

Nodejs and Reporting with Jenkins

by Prasant Sutaria

Pre-requisite: The Jenkins job is configured as explained in [Integration with Jenkins](#)

The reporting API requires the following three parameters which are provided by Jenkins job:

Parameters Name	Field Type
JOB_NAME	Mandatory provided by Jenkins
BUILD_NUMBER	Mandatory provided by Jenkins
*PERFECTO_TAGS	Optional Custom parameter

* PERFECTO_TAGS is option parameter and its name can be modified.

There are several ways to access the above parameters in our script:

1. Access as an environment variable

Jenkins exposes the above metadata as environment variables; we can use process.env to access the environment variables.

Env variables

```
value // If execution is not happening via Jenkins these variables can be set to provide default

var perfectoJobName = "<<Default Job Name>>";
var perfectoJobNumber = "<<Default Job Number>>";
var perfectoJobTags = [];

// Jenkins exposes its Job Name as JOB_NAME env variable
if (process.env.JOB_NAME !== undefined) {
  perfectoJobName = process.env.JOB_NAME;
}

// Jenkins exposes its Job Number as JOB_NUMBER env variable
if (process.env.BUILD_NUMBER !== undefined) {
  perfectoJobNumber = parseInt(process.env.BUILD_NUMBER);
}

// In Jenkins, one can create variable name PERFECTO_TAG
// Delimiter can be change based on need
if (process.env.PERFECTO_TAGS !== undefined) {
  perfectoJobTags = process.env.PERFECTO_TAGS.split(",");
}

// Initializing Perfecto Execution Context
var executionContext = new Reporting.Perfecto.PerfectoExecutionContext({
  webdriver: browser.driver,
  job: new Reporting.Model.Job({
    jobName: perfectoJobName,
    buildNumber: perfectoJobNumber
  }),
  tags: perfectoJobTags
});

// Initializing Perfecto Reporting Client
browser.reportingClient = new Reporting.Perfecto.PerfectoReportingClient(executionContext);
```

Note: If not using Jenkins still the above code can be used, we have to create temporary environment variable or we can use a batch file or a shell script like the following to set environment variables:

Shell Script for Unix based Systems

```
#!/bin/sh
echo "::Setting up environment variables"

# Perfecto JOB_NAME environment variable
export JOB_NAME="My Job Name"

# Perfecto BUILD_NUMBER environment variable
export BUILD_NUMBER=1

# Perfecto PERFECTO_TAGS environment variable. Comma separated tag names.
export PERFECTO_TAGS=Tag1,Tag2

# invoking test run
npm run test
```

2. Pass command line argument to NPM

Tested On:

Node Version	9.5.0
NPM Version	6.8.0

In the package.json file, modify the scripts entry as following:

package.json

```
{
  "name": "npm Sample",
  "version": "1.0.0",
  "description": "npm sample for providing command line arguments",
  "main": "index.js",

  "scripts": {
    "test": "node ./index.js $"
  },

  "author": "Prasants",
  "license": "ISC",
  "dependencies": {
  }
}
```

Script

```
var perfectoJobName = "<<Default Job Name>>";
var perfectoJobNumber = "<<Default Job Number>>";
var perfectoJobTags = [];

var argumentsLen = process.argv.length;

// 1st and 2nd arguments are reserved for node env and script file name
// 3rd command argument as Job Name
if (argumentsLen>2) {
  perfectoJobName = process.argv[2];
}

// 4th command argument as Job Number
if (argumentsLen>3) {
  perfectoJobNumber = parseInt(process.argv[3]);
}

// 5th command argument as tags
// Delimiter can be change based on need
if (argumentsLen>4) {
  perfectoJobTags = process.argv[4].split(",");
}

// Initializing Perfecto Execution Context
var executionContext = new Reporting.Perfecto.PerfectoExecutionContext({
  webdriver: browser.driver,
  job: new Reporting.Model.Job({
    jobName: perfectoJobName,
    buildNumber: perfectoJobNumber
  }),
  tags: perfectoJobTags
});

// Initializing Perfecto Reporting Client
browser.reportingClient = new Reporting.Perfecto.PerfectoReportingClient(executionContext);
```

Command to execute test:

Command to run test

```
npm run test "My Job" 1 "tag1,tag2"
```

Command to execute test using Jenkins environment variables:

Command to run test using Jenkins

```
npm run test $JOB_NAME $BUILD_NUMBER $PERFECTO_TAGS
```