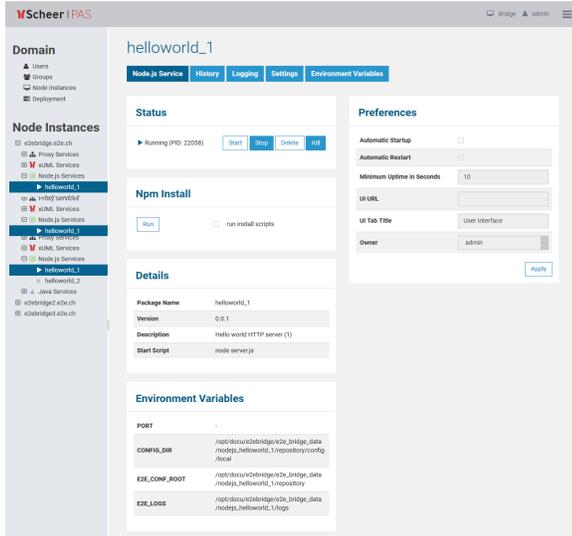


Node.js Service Details

You can only inspect the details of Node.js services of the node instance the used Bridge is running on. If you have aggregated multiple node instances into a Bridge domain, you need to use the Bridge of the specific node instance the Node.js service is running on.

All users may view details of a Node.js service. Expand the tree below a node instance in the **Node Instances** section of the navigation on the left. Then, navigate to a Node.js service entry below the sub-navigation item **Node.js**. The tab **Node.js Service** is initially displayed.

Figure: Node.js Service Details



On this Page:

- [Status Information](#)
- [Npm Install](#)
- [Service Details](#)
- [Preferences](#)
- [Environment Variables](#)
- [History](#)

Related Pages:

- [Logging of Node.js Services](#)
- [Setting Environment Variables for Node.js Services](#)
- [Developing Node.js Services](#)

Status Information

In the **Status** section, the status of the node.js service (**Running** or **Stopped**) is displayed. Here you can start, stop, delete, or kill the service.

- **Stopping** the service
Bridge 7.2.0 Clicking **Stop** will send `SIGTERM` to the service. Older Bridges called `Java Process.destroy()`.
- **Killing** the service
Bridge 7.2.0 The **Kill** functionality will first try to regularly stop the service. When the service is still running after 10 seconds, it will forcibly terminate the service with `Java Process.destroyForcibly()`.

If the service is up and running, you can see the system process id (**PID**) of the service. To match the PID with the Bridge service, you can also use system commands:

System	Command / Output
Linux	<pre>ps -efa grep -- --instance</pre> <pre>... 15228 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node index.js --instance /opt/e2e_bridge_data/nodejs_NodeService ... 15265 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node app.js --instance /opt/e2e_bridge_data/nodejs_api-test- helloworld_43 ... 15853 ... /opt/e2e_bridge_prog/j2re-11.0.2/linux-64/bin /java -jar repository.jar --instance /opt/e2e_bridge_data /java_helloworld</pre> <pre>[...]</pre>

Windows	<p>Run with administration privileges:</p> <pre>wmic process where "caption='bridgeserver.exe' or caption='node.exe' or caption='java.exe' " get processid,caption,commandLine /format:csv findstr /C:--instance</pre> <pre>... ,bridgeserver.exe, "C:\E2E_BRIDGE_PROG\bridgeserver-2018.12\win32-64\bridgeserver.exe" --config "C:\var\E2E_BRIDGE_DATA\server.cfg" --instance "C:\var\E2E_BRIDGE_DATA\bridge_SoapWait",17600 ... ,node.exe,C:\E2E_BRIDGE_PROG\nodejs-8.11.4\win32-64\node app.js "--instance C:\var\E2E_BRIDGE_DATA\nodejs_api-test-helloworld",20144</pre>
----------------	---

Npm Install

A Node.js service can have dependencies to other packages. Click **Run** to execute npm install to refresh dependent packages.

As per default, this 'npm install' does not run installer scripts. Check **run scripts**, if you want to execute npm installer scripts, but be aware of the fact that running scripts is a security vulnerability.

To inspect the logs of the 'npm install' command, switch to the deployed service and go to tab **Logging** (see also [Logging of Node.js Services](#)).

To execute npm install, the service has to be stopped. Otherwise, **Run** will be disabled.

Service Details

In the **Details** section, the following information is displayed:

Element	Description	Origin (package.json)
Version	Version of the Node.js service as specified in package.json.	version
Description	Description of the Node.js service as specified in package.json.	description
Start Script	Start Script of the Node.js service as specified in package.json.	scripts/start

Preferences

In the **Preferences** section, the following preferences can be modified:

Option	Description
Automatic Startup	<p>Select this option, if you want the service to startup automatically, whenever the Bridge is started. Only users who are member of a group, to which the role ADMIN has been assigned, are allowed to change this option.</p> <p>This option can be globally disabled by the Disable Automatic Service Startup option on the node instance preferences.</p> <p>When updating the BRIDGE, all deployed services (xUML, Node.js, and Java services) will be kept. However, the automatic startup option will be ignored on the very first start-up after the update.</p>

Automatic Restart	Whenever the service crashes, it will restart immediately. Nevertheless, in the navigation the icon  next to the xUML service name will indicate the abnormal termination. Also consider the implications of Minimum Uptime in Seconds when setting this option.
Minimum Uptime in Seconds	To allow the Bridge to distinct whether the service has crashed during start-up or not, specify the minimum uptime of the Node.js service in seconds. Implications: <ul style="list-style-type: none"> • If the service crashes during the uptime period, the Bridge will assume that the service could not be started. No Automatic Restart will be applied. • If the service crashes after the uptime period, the Bridge will assume a crash. If option Automatic Restart is set, the Bridge will try to restart the service.
UI URL	Specify the URL of the user interface of the Node.js service. It then will be displayed as an additional tab within the Bridge and you can access it directly via the Bridge.
UI Tab Title	If the Node.js service has a user interface that is integrated to the Bridge, you can specify the title of the tab here. The default title is User Interface .
Owner	The group id of the user who has deployed the service. Only users who are member of a group, to which the role ADMIN has been assigned, are allowed to modify the owner of the service.

Environment Variables

Each Node.js service runs with a specific set of environment variables which is provided by the operating system. They can be added and changed on tab **Environment Variables** (see [Setting Environment Variables for Node.js Services](#)). Additionally, the Bridge provides environment variables that are strongly recommended to be used by developers in their code (see [Developing Node.js Services](#)):

Variable	Description	Default
PORT	Node.js service port	-
CONFIG_DIR	Path to the configuration directory of the service (within the working directory of the service). <ul style="list-style-type: none"> • All configuration files (e.g. service settings) that are stored in this location will survive a Node.js service update. Files that have been saved to the working directory only, will be deleted on update. • This directory will be created by the Bridge on service startup, if not existent. 	reposit ory /config /local
E2E_CONF_ROOT	If you use the e2e-conf Node.js module , you can initialize e2e-conf with E2E_CONF_ROOT . This is especially useful if your e2e-conf is not a top level dependency but a sub-module dependency.	reposit ory
E2E_LOGS	If the Node.js service writes log files in this directory, you can view them in the Bridge. <ul style="list-style-type: none"> • If the log file keeps growing, you should rotate the log file per day and use this naming pattern: <prefix>_YYYY-MM-DD.log. • If the log file has a fixed size, you can use the naming pattern <prefix>.log. <p>Do not use the following prefixes: start, stdout, stderr and transaction. They are reserved prefixes by the Bridge.</p>	logs

These special environment variables are listed on the service details tab.

History

Switch to the **History** tab to view the history of all user actions that were executed on the selected Node.js service since its deployment.

In the Node.js service history all user actions are listed - comprising starting and stopping the service, modification of the preferences, etc. The list is sorted in a chronological order and also shows the user who initiated the action.

The history also lists internal actions, for instance, when the system started or stopped a Node.js service automatically.

All users have access to the history information.

Figure: Node.js Service History

The screenshot shows a web interface for 'helloworld_1' with a navigation bar containing 'Node.js Service', 'History', 'Logging', 'Settings', and 'Environment Variables'. The 'History' tab is active. Below the navigation bar, there is a 'History' section with a dropdown menu set to 'Max. entries to load: 20' and a 'Show 10 entries' button. A search filter is also present. The main content is a table with three columns: 'Date', 'User', and 'Action'. The table lists 10 entries, showing a sequence of 'Start Instance' and 'Stop Instance' actions performed by the 'admin' user. The entries are sorted chronologically. At the bottom of the table, there is a pagination control showing 'Showing 1 to 10 of 20 entries' and 'Previous 1 2 Next'.

Date	User	Action
2019/02/09 13:45:23 CET	admin	Start Instance
2019/02/05 10:59:50 CET	admin	Run 'npm install'
2019/02/05 10:56:33 CET	admin	Stop Instance
2019/02/05 10:56:02 CET	admin	Start Instance
2019/02/05 10:56:50 CET	admin	Stop Instance
2019/02/05 10:56:48 CET	admin	Start Instance
2019/02/05 10:56:40 CET	admin	Stop Instance
2019/02/05 10:56:34 CET	admin	Start Instance
2018/07/23 11:25:41 CEST	admin	Stop Instance
2018/07/23 11:13:23 CEST	admin	Start Instance