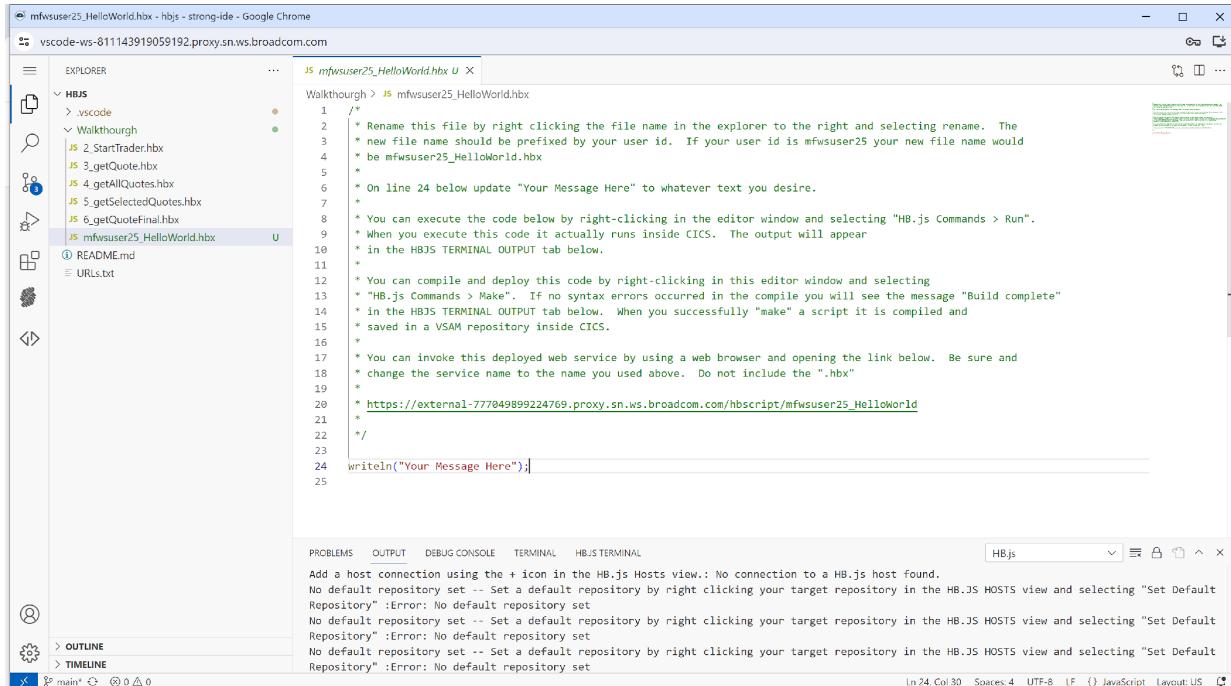
Rename the 1-HelloWorld.hbx, by right clicking the file and then selecting rename.



The new file name should be prefixed by your assigned userid **mfwsuser##** where the **##** is your assigned number. So the file name should look like:

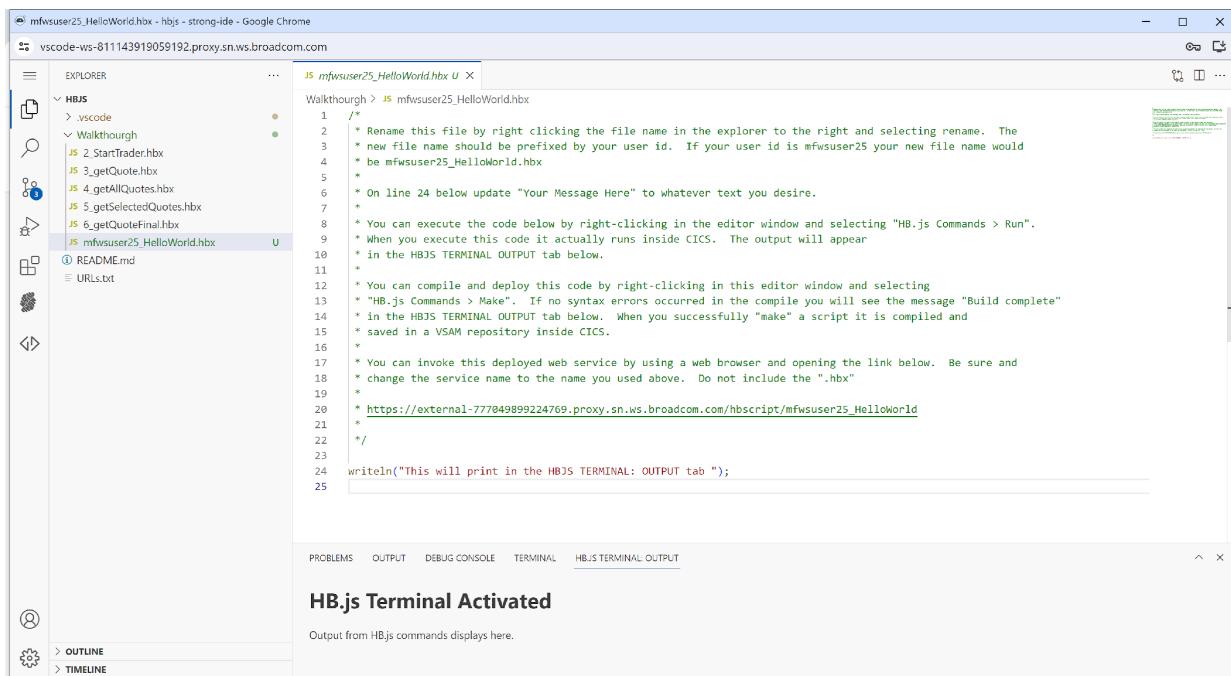
mfwsuser##_HelloWorld.hbx

The next step is to click on the file to open it and modify line 24:



```
1 /*  
2  * Rename this file by right clicking the file name in the explorer to the right and selecting rename. The  
3  * new file name should be prefixed by your user id. If your user id is mfwsuser25 your new file name would  
4  * be mfwsuser25_HelloWorld.hbx  
5  *  
6  * On line 24 below update "Your Message Here" to whatever text you desire.  
7  *  
8  * You can execute the code below by right-clicking in the editor window and selecting "HB.js Commands > Run".  
9  * When you execute this code it actually runs inside CICS. The output will appear  
10 * in the HBJS TERMINAL OUTPUT tab below.  
11 *  
12 * You can compile and deploy this code by right-clicking in this editor window and selecting  
13 * "HB.js Commands > Make". If no syntax errors occurred in the compile you will see the message "Build complete"  
14 * in the HBJS TERMINAL OUTPUT tab below. When you successfully "make" a script it is compiled and  
15 * saved in a VSAM repository inside CICS.  
16 *  
17 * You can invoke this deployed web service by using a web browser and opening the link below. Be sure and  
18 * change the service name to the name you used above. Do not include the ".hbx"  
19 *  
20 * https://external-777049899224769.proxy.sn.ws.broadcom.com/hbscript/mfwsuser25\_HelloWorld  
21 *  
22 */  
23  
24 writeln("Your Message Here");  
25
```

Change “Your Message Here” to the message you desire.



```
1 /*  
2  * Rename this file by right clicking the file name in the explorer to the right and selecting rename. The  
3  * new file name should be prefixed by your user id. If your user id is mfwsuser25 your new file name would  
4  * be mfwsuser25_HelloWorld.hbx  
5  *  
6  * On line 24 below update "Your Message Here" to whatever text you desire.  
7  *  
8  * You can execute the code below by right-clicking in the editor window and selecting "HB.js Commands > Run".  
9  * When you execute this code it actually runs inside CICS. The output will appear  
10 * in the HBJS TERMINAL OUTPUT tab below.  
11 *  
12 * You can compile and deploy this code by right-clicking in this editor window and selecting  
13 * "HB.js Commands > Make". If no syntax errors occurred in the compile you will see the message "Build complete"  
14 * in the HBJS TERMINAL OUTPUT tab below. When you successfully "make" a script it is compiled and  
15 * saved in a VSAM repository inside CICS.  
16 *  
17 * You can invoke this deployed web service by using a web browser and opening the link below. Be sure and  
18 * change the service name to the name you used above. Do not include the ".hbx"  
19 *  
20 * https://external-777049899224769.proxy.sn.ws.broadcom.com/hbscript/mfwsuser25\_HelloWorld  
21 *  
22 */  
23  
24 writeln("This will print in the HBJS TERMINAL: OUTPUT tab ");  
25
```

HB.js Terminal Activated

Output from HB.js commands displays here.

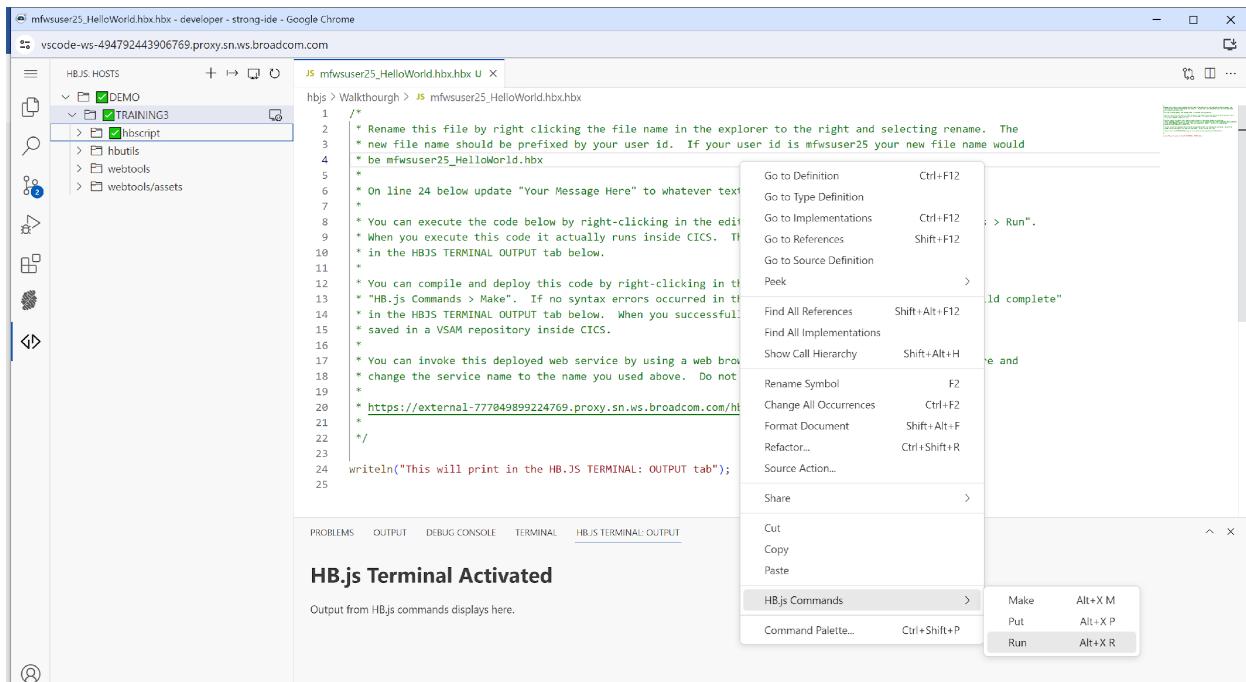
To execute the script right click anywhere inside the code and hover over HB.js Commands, notice three options:

Make – Places the script on the mainframe and compiles it. It will only put the script on the mainframe if there is a clean compile.

Put – Places the script on the mainframe

Run – executes the script

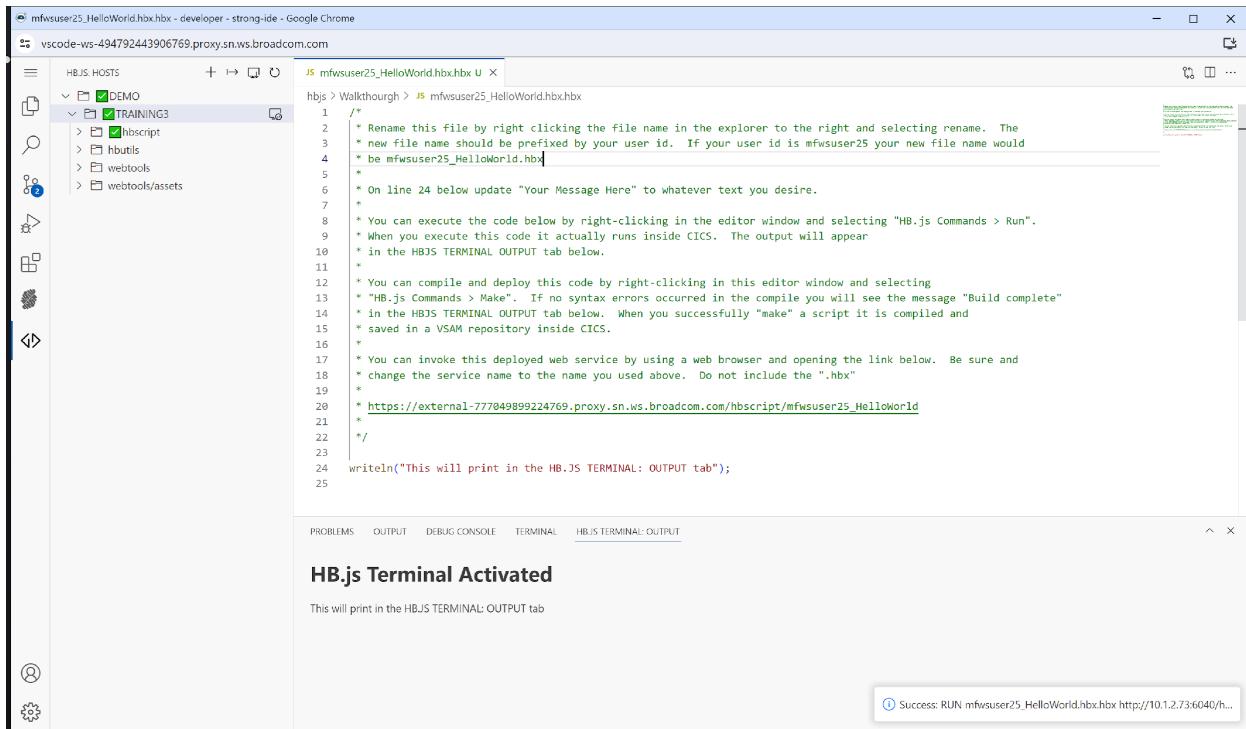
You can also use the listed hot keys if that is your preference.



Select the HB.JS TERMINAL:OUTPUT tab and then Click HB.js Commands > Run (the tab need only be selected once per session, after that it has been activated and will open for you when output is printed to it).

When you execute the script it is actually running inside the CICS region.

See your message in the HB.JS TERMINAL:OUTPUT and the Success prompt on the bottom right.



The screenshot shows a Microsoft Visual Studio Code interface. The title bar reads "mfwsuser25_HelloWorld.hbx - developer - strong-ide - Google Chrome". The left sidebar shows a tree view of "HB.JS HOSTS" with "DEMO" and "TRAINING3" expanded, and "hbscript" selected. The main editor area displays a file named "hbscript" with the following content:

```
1 /*  
2 * Rename this file by right clicking the file name in the explorer to the right and selecting rename. The  
3 * new file name should be prefixed by your user id. If your user id is mfwsuser25 your new file name would  
4 * be mfwsuser25_HelloWorld.hbx  
5 *  
6 * On line 24 below update "Your Message Here" to whatever text you desire.  
7 *  
8 * You can execute the code below by right-clicking in the editor window and selecting "HB.js Commands > Run".  
9 * When you execute this code it actually runs inside CICS. The output will appear  
10 * in the HBJS TERMINAL OUTPUT tab below.  
11 *  
12 * You can compile and deploy this code by right-clicking in this editor window and selecting  
13 * "HB.js Commands > Make". If no syntax errors occurred in the compile you will see the message "Build complete"  
14 * in the HBJS TERMINAL OUTPUT tab below. When you successfully "make" a script it is compiled and  
15 * saved in a VSAM repository inside CICS.  
16 *  
17 * You can invoke this deployed web service by using a web browser and opening the link below. Be sure and  
18 * change the service name to the name you used above. Do not include the ".hbx"  
19 *  
20 * https://external-777049899224769.proxy.sn.ws.broadcom.com/hbscript/mfwsuser25\_HelloWorld  
21 *  
22 */  
23  
24 writeln("This will print in the HB.JS TERMINAL: OUTPUT tab");  
25
```

Below the editor, tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "HB.JS TERMINAL: OUTPUT" are visible. The "HB.JS TERMINAL: OUTPUT" tab is active, displaying the message "HB.js Terminal Activated". A tooltip at the bottom right says "Success: RUN mfwsuser25_HelloWorld.hbx.hbx http://10.1.2.73:6040/h...".

Next right click inside the code and click Make.

HBJS HOSTS

hbjs > Walkthrough > JS mfwuser25_HelloWorld.hbx.hbx

```

1 /*
2 * Rename this file by right clicking the file name in the explorer to the right and selecting rename. The
3 * new file name should be prefixed by your user id. If your user id is mfwuser25 your new file name would
4 * be mfwuser25_HelloWorld.hbx
5 *
6 * On line 24 below update "Your Message Here" to whatever text you desire.
7 *
8 * You can execute the code below by right-clicking in the editor window and selecting "HB.js Commands > Run".
9 * When you execute this code it actually runs inside CICS. The output will appear
10 * in the HBJS TERMINAL OUTPUT tab below.
11 *
12 * You can compile and deploy this code by right-clicking in this editor window and selecting
13 * "HB.js Commands > Make". If no syntax errors occurred in the compile you will see the message "Build complete"
14 * in the HBJS TERMINAL OUTPUT tab below. When you successfully "make" a script it is compiled and
15 * saved in a VSAM repository inside CICS.
16 *
17 * You can invoke this deployed web service by using a web browser and opening the link below. Be sure and
18 * change the service name to the name you used above. Do not include the ".hbx"
19 *
20 * https://external-777049899224769.proxy.sn.ws.broadcom.com/hbscript/mfwuser25\_HelloWorld
21 *
22 */
23
24 writeln("This will print in the HB.JS TERMINAL: OUTPUT tab");
25

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL HBJS TERMINAL

HB.js Terminal Activated

This will print in the HBJS TERMINAL: OUTPUT tab

HB.js Commands >

- Make Alt+X M
- Put Alt+X P
- Run Alt+X R

If there are no errors you will have compiled and copied the code to the mainframe. Notice the message in the HB.JS TERMINAL:OUTPUT.

HBJS HOSTS

hbjs > Walkthrough > JS mfwuser25_HelloWorld.hbx.hbx

```

1 /*
2 * Rename this file by right clicking the file name in the explorer to the right and selecting rename. The
3 * new file name should be prefixed by your user id. If your user id is mfwuser25 your new file name would
4 * be mfwuser25_HelloWorld.hbx
5 *
6 * On line 24 below update "Your Message Here" to whatever text you desire.
7 *
8 * You can execute the code below by right-clicking in the editor window and selecting "HB.js Commands > Run".
9 * When you execute this code it actually runs inside CICS. The output will appear
10 * in the HBJS TERMINAL OUTPUT tab below.
11 *
12 * You can compile and deploy this code by right-clicking in this editor window and selecting
13 * "HB.js Commands > Make". If no syntax errors occurred in the compile you will see the message "Build complete"
14 * in the HBJS TERMINAL OUTPUT tab below. When you successfully "make" a script it is compiled and
15 * saved in a VSAM repository inside CICS.
16 *
17 * You can invoke this deployed web service by using a web browser and opening the link below. Be sure and
18 * change the service name to the name you used above. Do not include the ".hbx"
19 *
20 * https://external-777049899224769.proxy.sn.ws.broadcom.com/hbscript/mfwuser25\_HelloWorld
21 *
22 */
23
24 writeln("This will print in the HB.JS TERMINAL: OUTPUT tab");
25

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL HBJS TERMINAL: OUTPUT

HB.js Terminal Activated

make success to DEMO:TRAINING3:hbscript
MAKE2 00010 Build complete total length= 236.
mfwuser25_HelloWorld.hbx:0: Info: Script make successful in repository hbscript. Stored size is 236 bytes.

Success: MAKE mfwuser25_HelloWorld.hbx.hbx http://10.1.2.73:6040...

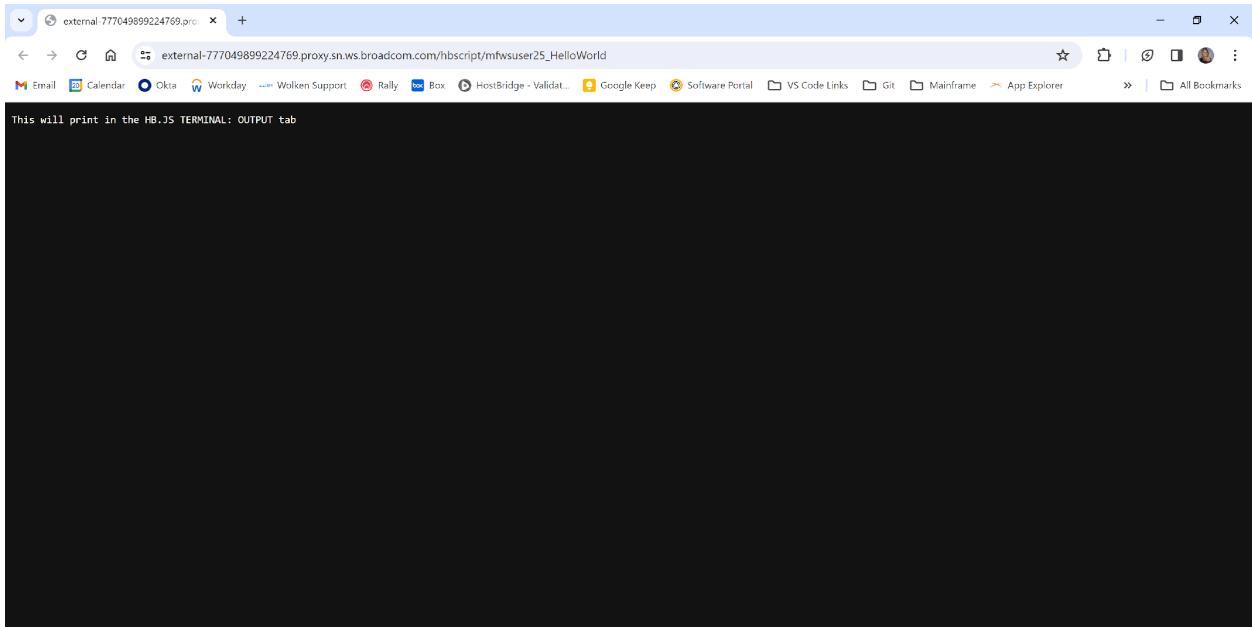
Next would be to invoke this service using a web browser and opening the following link. Note: Replace ## in the link with the User number allocated to you.

[https://external-
157476462544637.proxy.sn.ws.broadcom.com/hbscript/mfwsus
er##_HelloWorld](https://external-157476462544637.proxy.sn.ws.broadcom.com/hbscript/mfwsuser##_HelloWorld)

For example:

[https://external-
157476462544637.proxy.sn.ws.broadcom.com/hbscript/mfwsus
er25>HelloWorld](https://external-157476462544637.proxy.sn.ws.broadcom.com/hbscript/mfwsuser25>HelloWorld)

The following screen should be returned.



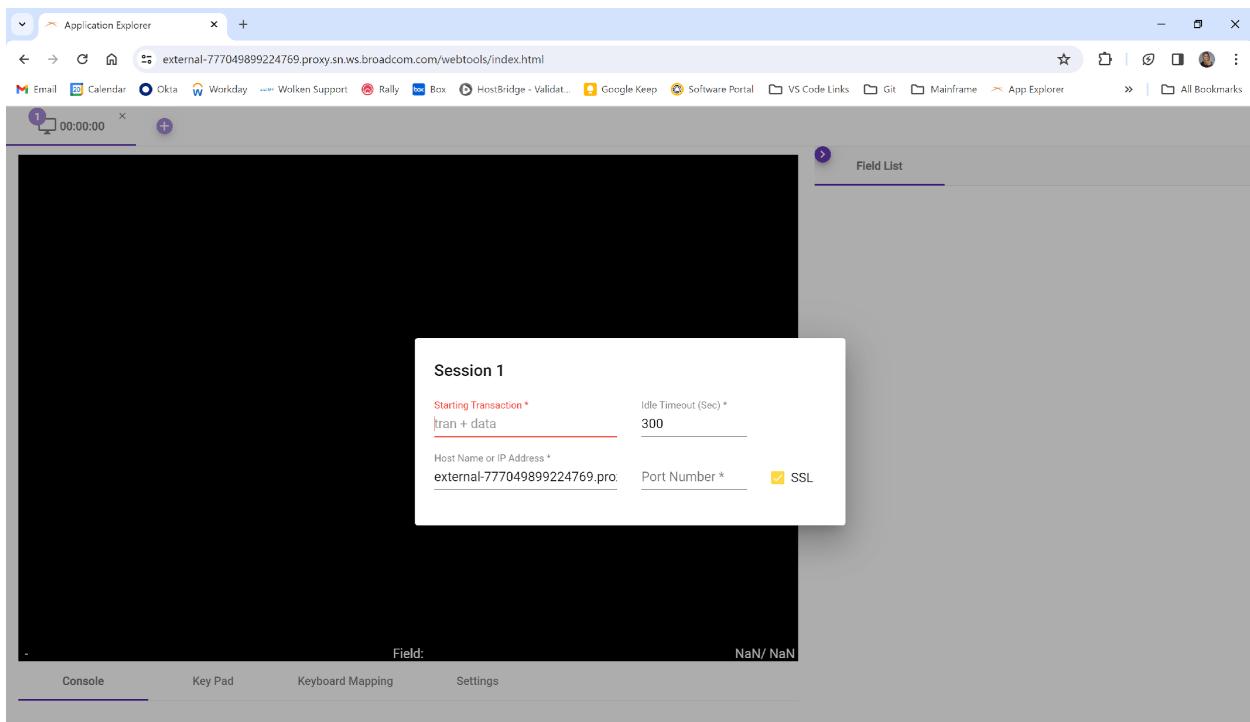
Congratulations you have completed the first exercise.

Workshop #2

Repeat the renaming steps in Workshop #1 on 2-StartTrader.hbx.

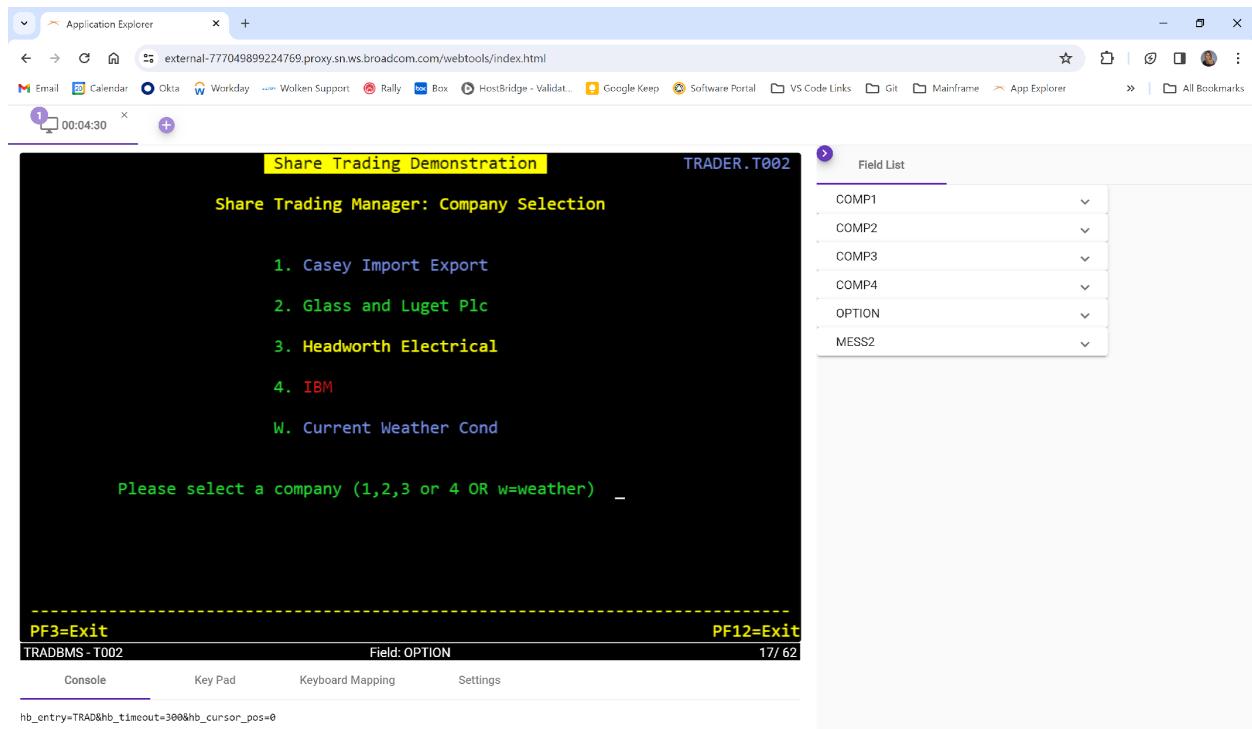
Once the file has been renamed we are going to use the utility Application Explorer. To use this utility open a browser and issue the following link.

<https://external-157476462544637.proxy.sn.ws.broadcom.com/webtools/index.html>

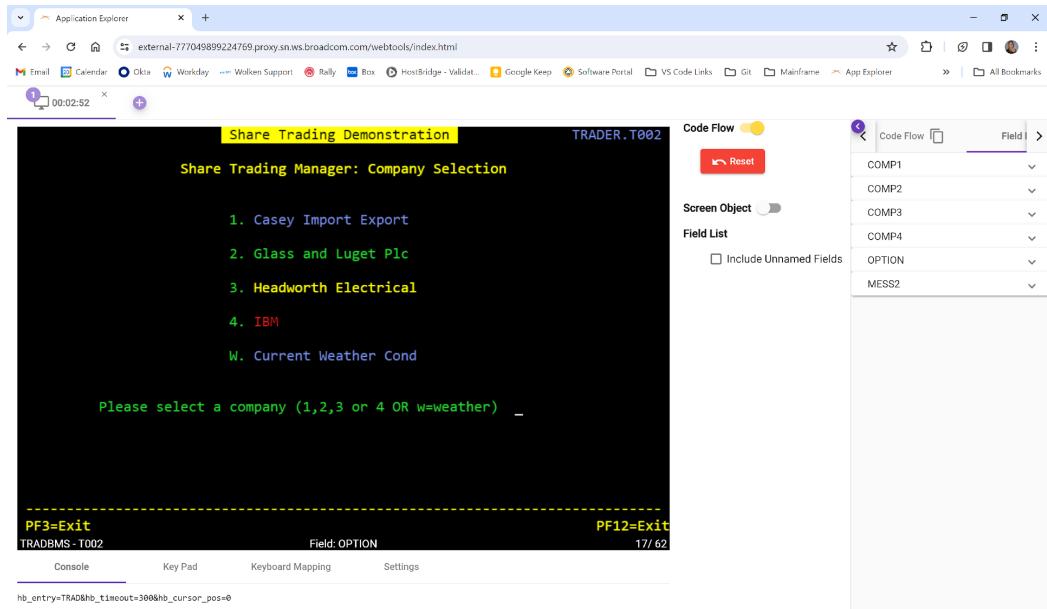


On the Starting Transaction field enter TRAD.

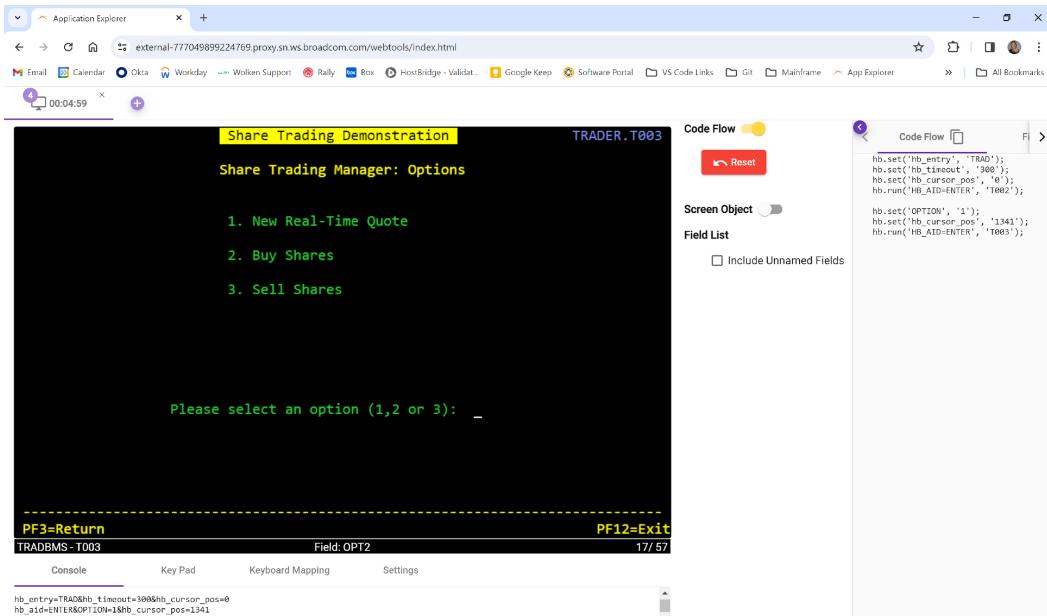
Press enter to navigate to the below screen.



One feature HB.js has is its Code Flow capabilities. Click on the purple arrow in the circle and turn it on by clicking on the slide bar.



At this point you are walking through an application and recording the steps. Enter a 1 on the Please select a company field and press enter.



The screen changes to Share Trading Demonstration but in the Code Flow you see the HB.js commands.

Enter a 1 on the Please select an option field and press enter. See how the Code Flow has changed.

Place your cursor on the Number of Shares Held field and double click. This will add the field to the Code Flow.

Repeat again with the Value of Shares Hold field.

Share Trading Demonstration

Share Trading Manager: Real-Time Quote

User Name: CICSUSER

Company Name: Casey Import Export

Share Values:	Commission Cost:
NOW: 00079.00	for Selling: 007
1 week ago: 00059.00	for Buying: 010
6 days ago: 00063.00	
5 days ago: 00065.00	
4 days ago: 00070.00	
3 days ago: 00072.00	
2 days ago: 00078.00	Number of Shares Held: 0504
1 day ago: 00077.00	Value of Shares Held: 000039816.00

Request Completed OK

PF3=Return PF12=Exit

TRADBMS - T004 Field: field_r17_c40 17/ 56

Console Key Pad Keyboard Mapping Settings

```
hb_entry='1440000_13341=13340_cursor_pos->
hb_aid=ENTER&OPTION=1&hb_cursor_pos=1341
hb_aid=ENTER&OPTION=1&hb_cursor_pos=1337
```

Now place your cursor on the value of the Value of Shares Held field which is 000039737 and double click. This will add the value to the Code Flow.

Share Trading Demonstration

Share Trading Manager: Real-Time Quote

User Name: CICSUSER

Company Name: Casey Import Export

Share Values:	Commission Cost:
NOW: 00079.00	for Selling: 007
1 week ago: 00059.00	for Buying: 010
6 days ago: 00063.00	
5 days ago: 00065.00	
4 days ago: 00070.00	
3 days ago: 00072.00	
2 days ago: 00078.00	Number of Shares Held: 0504
1 day ago: 00077.00	Value of Shares Held: 000039816.00

Request Completed OK

PF3=Return PF12=Exit

TRADBMS - T004 Field: VALUE 17/ 65

Console Key Pad Keyboard Mapping Settings

```
hb_entry='1440000_13341=13340_cursor_pos->
hb_aid=ENTER&OPTION=1&hb_cursor_pos=1341
hb_aid=ENTER&OPTION=1&hb_cursor_pos=1337
```

At this point HB.js has written the code for you. All you have to do is copy it and paste it into the script between the following lines:

```
// ----- Paste the copied code below here
```

```
// ----- Paste the copied code above here
```

So it will look like the following:

```
// ----- Paste the copied code below here
```

```
hb.set('hb_entry', 'trad');  
hb.set('hb_timeout', '300');  
hb.set('hb_cursor_pos', '0');  
hb.run('HB_AID=ENTER', 'T002');
```

```
hb.set('OPTION', '1');  
hb.set('hb_cursor_pos', '1341');  
hb.run('HB_AID=ENTER', 'T003');
```

```
hb.set('OPT2', '1');  
hb.set('hb_cursor_pos', '1337');  
hb.run('HB_AID=ENTER', 'T004');
```

```
hb.getFieldValue('field_r16_c40');  
hb.getFieldValue('field_r17_c40');  
hb.getFieldValue('VALUE');
```

// ----- Paste the copied code above here

At this point you can right click the script and issue the HB.js Commands > Run

The screenshot shows a VS Code interface with the following details:

- Explorer:** Shows files like .cache, .docker, .local, .npm, .sdkman, .ssh, .vscode, and hbjs.
- Editor:** Two tabs are open: "JS mfwuser25_HelloWorld.hbx" and "JS mfwuser25_StartTrader.hbx". The "mfwuser25_StartTrader.hbx" tab contains the following code:

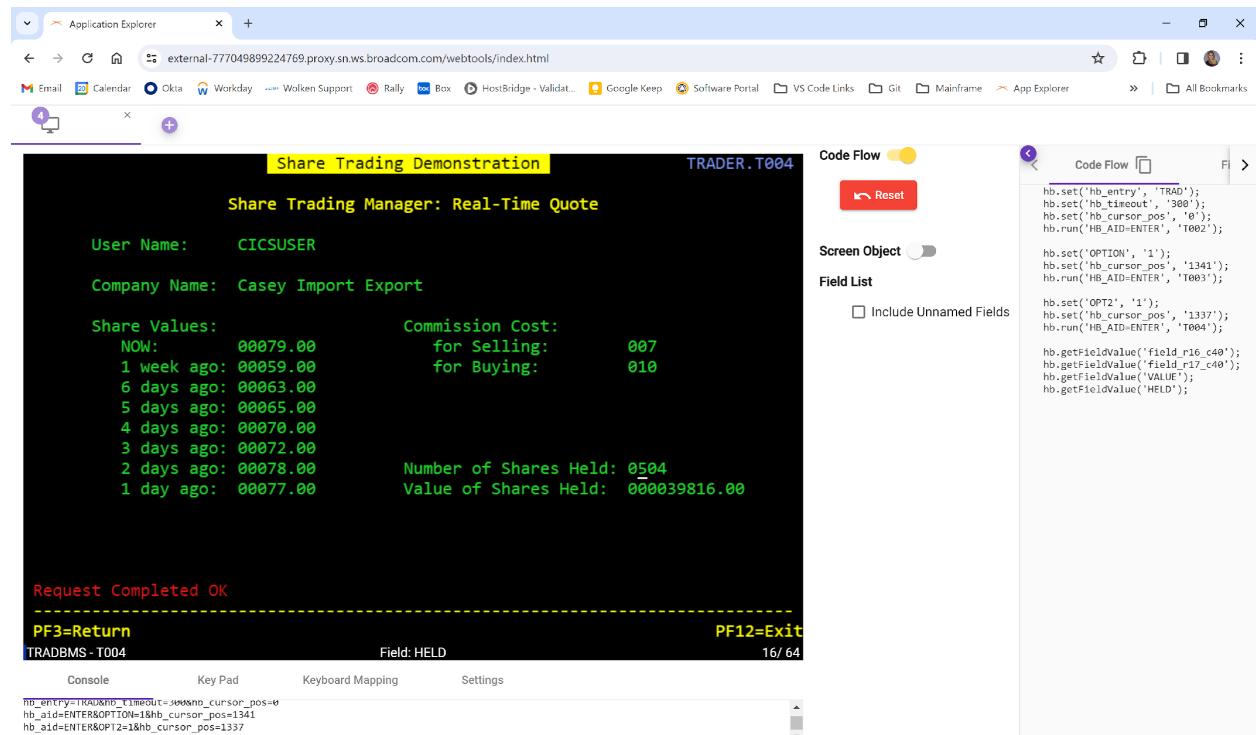
```
hbjs > Walkthrough > JS mfwuser25_StartTrader.hbx > ...
44
45 hb.set('OPT2', '1');
46 hb.set('hb_cursor_pos', '1337');
47 hb.run('HB_AID=ENTER', '1004');
48
49 hb.getFieldValue('field_r16_c40');
50 hb.getFieldValue('field_r17_c40');
51 hb.getFieldValue('VALUE');
52
53 // ----- Paste the copied code above
54
55
56 // Right-click in this editor window and select
57 // this should be displayed in the HBJS TERMINAL
58
59 writeln('Total Value of Shares Held = ', hb.get
60
61 /*
62 * Using the line above as a sample see if you
63 * can
64 * Field names like VALUE and HELD are field
65 * written. We are not screen scraping but
66 * chose.
67 */
68
```
- Terminal:** Shows the message "HB.js Terminal Activated". Below it, the output "Total Value of Shares Held = 000039816.00" is displayed.
- Context Menu:** A context menu is open over the code, with the "HB.js Commands" option highlighted. Other options include Go to Definition, Find All References, and Run.
- Status Bar:** Shows "Success: RUN mfwuser25_StartTrader.hbx http://10.1.2.73:6040/hbscri..."

You should see Total Value of Shares Held =
000039737.00

Once you have completed this part of the exercise add another output field Number of Shares Held. To do this you

have to go back to the Application Explorer screen and double click the value for Number of Shares Held field.

When you double click the value you will see the HB.js command added to the Code Flow.



hb.getFieldValue('HELD');

In the script let's add the HELD value to the output. To do this repeat line:

```
writeln('Total Value of Shares Held = ', hb.getFieldValue('VALUE'));
and change the wording to 'Number of Shares Held'
and VALUE to HELD
```

When you are finished it should look like the following.

```
writeln('Total Value of Shares Held = ', hb.getFieldValue('VALUE'));
writeln('Number of Shares Held = ', hb.getFieldValue('HELD'));
```

Repeat the above instructions on running the script and notice the second line of output.

The complete script should look close to the following:

```
/*
 * Rename this file by right clicking the file name in the explorer to the right
 and selecting rename. The
 * new file name should be prefixed by your user id. If your user id is
mfwsuser25 your new file name would
 * be mfwsuser25_StartTrader.hbx
*/
```

```

// Create the hb object that can interact with CICS transactions
let hb = new HB.Session();

// "require" JavaScript packages with predefined utilities that we will use when
// creating services
let common = require('common', 'hbutils');
let debugging = require('debugging', 'hbutils');

// Using utilities out of the packages above to setup some default error checking
// and debugging
debugging.checkDebugControl();
common.hbRunProto();

/*
 * Using the Application Explorer, run the TRAD transaction and get the Real-Time
 * Quote for Casey Import Export.
 * Be sure and turn on "Code Flow" by clicking the purple arrow and clicking the
 * toggle switch.
 *
 * After starting TRAD:
 *      Select Casey Import Export by entering a "1" in the input field and press
 * enter.
 *      Select New Real-Time Quote by entering a "1" in the input field and press
 * enter.
 *
 * On the "Real-Time Quote" screen double click on any fields of interest, like
 * "Number of Shares Held" and
 * "Value of Shares Held". Be sure and click on the values not the field
 * labels. Double clicking on these
 * fields will add them to the "Code Flow".
 *
 * Application Explorer: https://external-
157476462544637.proxy.sn.ws.broadcom.com/webtools/index.html
 *
 * Copy the code generated in "Code Flow" and paste it below
 */

```

// ----- Paste the copied code below here

```

hb.set('hb_entry', 'trad');
hb.set('hb_timeout', '300');
hb.set('hb_cursor_pos', '0');
hb.run('HB_AID=ENTER', 'T002');

hb.set('OPTION', '1');
hb.set('hb_cursor_pos', '1341');
hb.run('HB_AID=ENTER', 'T003');

```

```

hb.set('OPT2', '1');
hb.set('hb_cursor_pos', '1337');
hb.run('HB_AID=ENTER', 'T004');

hb.getFieldValue('field_r16_c40');
hb.getFieldValue('field_r17_c40');
hb.getFieldValue('VALUE');

// ----- Paste the copied code above here

// Right-click in this editor window and select "HB.js Commands > Run"  The total
// value of shares held for Casey Import Export
// should be displayed in the HBJS TERMINAL OUTPUT below

writeln('Total Value of Shares Held = ', hb.getFieldValue('VALUE'));
writeln('Number of Shares Held = ', hb.getFieldValue('HELD'));

/*
 * Using the line above as a sample see if you can output other fields off the
"Real-Time Quote" screen.
 *
 * Field names like VALUE and HELD are field names the programmer chose when TRAD
was originally
 * written.  We are not screen scraping but retrieving values using the field
names the programmer
 * chose.
 */

```

Congratulations you have completed Workshop #2

Workshop #3

In this workshop you will have to change a script to execute the TRAD transactions twice for two companies. To do this a new HB.js utility function is added which will allow you to issue a PF3 command to back up one screen at a time.

Repeat the renaming steps in Workshop #1 on 3-getQuote.hbx.

As you review the script you will notice that starting at line 62 there is a function called getQuote and it takes companyNumber in as input. But as you walk through the code it will fail because of the missing F3 or PF3 command. Also notice how a screen name check has been added ‘T003’ and ‘T004’. So when a value is entered on screen to change screens, this check will make sure you are on the correct screen.

Follow the instructions provided in your file to make the required edits.

```

function initialize() {
    hb.set('hb_entry', 'trad');
    hb.run('HB_AID=ENTER', 'T002');
}

function getQuote(companyNumber) {
    hb.set('OPTION', companyNumber);
    hb.run('HB_AID=ENTER', 'T003');

    hb.set('OPT2', '1');
    hb.run('HB_AID=ENTER', 'T004');

    writeln('Company Name: ', hb.getFieldValue('COMP41'));
    writeln('Total Value of Shares Held = ', hb.getFieldValue('VALUE'));
    writeln();
}

initialize();
getQuote('1');
getQuote('2');

```

The complete script should look close to the following:

```

/*
 * Rename this file by right clicking the file name in the explorer to the right
 and selecting rename. The
 * new file name should be prefixed by your user id. If your user id is
mfwuser25 your new file name would
 * be mfwuser25_getQuote.hbx
 */
let hb = new HB.Session();

let common = require('common', 'hbutils');
let debugging = require('debugging', 'hbutils');

debugging.checkDebugControl();
common.hbRunProto();

/*
 * Application Explorer: https://external-
157476462544637.proxy.sn.ws.broadcom.com/webtools/index.html
 *
 * Below is close to the code you should have had in the last step. I've taken
out some
 * unnecessary statements like "hb.set('hb_cursor_pos', '1337');" because TRAD is
not
 * sensitive to cursor position. I've also wrapped the statements in a
function definition.
 * This allows us to get the "Real-Time Quote" for any company by simply passing
 * a different company number.

```

```

*
* However when you execute the code below (try it: right click HB.js Commands >
Run) you receive
* the same company with the same values.
*
* Do you know why?
*
* If you look at the flow of TRAD, you have to navigate back to the "Select
Company"
* screen before selecting the next company. On the Application Explorer you see
* that means pressing the F3 or PF3 key twice. In the getQuote function below
* we are not backing up to the Select Company screen. HB.js does not change the
flow
* of the transaction.
*
* The command to press the PF3 is hb.run('HB_AID=PF3'); That command needs
* to be entered twice to back up to the select company screen. Add it to
getQuote
* function now and retest.
*
* If you have done the above correctly you will receive Total Value of Shares
Held for two
* companies. But notice that the hb.run you added does not look like other
hb.runs, they all
* have two parameters whereas yours only has one:
* hb.run('HB_AID=ENTER', 'T002'); vs hb.run('HB_AID=PF3');
*
* The T002 above is the BMS map name for that screen. The hb.run is checking to
make sure you
* ended up on the screen you expected to be on after the command was
executed. The statement
* hb.run('HB_AID=ENTER', 'T002') says that after pressing the enter I expect to
be on screen T002
* and if not then throw an error.
*
* How do you know what the BMS map names are? They are displayed in the lower
left corner
* of the application explorer. Update the two statements you added with a
second parameter
* that is the BMS map name the application should be on after pressing the PF3
key.
* hb.run('HB_AID=PF3', 'BOB'); You can get the BMS map name from the
Application Explorer. If
* you get it wrong the error message will tell you the BMS map name that was
encountered and
* you can correct your code and retest.
*/

```

```

function initialize() {
    hb.set('hb_entry', 'trad');
    hb.run('HB_AID=ENTER', 'T002');
}

function getQuote(companyNumber) {
    hb.set('OPTION', companyNumber);
    hb.run('HB_AID=ENTER', 'T003');

    hb.set('OPT2', '1');
    hb.run('HB_AID=ENTER', 'T004');

    writeln('Company Name: ', hb.getFieldValue('COMP41'));
    writeln('Total Value of Shares Held = ', hb.getFieldValue('VALUE'));
    writeln();

    hb.run('HB_AID=PF3');
    hb.run('HB_AID=PF3');

}

initialize();
getQuote('1');
getQuote('2');

```

Congratulations, you have successfully completed the scheduled workshops. We have included an additional three workshops for you to try on your own if interested.