

TraceX

Content

1. Introduction
2. Working Mechanism, API's and Methods Used
3. Downloads
4. Folder Structure
5. Presentation
6. Installation
7. Execution Instructions
8. References

Introduction

TraceX is a devtools extension which allows you to get the "trace-ids" and "link to the trace view" on Octo. Along with the start time and duration of the microservice, it also provides you the microservice's requested URL.

URL

- 1 <https://hybrid-ga-fphchyb30.beta.spectra.us-phoenix-1.ocs.oc-test.com/api/erp/expenses/resource>
- 2 <https://hybrid-ga-fphchyb30.beta.spectra.us-phoenix-1.ocs.oc-test.com/api/erp/expenses/resource>
- 3 [https://hybrid-ga-fphchyb30.beta.spectra.us-phoenix-1.ocs.oc-test.com/api/erp/expenses/resourcefinder=IncompleteExpensesFinder&onlyData=true&fields=CardBrandLookupCode%2CCountryCode'](https://hybrid-ga-fphchyb30.beta.spectra.us-phoenix-1.ocs.oc-test.com/api/erp/expenses/resourcefinder=IncompleteExpensesFinder&onlyData=true&fields=CardBrandLookupCode%2CCountryCode)
- 4 <https://hybrid-ga-fphchyb30.beta.spectra.us-phoenix-1.ocs.oc-test.com/api/erp/expenses/resourcefinder=ExpenseReadyForSubmissionFinder&onlyData=true&fields=CardBrandLookupCode%2CCour>
- 5 <https://hybrid-ga-fphchyb30.beta.spectra.us-phoenix-1.ocs.oc-test.com/api/erp/expenses/resource>

Start Time	Duration	X-B3-TraceID	Octo Link
11/07/2022, 21:54:32	1346.167ms	20b6b4cda93c412c842d8695c196a192	Link
11/07/2022, 21:54:32	1346.167ms	20b6b4cda93c412c842d8695c196a192	Link
11/07/2022, 21:54:32	1346.167ms	20b6b4cda93c412c842d8695c196a192	Link
11/07/2022, 21:54:32	1346.167ms	20b6b4cda93c412c842d8695c196a192	Link
11/07/2022, 21:54:32	1346.167ms	20b6b4cda93c412c842d8695c196a192	Link

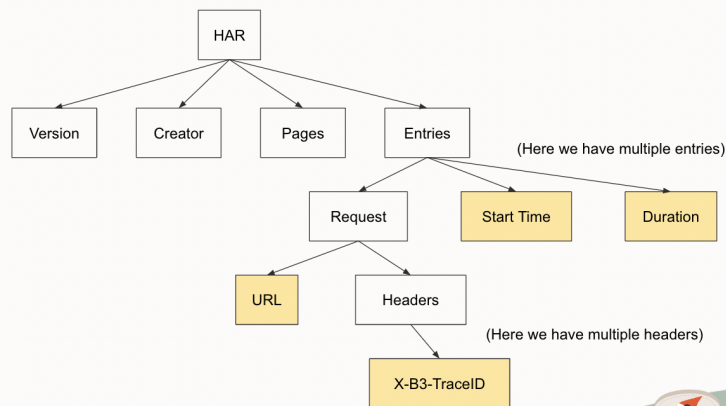
Working Mechanism, API's and Methods Used:

Overview: The [HTTP Archive format \(HAR\)](#) is used to represent information about network requests. It tracks all the logging of web browser's interaction with a site. So we have used [chrome.devtools.network.getHAR\(\)](#) method which returns the entire HAR log. The x-b3-traceid, start time, requested url, and time duration will then be obtained from the HAR log. Octo link to trace view can be generated using x-b3-traceid. If the Developer Tools window is opened after the page is loaded, some requests may be missing in the array of entries returned by getHAR(). Reload the page to get all requests. In general, the list of requests returned by getHAR() should match that displayed in the Network panel.

```
chrome.devtools.network.getHAR(
  callback: function,
)
```

The above mentioned can be extracted by following below flowchart.

Working Mechanism



7

Working Mechanism



Generating [Octo Link to trace view](#) - 'pre' + traceid + 'post'

- Trace id - **68ad4f711b3c44b2b5182732cc64169b**
<https://preprod-fusion.octo.oraclecloud.com/tv/explore?orgId=3&left=%7B%22datasource%22:%22Jaeger-us-phoenix-1%22,%22queries%22:%5B%7B%22refId%22:%22A%22,%22instant%22:true,%22range%22:true,%22exemplar%22:true,%22query%22:%2268ad4f711b3c44b2b5182732cc64169b%22%7D%5D,%22range%22:%7B%22from%22:%22now-1h%22,%22to%22:%22now%22%7D%7D>
- Trace id - **ea6bc903cb8146ae99dcea8a637e9af7**
<https://preprod-fusion.octo.oraclecloud.com/tv/explore?orgId=3&left=%7B%22datasource%22:%22Jaeger-us-phoenix-1%22,%22queries%22:%5B%7B%22refId%22:%22A%22,%22instant%22:true,%22range%22:true,%22exemplar%22:true,%22query%22:%22ea6bc903cb8146ae99dcea8a637e9af7%22%7D%5D,%22range%22:%7B%22from%22:%22now-1h%22,%22to%22:%22now%22%7D%7D>

8

[chrome.runtime.onConnect\(\)](#): To keep the track of whether DevTools window is active or not I have added an onConnect listener to the background page and use the connect() method from the DevTools page.

```
chrome.runtime.onConnect.addListener(
  callback: function,
)
```

Message Passing: Communication between extensions and their content scripts works by using message passing. So, ports are designed as a two-way communication method between different parts of the extension, where a (top-level) frame is viewed as the smallest part. Upon calling `runtime.connect` a Port is created and this port can immediately be used for sending messages to the other end via `postMessage`. Refer [link](#) for an example of Message Passing.

chrome.devtools.panels: This API is used to integrate the Octo extension into the Developers Tool Window.

```
chrome.devtools.panels.create(  
  title: string,  
  iconPath: string,  
  pagePath: string,  
  callback?: function,  
)
```

Refer [link](#) for more example of DevTools extension.

Downloads

Download and unpack the [zip](#) file. Or clone the repository [TraceX](#)(In master branch).

1. Clone with HTTPS - [Link](#)
2. Clone with SSH - [Link](#)

Folder Structure

The downloaded zip file have below files -

1. background.js
2. devtools.html
3. devtools.js
4. icon.png
5. manifest.json
6. octo.html
7. popup.html
8. popup.js

Presentation

Please find the attached [ppt](#).

Installation

Chrome:

1. Open Google Chrome and go to [chrome://extensions/](#) page. (Or click on Settings->Extensions)
2. Enable developer mode(if it's not already).
3. Click on the Load unpacked extension.
4. Navigate to the folder where you have cloned this repository. Select the folder for this extension.
5. You can find TraceX extension in extensions tab.

Please find the detailed video of installation. [Installation Guide.mp4](#)

Execution Instructions

You can test the TraceX on [expenses](#).

1. Right click and select Inspect Element to open DevTools.
2. Open Network Tab in DevTools.
3. Reload Network tab.
4. Click on the TraceX extension and select the Trace Log option and hit the Get button.
5. Now you can find Octo tab next to Network Tab in DevTools.
6. Click on the Octo tab to get all the required information.

Please find the detailed video of execution. [Execution Guide.mp4](#)

References

1. [Inspect Network Activity](#)
2. [Message Passing](#)
3. [Extending DevTools](#)
4. [network.getHAR\(\)](#)
5. [chrome.devtools.network](#)
6. [Web APIs](#)
7. [HAR](#)
8. [manifest.json](#)
9. [getHAR\(\)](#)
10. [har-export-trigger](#)