
What's New in EF Portal 2025.0



info@ni-sp-software.com / www.ni-sp-software.com

Copyright © 2023-2025 NI SP Software GmbH and/or its affiliates. All rights reserved. NI SP Software's trademarks and All other trademarks not owned by NI SP Software are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by NI SP Software.

Table of Contents

- Table of Contents.....2
- EF Portal REST API..... 3**
 - API Structure..... 4
 - Key Features..... 4
 - Grid Operations.....4
 - System Operations..... 4
 - Monitor Operations (Admin Only)..... 5
 - EF Portal Client - efpclient..... 5
- EF Portal Modern Look and Feel..... 10**
- Prevent File Downloads by Optional Allowlist..... 14**
- Limiting the Hosts Displayed..... 14**
- Jupyter Notebook Integration..... 15
- Other Updates..... 17**
 - Other Features and Improvements..... 17
 - Security / Upgrades..... 17

Please check out the EF Portal 2025.0 video, where we walk you through all the new features:
<https://www.youtube.com/watch?v=ILoMq2OkCbl>:



EF Portal REST API

EF Portal 2025.0 introduces a cutting-edge, feature-rich REST API designed to streamline and enhance your integration experience. This modern API replaces the legacy Web Services approach, which was previously supported by a Java-based efclient, with newer, more efficient technologies that align with industry standards.

Key Features of the EF Portal REST API:

- **JSON Output for Easy Consumption:** Simplified data exchange for seamless integration into scripts, command-line tools, and third-party applications.
- **Programmatic Job Submission:** Automate job submissions directly from scripts or the command line.
- **Enhanced Automation and Batch Operations:** Effortlessly manage repetitive tasks with robust automation capabilities.
- **Comprehensive Job Management:** Query job information, control job states (Cancel, Suspend, Resume), and monitor progress in real-time.
- **Cluster Node Insights:** Gather detailed information about cluster nodes for improved resource management.
- **File Management Capabilities:**
 - Upload files directly to EF Portal from scripts or the command line.
 - Monitor and download output files from HPC jobs efficiently.
- **User Monitoring:** Monitor logged-in users for better oversight and security and license usage analysis

Why Choose EF Portal REST API?

The EF Portal REST API is built on OpenAPI/Swagger standards, offering unparalleled ease of integration into workflows. It supports:

- Cross-scheduler and cross-platform compatibility for diverse environments.
- Real-time monitoring of jobs and resources.
- Seamless interaction with third-party tools based on industry-standard protocols.

With the REST API you can extend automation a step further and push the boundaries towards an end-to-end HPC automated workflow: from job submit to monitoring, to downloading results, to post-processing.

Developer-Friendly Features

With support for popular programming languages like Python, PHP, Node.js, Go, Rust, and more, developers can quickly generate EF Portal REST clients tailored to their needs. For simpler use cases, direct CURL access is also available.

EF Portal 2025.0 empowers users with simplified integration processes, enhanced automation capabilities, and real-time monitoring—all designed to optimize your HPC workflows and improve operational efficiency.

The new EF Portal client - `efpclient` - is implemented in Python and offers easy consumption of the REST API via an auto-completing client supporting all REST API features.

Below is an overview of EF Portal REST API endpoints and features.

API Structure

The EF Portal API is organized into three distinct categories for easy navigation and use:

1. HPC Schedulers: Manage hosts, jobs and queues.
2. Monitor: Access user information and manage licenses (admin-only operations).
3. System: Work with services, spoolers, and file operations.

Key Features

Grid Operations

- Host and queue management
 - List hosts
 - Get details of a host
 - Get details of a cluster in a scheduler
- Job control
 - List all jobs
 - List jobs belonging to a user
 - List jobs running on a specific host
 - Get details of a job
 - Delete a job
 - Resume a job
 - Suspend a job
 - Find a job related to a spooler

System Operations

- Create an REST API Access Token
- Service management
 - Submit a Service including file upload
 - List options of a service
- Spooler control and file management
 - List all Spoolers
 - Remove a Spooler

- List files in a Spooler
- Upload file into a spooler

Monitor Operations (*Admin Only*)

- User session management (List users logged in, logout a user)
- License control and usage monitoring

The REST API Documentation including the OpenAPI/Swagger `openapi.json` definition can be found at `docs/rest/` in the EF Portal installation, e.g. <https://efp:8443/enginframe/docs/rest/>.

EF Portal Client - `efpclient`

Here you can find the documentation of the new EFPCClient implemented in Python in PyPi and then download and install the `efpclient`: <https://pypi.org/project/efpclient>. The `efpclient` is also available as a downloadable package.

Here is an overview of typical EF Portal Client use cases:

- Create a New Authentication Token
- Get all Jobs on the default HPC Scheduler or VDI Session Manager
- Get Info for Job with id 142
- List all Services available for the user in the HPC Applications Portal
- Describe a Service available in the HPC Applications Portal
- Submit the `job.submission` Service Using local file `$HOME/file.log` and Compression Level 4
- List Files in Spooler after a Service Submission
- Download a File from the Spooler
- Upload a File into the Spooler
- List Spoolers

The EFPCClient supports the same configuration file like the `efclient` (EnginFrame WS Client) in the past for backward compatibility and a new configuration file: `$HOME/.efpclient.conf`.

Here we show the commands of a typical use case in the shell or a script:

```
# Install the EFP Client
> python3 -m pip install efpclient
# Start the client command
> efpclient
Commands:
  admin      Administration related commands group.
  clusters   Clusters related commands group.
  hosts      Hosts related commands group.
  jobs       Jobs related commands group.
```

queues Queues related commands group.
services Services related commands group.
spoolers Spoolers related commands group.
token Authentication token commands group.

```
# Get the REST API token
```

```
> efpclient token create --token-only
```

```
[https://demo.ni-sp.com:8448/enginframe] Enter the username to  
authenticate: username
```

```
[https://demo.ni-sp.com:8448/enginframe] Enter the password to  
authenticate:
```

```
20d00e78fd87cc6179e16e7b02fe1d1427ce340d
```

```
# Or in one line with the credentials in the config file only  
readable for the user:
```

```
# export EFP_CLIENT_TOKEN=$(efpclient token create --token-only)
```

```
> efpclient jobs all
```

```
[  
  {  
    "manager": "slurm",  
    "id": "85",  
    "name": "Job_Compress_Image_078.png",  
    "owner": "efadmin",  
    "queue": "test",  
    "total_cpu_usage": "0:01",  
    "memory_usage": "0",  
    "swap_usage": "0",  
    "execution_hosts": "demo",  
    "submission_time": "2025-04-01T11:13:53",  
    "execution_time": "2025-04-01T11:13:53",  
    "execution_directory":  
"/opt/nisp/enginframe/spoolers/efadmin/tmp3093980369702684817.ef",  
    "nice": "0",  
    "reasons": [  
      {  
        "value": "None"  
      }  
    ],  
    "status": {  
      "ef": "Done",  
      "grid": "COMPLETED",  
      "value": "DONE"  
    }  
  }  
]
```

```
> efpclient hosts
```

```
Commands:
```

```

all    List all available hosts.
info   Get host info for a given host name.
jobs   List jobs by host name, same as 'jobs host' command.
# Enable autocompletion
eval "$(_EFCLIENT_COMPLETE=bash_source efpclient)"
efpclient hosts <tab>
all    info jobs    # output from the auto completion

# Configure the HPC Applications SDF in $HOME/.efpclient.conf:
# sdf =
https://demo.ni-sp.com/enginframe/applications/applications.xml
# List the services available
> efpclient services list
[
  {
    "id": "batch_builtin_jupyter_notebook.published",
    "name": "Jupyter Notebook",
    "uri":
"//applications/batch_builtin_jupyter_notebook.published"
  },
  {
    "id": "batch_builtin_sample_compress_job.published",
    "name": "Sample Compress Job",
    "uri":
"//applications/batch_builtin_sample_compress_job.published"
  }
]
> efpclient services describe -s
//applications/batch_builtin_sample_compress_job.published
{
  "id": "batch_builtin_sample_compress_job.published",
  "uri":
"//applications/batch_builtin_sample_compress_job.published",
  "name": "Sample Compress Job",
  "options": [
    {
      "id": "file",
      "label": "File to compress:",
      "type": "sfu",
      "option": "--opt-file FILENAME",
      "help": "File to compress:. Specify a single local file
to upload."
    },
    {
      "id": "level",

```

```

    "label": "Compression level:",
    "type": "list",
    "value": "9",
    "choices": [
      {
        "value": "9",
        "label": "maximum"
      },
      {
        "value": "4",
        "label": "medium"
      },
      {
        "value": "1",
        "label": "minimum"
      }
    ],
    "option": "--opt-level VALUE_1|VALUE_2|...",
    "help": "Compression level:. Default value is: 9. Valid
values to choose from are: 9, 4, 1"
  },
  {
    "id": "cluster",
    "label": "Execution cluster:",
    "type": "list",
    "value": "cluster:slurm",
    "choices": [
      {
        "value": "cluster:slurm",
        "label": "cluster"
      }
    ],
    "option": "--opt-cluster VALUE_1|VALUE_2|...",
    "help": "Execution cluster:. Default value is:
cluster:slurm. Valid values to choose from are: cluster:slurm"
  }
],
"actions": [
  {
    "id": "submit",
    "result": "text/xml"
  }
]
}
# Submit the compress service with file results-112.out

```

```
> efpclient services submit -s
//applications/batch_builtin_sample_compress_job.published --opt-file
results-112.out --opt-level 4
{
  "uri":
  "spooler:///opt/nisp/enginframe/spoolers/efadmin/tmp9559103907370519188.ef",
  "output":
  "/enginframe/rest/system/services?_uri=//com.enginframe.system/show.spooler&_spooler=spooler%3A%2F%2F%2Fopt%2Fnisp%2Fenginframe%2Fspoolers%2Fefadmin%2Ftmp9559103907370519188.ef"
}
# Use the Spooler URI to show the files in the Spooler
> efpclient spoolers files -u
spooler:///opt/nisp/enginframe/spoolers/efadmin/tmp9559103907370519188
.ef
[
  {
    "path": "/",
    "vroot": "8b45963b96801790f315dff51718ca9cb23db834",
    "name": "results-112.out.gz",
    "type": "file",
    "modified": "2025-04-08T11:31:00+00:00",
    "url":
    "https://demo.ni-sp.com/enginframe/download?file=/8b45963b96801790f315dff51718ca9cb23db834//results-112.out.gz&_spooler=spooler:///opt/nisp/enginframe/spoolers/efadmin/tmp9559103907370519188.ef&_size=51&_plugin=fm",
    "size": {
      "value": 51,
      "unit": "bytes"
    }
  },
  {
    "path": "/",
    "vroot": "8b45963b96801790f315dff51718ca9cb23db834",
    "name": "results-87.out",
    "type": "file",
    "modified": "2025-04-08T11:31:00+00:00",
    "url":
    "https://demo.ni-sp.com/enginframe/download?file=/8b45963b96801790f315dff51718ca9cb23db834//results-87.out&_spooler=spooler:///opt/nisp/enginframe/spoolers/efadmin/tmp9559103907370519188.ef&_size=0&_plugin=fm",
    "size": {
```

```

        "unit": "bytes"
    }
}
]
# Download a file from the Spooler
> efpclient spoolers download -u
spooler:///opt/nisp/enginframe/spoolers/efadmin/tmp9559103907370519188
.ef -f results-112.out.gz
Content saved to results-112.out.gz

```

The EF Portal REST API offers a powerful and comprehensive control of EF Portal via own API clients or the Python-based efpclient installable from PyPi or as download.

EF Portal Modern Look and Feel

The EF Portal look and feel has been updated to a more modern design. Here are some examples:

The screenshot displays the EF Portal HPC Workspace interface. At the top, there is a navigation bar with the logo 'EF PORTAL HPC Workspace' and user information 'Welcome, efdadmin'. Below the navigation bar is a sidebar menu with categories like Data, Infrastructure, Workloads, Services, and Batch. The main content area is titled 'Hosts' and features a search bar and a 'Refresh' button. The hosts are displayed in a grid of cards, each representing a host (demo1 through demo7). Each card shows the host's status (Ok or Unavailable), the number of jobs, the current load, and the memory usage. A legend at the bottom right indicates that 6 hosts are 'Ok' and 1 is 'Unavailable'.

Host Name	Status	Jobs	Load	Memory
demo	Ok	0/2	1.03 (r1m)	0/1048576B
demo2	Ok	0/1	0.00 (r1m)	0/1048576B
demo3	Ok	0/1	0.00 (r1m)	0/1048576B
demo4	Ok	0/1	0.00 (r1m)	0/1048576B
demo5	Ok	0/1	0.00 (r1m)	0/1048576B
demo6	Ok	0/1	0.00 (r1m)	0/1048576B
demo7	Unavailable	0/1	0.00 (r1m)	0/1048576B

EF PORTAL HPC Workspace

Welcome, efadmin Switch to User View Logout

Menu ✕ Jobs

Search in all columns Refresh

ID	Status	Owner	Queue	Submission Time	Running on	Nice	Job Name
85	Done	efadmin	test	Yesterday 11:13:53	demo	0	Job_Compress_Image_078.png
86	Exit	efadmin	test	Yesterday 12:35:32	demo	0	Linux_Desktop
87	Done	efadmin	test	Today 11:31:07	demo	0	Job_Compress_slurm-112.out

Page 1 of 1 25 View 1 - 3 of 3

Clusters

cluster [READY]
dcvsm_cluster1 [READY]

Filters

All
Submitted Today
Running
Pending
Finished

EF PORTAL HPC Workspace

Welcome, efadmin Switch to User View Logout

Menu ✕ Services

New Import Refresh

Name	Status	Author	Last modification	Published Folder	Availability
Sample AWS Batch Compre...	Not publi...	NI SP	Dec 13, 2018 by NI SP		
Sample AWS Batch MNP Job	Not publi...	NI SP	Dec 17, 2018 by NI SP		
DD Upload Support (do not...	Published	NI SP	Feb 10, 2017 by NI SP	services	all-users
Unzip file	Published	NI SP	Feb 10, 2017 by NI SP	services	all-users
Job History	Not publi...	NI SP	Feb 28, 2024 by NI SP		
Job Submission	Not publi...	NI SP	Feb 04, 2015 by NI SP		
Jupyter Notebook	Published	NI SP	Dec 30, 2024 by NI SP	services	all-users
Sample Compress Job	Published	NI SP	Feb 10, 2017 by NI SP	services	all-users
Linux Desktop	Published	NI SP	Yesterday 19:02:22 by efadmin	services	all-users
Windows Desktop	Published	NI SP	Oct 28, 2014 by NI SP	services	all-users

Page 1 of 1 20 View 1 - 10 of 10

Filters

All
Published
Not published

Availability

all-users

The former nice-jump theme is still available in case of interest and can easily be configured.

Configurable Quick Commands for Host Information

To access information on Linux cluster hosts quickly EF Portal 2025.0 introduces quick commands available in the upper right of the respective host information. This is supported for Linux schedulers and access to the hosts can be configured via ssh or a scheduler command.

By default ps, top, free, w, df are configured:

Hosts: demo

PS W Top Free DF Refresh

Status **Ok (IDLE)**

Load

Job Slots

CPU run queue length:

Averaged one minute (r1m) **0.80**

Physical memory

Free tmp space **0**

The output of the “top” command e.g. looks like this:

EF PORTAL HPC Workspace Welcome, eadmin Settings Switch to Admin View Logout

Menu Hosts: demo PS W Top Free DF Refresh

Output of Command 'Top' on host 'demo':

```

top - 08:17:46 up 82 days, 14:00, 5 users, load average: 0.67, 0.76, 0.58
Tasks: 158 total, 1 running, 156 sleeping, 1 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3820.0 total, 323.4 free, 2174.6 used, 1613.3 buff/cache
MiB Swap: 4096.0 total, 4042.7 free, 53.2 used, 1645.5 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
3844591 eadmin   20   0 11932  5504 3456 R 10.0   0.1   0:00.01 top
  1 root     20   0 22924 10752 7040 S  0.0   0.3 55:45.48 systemd
  2 root     20   0    0    0    0 S  0.0   0.0 0:01.53 kthread
  3 root     20   0    0    0    0 S  0.0   0.0 0:00:00 pool_workqueue_release
  4 root     0 -20    0    0    0 I  0.0   0.0 0:00:00 kworker/R-rcu_g
  5 root     0 -20    0    0    0 I  0.0   0.0 0:00:00 kworker/R-rcu_p
  6 root     0 -20    0    0    0 I  0.0   0.0 0:00:00 kworker/R-slub
  7 root     0 -20    0    0    0 I  0.0   0.0 0:00:00 kworker/R-netns
10 root     0 -20    0    0    0 I  0.0   0.0 0:00:00 kworker/0:0H-events_highpri
12 root     0 -20    0    0    0 I  0.0   0.0 0:00:00 kworker/R-mm_pg
13 root     20   0    0    0    0 I  0.0   0.0 0:00:00 rcu_tasks_kthread
14 root     20   0    0    0    0 I  0.0   0.0 0:00:00 rcu_tasks_rude_kthread
15 root     20   0    0    0    0 I  0.0   0.0 0:00:00 rcu_tasks_trace_kthread
16 root     20   0    0    0    0 S  0.0   0.0 3:04.22 ksoftirqd/0

```

Close Refresh

ID	Status	Owner	Queue	Job Name
<input type="checkbox"/> 43	Done	efadmin	test	ps

Using either one of the following conf file:

- `$EF_ROOT/plugins/hydrogen/conf/host.commands.conf` (preinstalled)
- `$EF_CONF_ROOT/plugins/hydrogen/host.commands.conf`

admins can define one or more global actions to run custom commands on the host the user is viewing, and display the command output in a dialog window. Quick Commands are available to admins only by default.

Administrators can also allow regular users to use and also override admin defined commands, setting the configuration parameter `HY_HOST_INFO_COMMANDS_ADMIN_ONLY` to "false" in either `$EF_ROOT/plugins/hydrogen/conf/host.commands.conf` or `$EF_CONF_ROOT/plugins/hydrogen/host.commands.conf`. In this case, users can define their actions using the file `$HOME/.ef/host.commands.conf`.

New commands can be added with the syntax “description|command|question”. A question can be configured for further input to the command which is stored in %INPUT% and can then be used in the command replacing the value. Pipes are allowed in the command as well escaped with \.

The remote command to execute can be configured. The default is ssh with examples for SLURM and LSF.

Every admin user - and standard users as well in case allowed - can configure their own commands overwriting the system-wide configured commands in the file

~/.ef/host.commands.conf.

Example configuration in host.commands.conf:

```
# System processes
PS|ps -ef
W|w
# Performance monitoring
Top|top -b -n 1 -w 120 --sort %CPU
Free|free -h
# Storage information
DF|df -h
DF Grep|df -h \|| grep %INPUT%|Grep for
# SLURM
#sinfo|sinfo -l
#show hosts|scontrol show nodes
#show node|scontrol show node=%INPUT%|Which host to show?
# Hardware information
#lspci|lspci
#lsusb|lsusb
```

In ui.hydrogen.conf (\$EF_ROOT/plugins/hydrogen/conf/ui.hydrogen.conf or \$EF_CONF_ROOT/plugins/hydrogen/ui.hydrogen.conf) the Host Info functionality can be configured:

```
# Host Info
HY_HOST_INFO_SERVICE_URI="//ui.hydrogen/grid.host.info.data"
HY_HOST_INFO_COMMANDS_CONF="host.commands.conf"
HY_HOST_INFO_COMMANDS_ADMIN_ONLY="true"
HY_HOST_INFO_COMMANDS_REMOTE_CMD="ssh" # ssh from EF server
as logged in user
# HY_HOST_INFO_COMMANDS_REMOTE_CMD="srun -w" # SLURM
# HY_HOST_INFO_COMMANDS_REMOTE_CMD="lsrun -m" # LSF
```

Prevent File Downloads by Optional Allowlist

To prevent download, preview or streaming of files via EF Portal and allow only specific users in case to download files a new feature and configuration files have been introduced:

- `$EF_ROOT/plugins/fm/conf/fm.download.allowed.conf`
- `$EF_CONF_ROOT/plugins/fm/fm.download.allowed.conf`

```
# Every line is a user or a regex matching a set of user
# Comments with # are supported as part of the configuration
#efadmin
#ef.*
#root
```

Limiting the Hosts Displayed

In certain cases the host list should only show a limited set of hosts. This can be implemented with the new configuration introduced via

- `$EF_ROOT/plugins/hydrogen/conf/list.hosts.custom.filter.conf`
- `$EF_CONF_ROOT/plugins/hydrogen/conf/list.hosts.custom.filter.conf`

With this configuration only hosts matching certain regexps are shown in the Custom View available in the Filters on the right side:

```
# Here you can configure a custom filter for List Hosts view
# to show only a selected group of hosts based on their names.
# Every line not commented is a host name or a regex matching a set
of host names.
# Examples:
#   node-1
#   node-2
#   node-[3-4]
#   node-[1-2,6-7].*
demo
demo[4-5]
```

So the standard display of hosts changes from:

The screenshot shows the EF Portal HPC Workspace interface. The main content area displays a list of hosts under the heading "Hosts". The hosts are arranged in a grid, each with a card showing its name, status, jobs, load, and memory usage. The status of each host is indicated by a colored bar: green for "Ok" and red for "Unavailable".

Host Name	Status	Jobs	Load	Memory
demo	Ok	1/2	0.96 (r1m)	0/1048576B
demo2	Ok	0/1	0.00 (r1m)	0/1048576B
demo3	Ok	0/1	0.00 (r1m)	0/1048576B
demo4	Ok	0/1	0.00 (r1m)	0/1048576B
demo5	Ok	0/1	0.00 (r1m)	0/1048576B
demo6	Ok	0/1	0.00 (r1m)	0/1048576B
demo7	Unavailable	0/1	0.00 (r1m)	0/1048576B

To the filtered host list when selecting “Custom View”:

The screenshot shows the EF Portal HPC Workspace interface with a filtered host list. The main content area displays three host cards: demo, demo4, and demo5. All three hosts have a status of "Ok".

Host Name	Status	Jobs	Load	Memory
demo	Ok	1/2	1.24 (r1m)	0/1048576B
demo4	Ok	0/1	0.00 (r1m)	0/1048576B
demo5	Ok	0/1	0.00 (r1m)	0/1048576B

The custom view can be selected as default view with a configuration in `ui.hydrogen.conf`.

Jupyter Notebook Integration

EF Portal now offers Jupyter Notebook integration with a new service included in the EF Portal services distribution. Using the Jupyter Notebook service the user can upload a notebook or select a notebook on the server side via Remote File Browser:

Other Updates

Other Features and Improvements

- A new Star-CCM+ service template has been added.
- File Manager: Display of clickable URLs for files ending with .url
- File Manager: added more file types to preview as text files
- VDI: Connection Quick Access bar in session preview can be hidden via configuration

Security / Upgrades

- Upgraded Apache Tomcat to latest version 9.0.102
- Upgraded MyBatis to 3.5.6
- Upgraded Apache Commons IO to 2.18.0
- Upgraded Google Guava to 32.0.0-jre
- Upgraded Xalan-J to 2.7.3
- Upgraded Xalan-J Serializer to 2.7.3
- Upgraded Xerces2-J to 2.12.2

Demo Video

Please check out the EF Portal 2025.0 video, where we walk you through all the new features:
<https://www.youtube.com/watch?v=ILoMq2OkCbl>:

