



This repository

Pull requests

Issues

Marketplace

Gist

rse / node-prince

<> Code

! Issues 1

🔗 Pull requests 2

📁 Projects 0

Insights ▼

Node API for executing PrinceXML via prince(1) CLI <https://www.npmjs.org/package/prince>

📦 96 commits

🔗 1 branch

📦 28 releases

Branch: master ▼

New pull request

Create

Upload



rse bump version before release

.gitignore	ignore some files
.npmignore	ignore temp dir
Gruntfile.js	bump year in all copyright messages
Makefile	remove obsolete entry in Makefile
README.md	increase default maximum stdout/stderr buffer to 10MB and let the use...
eslint.json	update dependencies
jshint.json	place initial version under version control
package.json	bump version before release
prince-api.js	increase default maximum stdout/stderr buffer to 10MB and let the use...

 [prince-npm.js](#) bump year in all copyright messages

 [shtool](#) use local version of my GNU shtool for the platform checking


 [test.html](#) place initial version under version control

 [test.js](#) bump year in all copyright messages

README.md

Node-Prince

[Node](#) API for executing the XML/HTML to PDF renderer [PrinceXML](#) via `prince` CLI.


1★

npm install prince
8 dependencies version 1.4.0
5 dependents updated 2 months ago
866 downloads in the last month

dependencies up to date

Abstract

This is a [Node](#) API for executing the XML/HTML to PDF renderer [PrinceXML](#) CLI `prince` from [my](#) [github](#). The essential point of this Node extension is not primarily to just abstract away the asynchronous CLI. There are two other major points: First, this Node extension provides a fixed dependency, as other Node extensions do. PrinceXML can just depend (via their NPM `package.json` file) onto this extension. Second, as this extension is available across platforms — automatically download, locally unpack and use a PrinceXML distribution, then use the previously available global PrinceXML installation. Just depend on this Node extension and PrinceXML.

Installation

Use the Node Package Manager (NPM) to install this module locally (default) or globally (with optional `-g`).

```
$ npm install [-g] prince
```

ATTENTION: In case you are behind a corporate firewall, you usually have to configure your corporate proxy for this module. For this, either configure the proxy inside NPM (e.g. `npm config set proxy http://proxy:8080`) or alternatively set the environment variable `http_proxy` (e.g. `export http_proxy=http://proxy:8080`).

NOTICE: PrinceXML provides just distribution-specific and dynamically linked Linux binaries. The installation procedure tries to detect your particular distribution on `npm install [-g] prince` and download the necessary binaries. But because of the dynamic linking of PrinceXML binaries, it still can be that you first have to install system dependencies. For instance, under Debian 8.0 you first have to install the necessary system dependencies: `install nodejs nodejs-legacy npm libgif4 curl`.

Usage

```
var Prince = require("prince");

Prince()
  .inputs("test.html")
  .output("test.pdf")
  .execute()
  .then(function () {
    console.log("OK: done");
  }, function (error) {
    console.log("ERROR: ", util.inspect(error));
  })
```

API

- `Prince([options])`: `Prince` : constructor for the API. Call this once for every XML processing. This returns the Prince API for further method chaining.
- `Prince#binary(binary)`: `Prince` : set the path to the prince(1) binary. By default it is the path to the PrinceXML was found globally installed at the Node API installation time) or the path to the locally installed PrinceXML distribution (in case PrinceXML was not found globally installed at installation time). This returns the Prince API for further method chaining.

- `Prince#prefix(prefix): Prince` : set the path to the PrinceXML installation. This is the path to the PrinceXML installation (in case PrinceXML was found globally installed at the Node API installation time) or the path to the PrinceXML distribution (in case PrinceXML was not found globally installed at the Node API installation time). This returns the Prince API for further method chaining.
- `Prince#license(filename): Prince` : set the path to the PrinceXML license file. This is the path to the license file (in case PrinceXML was found globally installed at the Node API installation time) or the path to the PrinceXML license file (in case PrinceXML was not found globally installed at the Node API installation time). This returns the Prince API for further method chaining.
- `Prince#timeout(timeout): Prince` : set the execution timeout in milliseconds. The default is 10000 (10 seconds). This returns the Prince API for further method chaining.
- `Prince#maxbuffer(maxbuffer): Prince` : set the execution maximum stdout/stderr buffer size in bytes. The default is 10485760 (10MB). This returns the Prince API for further method chaining.
- `Prince#cwd(dirname): Prince` : set the current working directory for execution. The default is the current working directory. This returns the Prince API for further method chaining.
- `Prince#inputs(filename): Prince` : set one (in case `filename` is a string) or multiple (in case `filename` is an array of strings) input XML/HTML files. This returns the Prince API for further method chaining.
- `Prince#output(filename): Prince` : set the output PDF file. This returns the Prince API for further method chaining.
- `Prince#option(name, value[, forced]): Prince` : set a PrinceXML option name and value. The API knows the officially supported options of PrinceXML 9.0 and by default rejects any arbitrary options. Arbitrary options can be passed by setting `forced` to `true` in case a different PrinceXML version is used. This returns the Prince API for further method chaining.
- `Prince#execute(): Promise` : asynchronously execute the conversion process. This returns a Promise that resolves to the Prince API for further method chaining.

See Also

Companion Grunt task [grunt-princess](#)

License

Copyright (c) 2014-2017 Ralf S. Engelschall (<http://engelschall.com/>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

