

FLOW SOFTWARE

Diagnostics and Troubleshooting Basics

IT'S ABOUT TIME

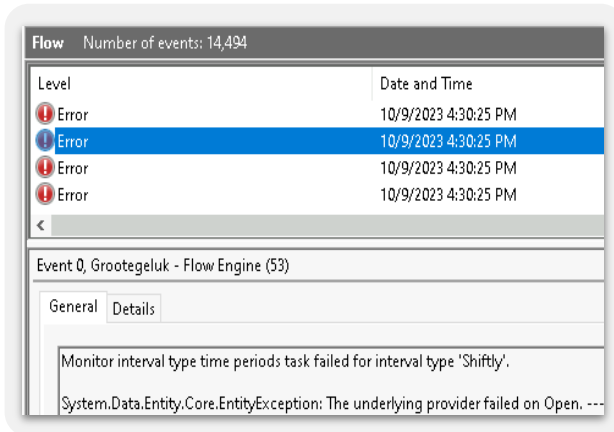


Available Diagnostic Resources

Loggers

Flow, Windows, and Linux specific journals, keeping track of Flow system changes and errors.

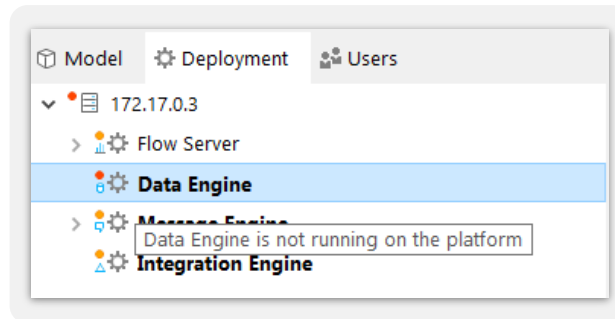
- Windows Event Viewer
- Bootstrap.exe
- Linux Journal



Flow Config Tool

Built-in features and menus found in the Flow Configuration tool.

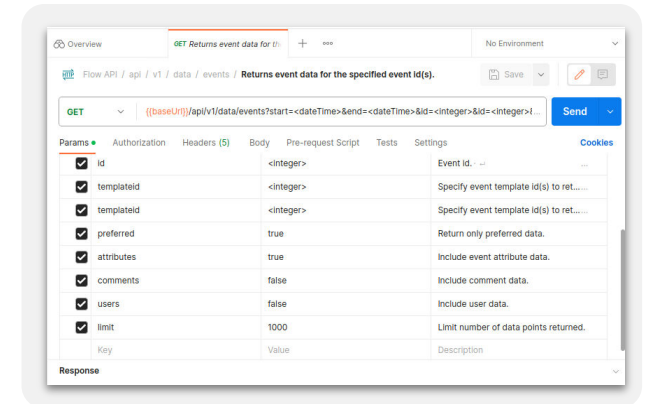
- Activity View
- Monitor Tool
- Change Logs
- Server Statistics



Third-party Tools

Tools and service publicly available that can be used to aid in the troubleshooting process. They include but are not limited to:

- SQL Server
- Postman



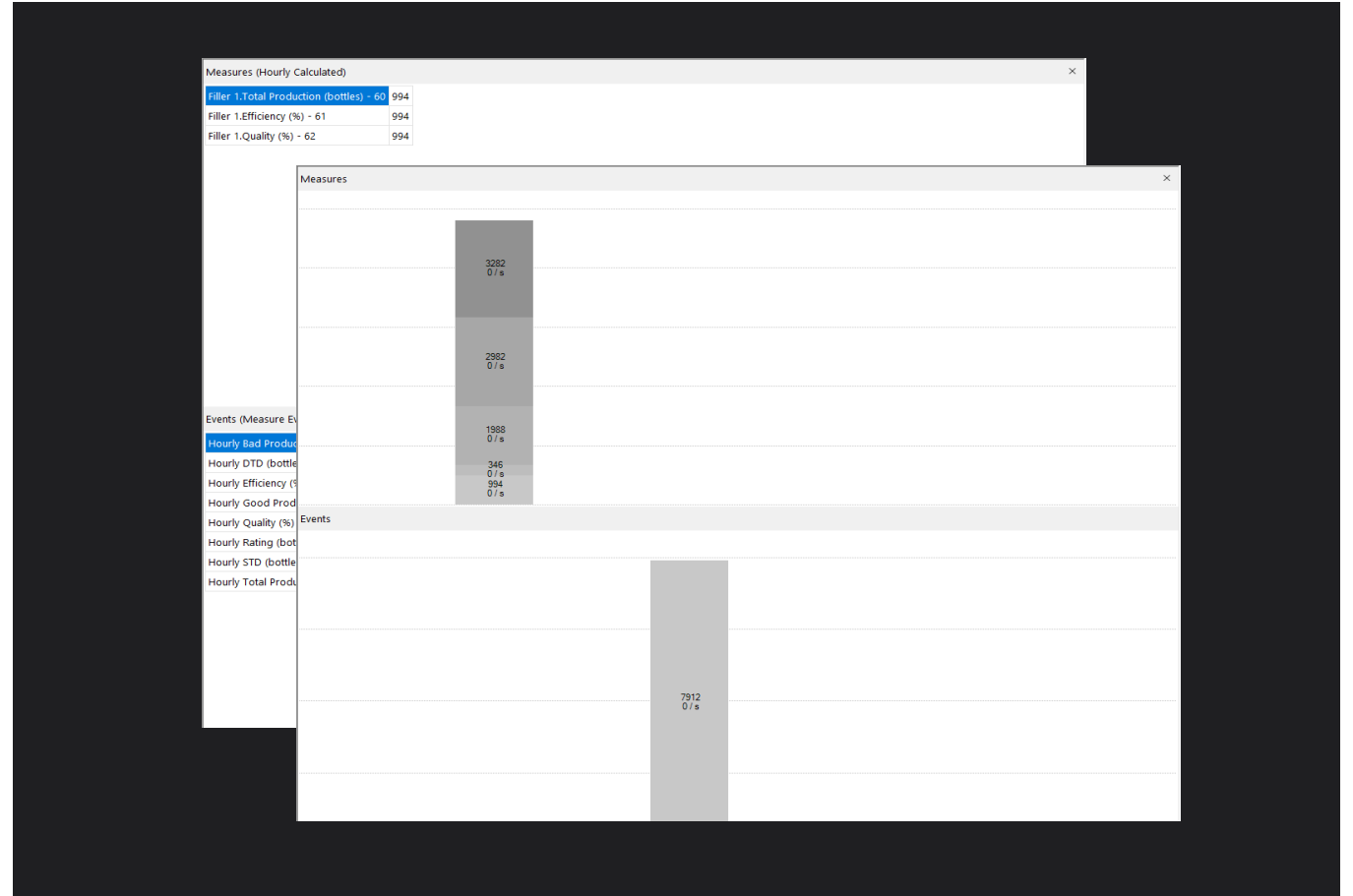


DIAGNOSTIC TOOLSET

- Flow Config - Monitor

A visual representation of the Flow Data engine processing queue, is grouped by interval and stacked by retrieval type.

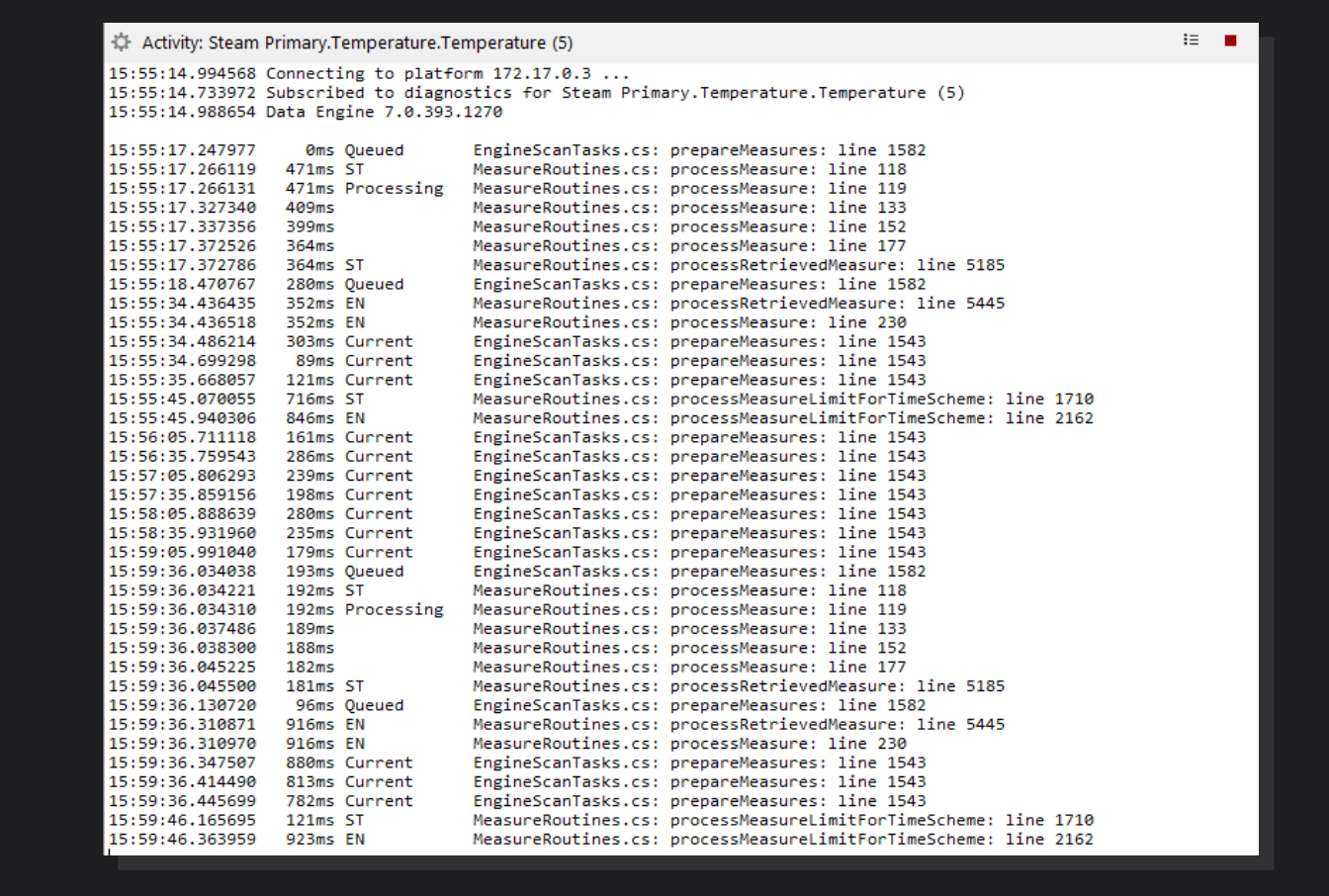
- Heaps shrink as engines complete tasks
- Common Use:
 - Faulted & Quarantined measures
 - Measure/event processing rate
 - Queue size
- Drill-down for measure-level stats
- Found in the View menu



- Flow Config – Activity Monitor

The Activity Monitor provides a view of current engine tasks on and can be found on Engines, Measures & Events.

- Includes:
 - Status
 - Processing Time
- Common Use:
 - Measure/Event processing state in engine
- Found in the item's Context menu



```
Activity: Steam Primary.Temperature.Temperature (5)
15:55:14.994568 Connecting to platform 172.17.0.3 ...
15:55:14.733972 Subscribed to diagnostics for Steam Primary.Temperature.Temperature (5)
15:55:14.988654 Data Engine 7.0.393.1270

15:55:17.247977      0ms Queued      EngineScanTasks.cs: prepareMeasures: line 1582
15:55:17.266119    471ms ST          MeasureRoutines.cs: processMeasure: line 118
15:55:17.266131    471ms Processing  MeasureRoutines.cs: processMeasure: line 119
15:55:17.327340    409ms            MeasureRoutines.cs: processMeasure: line 133
15:55:17.337356    399ms            MeasureRoutines.cs: processMeasure: line 152
15:55:17.372526    364ms            MeasureRoutines.cs: processMeasure: line 177
15:55:17.372786    364ms ST          MeasureRoutines.cs: processRetrievedMeasure: line 5185
15:55:18.470767    280ms Queued      EngineScanTasks.cs: prepareMeasures: line 1582
15:55:34.436435    352ms EN          MeasureRoutines.cs: processRetrievedMeasure: line 5445
15:55:34.436518    352ms EN          MeasureRoutines.cs: processMeasure: line 230
15:55:34.486214    303ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:55:34.699298     89ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:55:35.668057    121ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:55:45.070055    716ms ST          MeasureRoutines.cs: processMeasureLimitForTimeScheme: line 1710
15:55:45.940306    846ms EN          MeasureRoutines.cs: processMeasureLimitForTimeScheme: line 2162
15:56:05.711118    161ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:56:35.759543    286ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:57:05.806293    239ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:57:35.859156    198ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:58:05.888639    280ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:58:35.931960    235ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:59:05.991040    179ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:59:36.034038    193ms Queued      EngineScanTasks.cs: prepareMeasures: line 1582
15:59:36.034221    192ms ST          MeasureRoutines.cs: processMeasure: line 118
15:59:36.034310    192ms Processing  MeasureRoutines.cs: processMeasure: line 119
15:59:36.037486    189ms            MeasureRoutines.cs: processMeasure: line 133
15:59:36.038300    188ms            MeasureRoutines.cs: processMeasure: line 152
15:59:36.045225    182ms            MeasureRoutines.cs: processMeasure: line 177
15:59:36.045500    181ms ST          MeasureRoutines.cs: processRetrievedMeasure: line 5185
15:59:36.130720     96ms Queued      EngineScanTasks.cs: prepareMeasures: line 1582
15:59:36.310871    916ms EN          MeasureRoutines.cs: processRetrievedMeasure: line 5445
15:59:36.310970    916ms EN          MeasureRoutines.cs: processMeasure: line 230
15:59:36.347507    880ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:59:36.414490    813ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:59:36.445699    782ms Current     EngineScanTasks.cs: prepareMeasures: line 1543
15:59:46.165695    121ms ST          MeasureRoutines.cs: processMeasureLimitForTimeScheme: line 1710
15:59:46.363959    923ms EN          MeasureRoutines.cs: processMeasureLimitForTimeScheme: line 2162
```

- Loggers – Windows Event Viewer

Windows event logs that provide a detailed record of system and application activity, including bootstrap, platform, component, application, and security events.

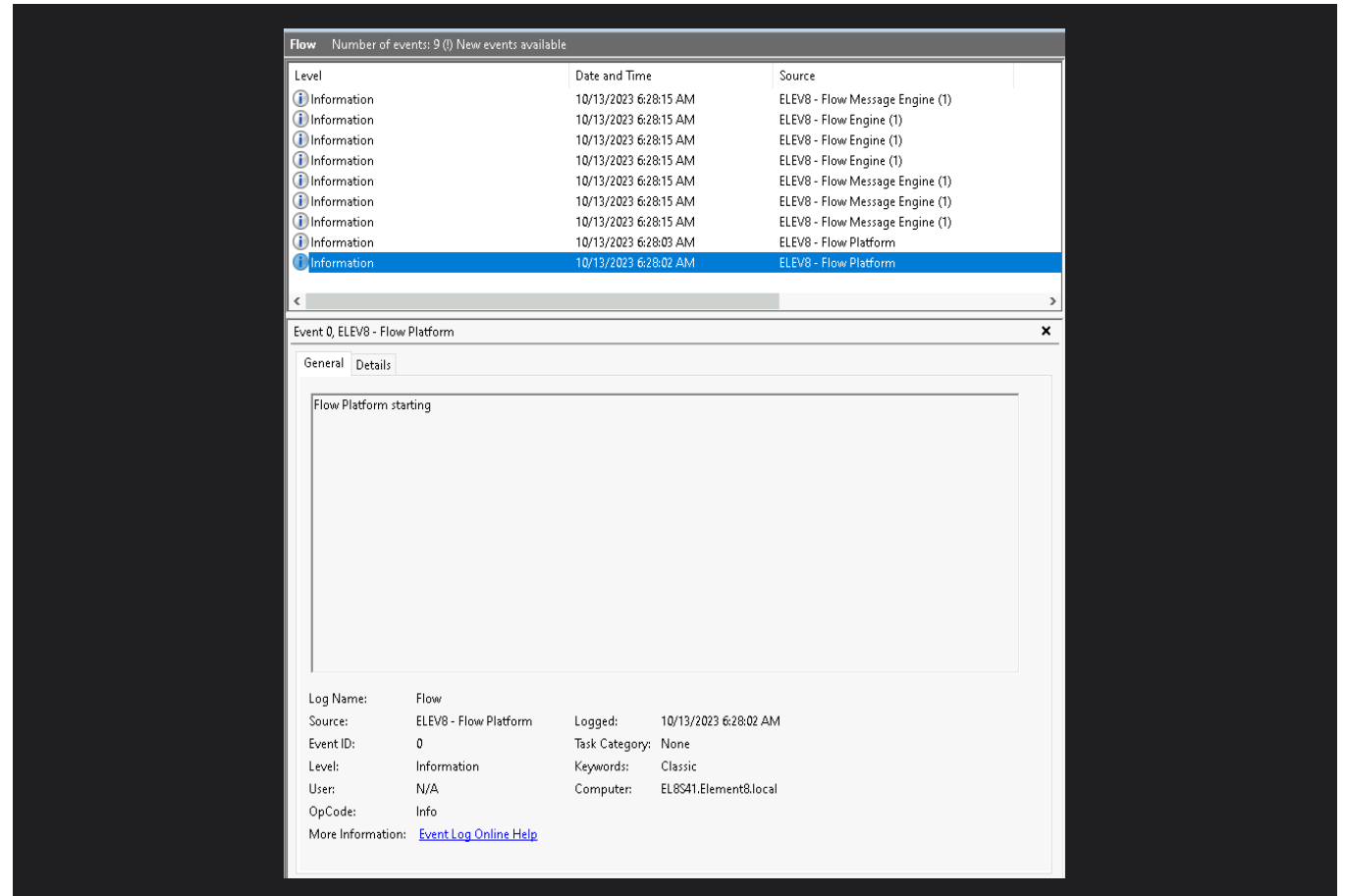
- Includes:

- Categorization by component
- Event, Measure, other IDs included in message

- Common Use:

- Config errors & faults
- Data Source/Consumers errors

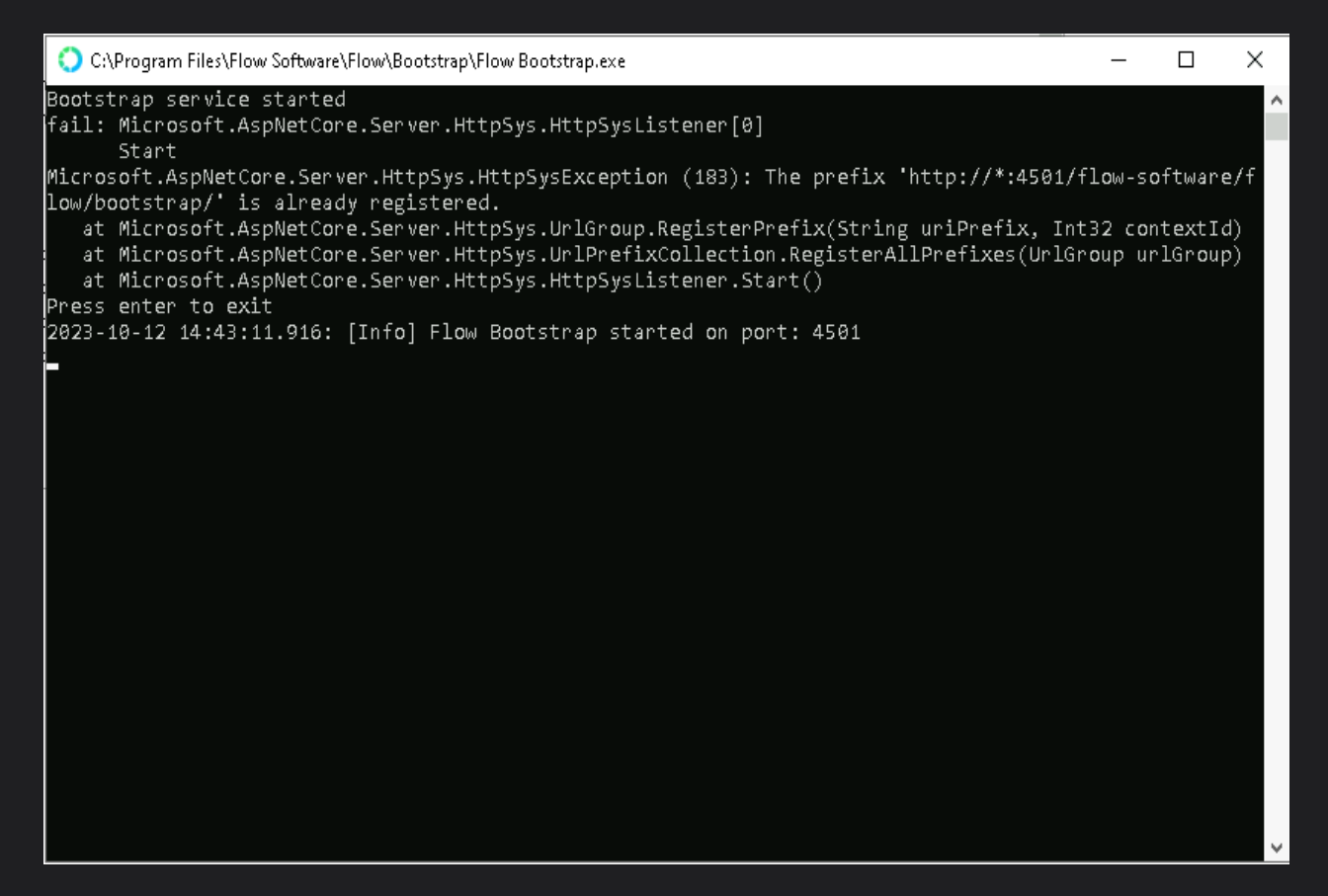
- Found in Windows Admin Tools



- Loggers – Bootstrap.exe

The Flow bootstrap service is the foundation for all other Flow components, is first to start and can be executed directly in the command prompt to identify boot level errors.

- Common Use:
 - Lists errors before other services start
 - Used on critical failure
- Found in C:\Program Files\Flow Software\Flow\Bootstrap\Bootstrap.exe



```
C:\Program Files\Flow Software\Flow\Bootstrap\Flow Bootstrap.exe
Bootstrap service started
fail: Microsoft.AspNetCore.Server.HttpSys.HttpSysListener[0]
      Start
Microsoft.AspNetCore.Server.HttpSys.HttpSysException (183): The prefix 'http://*:4501/flow-software/flow/bootstrap/' is already registered.
   at Microsoft.AspNetCore.Server.HttpSys.UrlGroup.RegisterPrefix(String uriPrefix, Int32 contextId)
   at Microsoft.AspNetCore.Server.HttpSys.UrlPrefixCollection.RegisterAllPrefixes(UrlGroup urlGroup)
   at Microsoft.AspNetCore.Server.HttpSys.HttpSysListener.Start()
Press enter to exit
2023-10-12 14:43:11.916: [Info] Flow Bootstrap started on port: 4501
```



DEMO

- Loggers – Docker CLI/Linux Journal

The Journal is the Linux equivalent of the Windows event logs and provides a detailed record of bootstrap events.

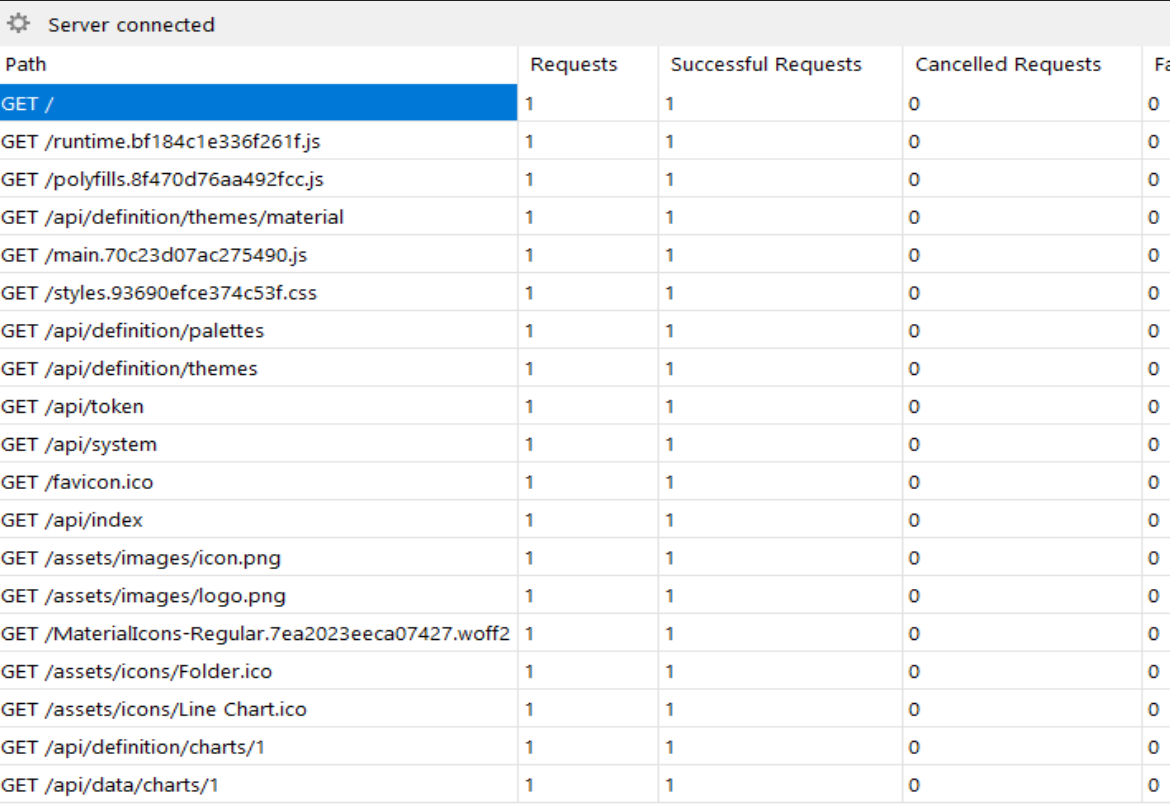
- Analogous to Bootstrap.exe
 - Flow for Linux and Docker image
- Viewed from Bash or preferred container manager

```
Info: Flow.Worker[0]
      at System.Linq.Enumerable.TryGetSingle[TSource](IEnumerable`1 source, Boolean& found)
Info: Flow.Worker[0]
      at lambda_method551(Closure , QueryContext )
Info: Flow.Worker[0]
      at Flow.Data.FlowContext.GetSystemProperty(String Name)
Info: Flow.Worker[0]
      at Flow.Bootstrap.MonitorPlatforms() in C:\Source\Flow\Flow-Bootstrap\Bootstrap\Main\B
Info: Flow.Worker[0]
      ClientConnectionId:00000000-0000-0000-0000-000000000000
Info: Flow.Worker[0]
      2023-10-13 04:40:12.009: [Info] Starting the 'Papercut Email' notification server
Info: Flow.Worker[0]
      2023-10-13 04:40:12.031: [Info] Flow Notification Server Host starting
Info: Flow.Worker[0]
      2023-10-13 04:40:12.055: [Info] Flow Notification Server Host stopped
Info: Flow.Worker[0]
Info: Flow.Worker[0]
Info: Flow.Worker[0]
      2023-10-13 04:40:22.053: [Info] Starting the 'Papercut Email' notification server
Info: Flow.Worker[0]
      2023-10-13 04:40:22.077: [Info] Flow Notification Server Host starting
Info: Flow.Worker[0]
      2023-10-13 04:40:22.102: [Info] Flow Notification Server Host stopped
Info: Flow.Worker[0]
Info: Flow.Worker[0]
```

- Flow Config – Flow Server Statistics

The server statistics provide summary on endpoint, charts, dashboard usage and performance.

- Common Use:
 - Popular or problematic charts
 - Average chart performance
- Found in Deployment view > Flow Server context menu



The screenshot displays a table titled "Server connected" with the following columns: Path, Requests, Successful Requests, Cancelled Requests, and Failure Rate. The table lists 20 different endpoints, all of which have 1 request, 1 successful request, 0 cancelled requests, and a 0% failure rate.

Path	Requests	Successful Requests	Cancelled Requests	Failure Rate
GET /	1	1	0	0
GET /runtime.bf184c1e336f261f.js	1	1	0	0
GET /polyfills.8f470d76aa492fcc.js	1	1	0	0
GET /api/definition/themes/material	1	1	0	0
GET /main.70c23d07ac275490.js	1	1	0	0
GET /styles.93690efce374c53f.css	1	1	0	0
GET /api/definition/palettes	1	1	0	0
GET /api/definition/themes	1	1	0	0
GET /api/token	1	1	0	0
GET /api/system	1	1	0	0
GET /favicon.ico	1	1	0	0
GET /api/index	1	1	0	0
GET /assets/images/icon.png	1	1	0	0
GET /assets/images/logo.png	1	1	0	0
GET /MaterialIcons-Regular.7ea2023eeca07427.woff2	1	1	0	0
GET /assets/icons/Folder.ico	1	1	0	0
GET /assets/icons/Line Chart.ico	1	1	0	0
GET /api/definition/charts/1	1	1	0	0
GET /api/data/charts/1	1	1	0	0

- Loggers – Docker CLI/Bash

Analogous to the Windows event logs, it provide a detailed record of of the the Bootstrap services events on Linux systems.

- Analogous to Bootstrap.exe
 - Flow for Linux and Docker image
- Viewed from Bash or preferred container manager

```
Info: Flow.Worker[0]
      at System.Linq.Enumerable.TryGetSingle[TSource](IEnumerable`1 source, Boolean& found)
Info: Flow.Worker[0]
      at lambda_method551(Closure , QueryContext )
Info: Flow.Worker[0]
      at Flow.Data.FlowContext.GetSystemProperty(String Name)
Info: Flow.Worker[0]
      at Flow.Bootstrap.MonitorPlatforms() in C:\Source\Flow\Flow-Bootstrap\Bootstrap\Main\B
Info: Flow.Worker[0]
ClientConnectionId:00000000-0000-0000-0000-000000000000
Info: Flow.Worker[0]
2023-10-13 04:40:12.009: [Info] Starting the 'Papercut Email' notification server
Info: Flow.Worker[0]
2023-10-13 04:40:12.031: [Info] Flow Notification Server Host starting
Info: Flow.Worker[0]
2023-10-13 04:40:12.055: [Info] Flow Notification Server Host stopped
Info: Flow.Worker[0]
Info: Flow.Worker[0]
Info: Flow.Worker[0]
2023-10-13 04:40:22.053: [Info] Starting the 'Papercut Email' notification server
Info: Flow.Worker[0]
2023-10-13 04:40:22.077: [Info] Flow Notification Server Host starting
Info: Flow.Worker[0]
2023-10-13 04:40:22.102: [Info] Flow Notification Server Host stopped
Info: Flow.Worker[0]
Info: Flow.Worker[0]
```

- Third-party tool – SQL Server

The SQL server and related SQL management tools allow for running diagnostic queries against Flow DB.

- Common Use:

- Confirm SQL compatibility level
- SQL DS and DC connection testing
- Performance & Maintenance Queries
- Flow Support issued queries

- Accessed via SQL management client

The screenshot displays the 'Database Properties - ELEV8' window in SQL Server Enterprise Manager. The 'Automatic' section is expanded, showing various server options. A query window is open with a query that selects top 10 records from sys.dm_os_sys_info, including record_id, EventTime, SQLProcess (%), SystemIdle, and OtherProcess (%). The results pane shows a table with 5 rows of data.

record_id	EventTime	SQLProcess (%)	SystemIdle	OtherProcess (%)
1	207	2023-10-13 06:20:51.143	0	100
2	206	2023-10-13 06:19:51.143	0	100
3	205	2023-10-13 06:18:51.140	0	100
4	204	2023-10-13 06:17:51.137	0	100
5	203	2023-10-13 06:16:51.133	0	100

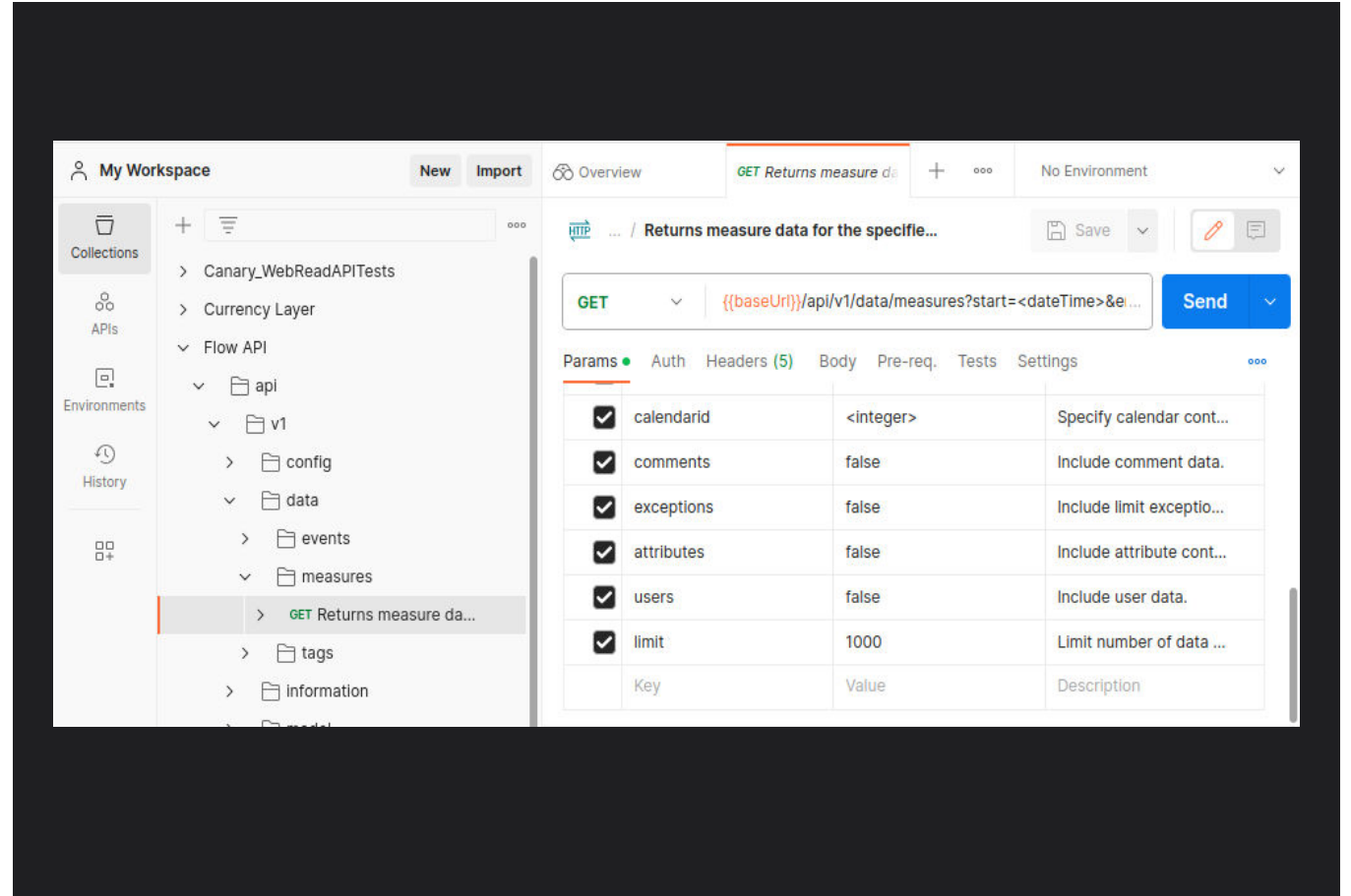


Unsanctioned changes to the Flow DB violate EULA

- Third-party tool - Postman

Postman is a powerful and versatile platform for testing APIs.

- Common Use:
 - Flow RestAPI connection testing
 - Flow RestAPI response result auditing
- Found on Postman.com



Tebello Masedi
Element8
support@element8.co.za