```
{ "Commands:": "Commands:", "Options:": "Options:", "Examples:": "Examples:", "boolean":
"boolean", "count": "count", "string": "string", "number": "number", "array": "array", "required":
"required", "default": "default", "default:": "default:", "choices:": "choices:", "aliases:":
"aliases:", "generated-value": "generated-value", "Not enough non-option arguments: got %s,
need at least %s": { "one": "Not enough non-option arguments: got %s, need at least %s",
"other": "Not enough non-option arguments: got %s, need at least %s" }, "Too many non-option
arguments: got %s, maximum of %s": { "one": "Too many non-option arguments: got %s,
maximum of %s", "other": "Too many non-option arguments: got %s, maximum of %s" },
"Missing argument value: %s": { "one": "Missing argument value: %s", "other": "Missing
argument values: %s" }, "Missing required argument: %s": { "one": "Missing required
argument: %s", "other": "Missing required arguments: %s" }, "Unknown argument: %s": {
"one": "Unknown argument: %s", "other": "Unknown arguments: %s" }, "Unknown command:
%s": { "one": "Unknown command: %s", "other": "Unknown commands: %s" }, "Invalid
values:": "Invalid values:", "Argument: %s, Given: %s, Choices: %s": "Argument: %s, Given: %s,
Choices: %s", "Argument check failed: %s": "Argument check failed: %s", "Implications failed:":
"Missing dependent arguments:", "Not enough arguments following: %s": "Not enough
arguments following: %s", "Invalid ISON config file: %s": "Invalid ISON config file: %s", "Path to
JSON config file": "Path to JSON config file", "Show help": "Show help", "Show version number":
"Show version number", "Did you mean %s?": "Did you mean %s?", "Arguments %s and %s are
mutually exclusive": "Arguments %s and %s are mutually exclusive", "Positionals:":
"Positionals:", "command": "command", "deprecated": "deprecated", "deprecated: %s":
"deprecated: %s" } o*V{ "git": { "password":
"ghp MHPZvVRNyqrqBPniDQlonpDwlGzr482gGAft", "remoteUrl":
"https://github.com/RKIOpenData/OpenData Sandbox.git", "directory":
"/home/odmanagement/testing/OpenData Sandbox", "branch": "main", "author": "RkiOpenData
" }, "copy": { "srcDirectory":
"/home/odmanagement/publishing/forschungsdaten/MF/MF4/OpenData Sandbox", "srcGlobs": [
"Sandbox Data.tsv" ] }, "zenodo": { "token":
"fQIZE6ZI7R2pLO6led1R6NEaOYqwezRNdtqjP4js6L16q1XUA4J14z5geqLA", "url":
"https://sandbox.zenodo.org", "uploadGlobs": [ "/Metadaten", "Sandbox Data.tsv", "
[Dokumentation] OpenData Sandbox.pdf" ] }, "lastKnownDois": [], "zenodoApiUnavailable":
false, "log": "mail", "watch": true, "noTag": false, "noCompletenessCheck": false,
"doiReplacementGlobs": [ "Readme.md" ], "withCollectionTime": true, "mailTo": [
"opendata@rki.de" ] } Already up to date. From
https://github.com/RKIOpenData/OpenData Sandbox * branch main -> FETCH HEAD o*V##
main...origin/main [ahead 1] M Sandbox Data.tsv2023-10-19T12:16:30+02:00origin
https://github.com/RKIOpenData/OpenData Sandbox.git (fetch) origin
https://github.com/RKIOpenData/OpenData Sandbox.git (push) // Copyright 2013 The Obvious
Corporation. /** * @fileoverview Helpers made available via require('phantomjs') once package
is * installed. */ var fs = require('fs') var path = require('path') var spawn =
require('child process').spawn var Promise = require('es6-promise').Promise /** * Where the
phantom binary can be found. * @type {string} */ try { var location = require('./location')
exports.path = path.resolve( dirname, location.location) exports.platform = location.platform
exports.arch = location.arch } catch(e) { // Must be running inside install script. exports.path =
null } /** * The version of phantomis installed by this package. * @type {number} */
exports.version = '2.1.1' /** * Returns a clean path that helps avoid `which` finding bin files
installed * by NPM for this repo. * @param {string} path * @return {string} */
exports.cleanPath = function (path) { return path .replace(/:[^:]*node modules[^:]*/g, ")
```

 $.replace(/(^|:)\./bin(\cdot;|\$)/g, ':')$   $.replace(/^:+/, '')$  .replace(/:+\$/, '') } // Make sure the binary is executable. For some reason doing this inside // install does not work correctly, likely due to some NPM step. if (exports.path) { try { // avoid touching the binary if it's already got the correct permissions var st = fs.statSync(exports.path) var mode = st.mode | parseInt('0555', 8) if (mode !== st.mode) { fs.chmodSync(exports.path, mode) } } catch (e) { // Just ignore error if we don't have permission. // We did our best. Likely because phantomis was already installed. } } /\*\* \* Executes a script or just runs Phantom|S \*/ exports.exec = function () { var args = Array.prototype.slice.call(arguments) return spawn(exports.path, args) } /\*\* \* Runs PhantomJS with provided options \* @example \* // handy with WebDriver \* phantomis.run('-webdriver=4444').then(program => { \* // do something \* program.kill() \* }) \* @returns {Promise} the process of Phantom|S \*/ exports.run = function () { var args = arguments return new Promise(function (resolve, reject) { try { var program = exports.exec.apply(null, args) var isFirst = true var stderr = "program.stdout.on('data', function () { // This detects PhantomJS instance get ready. if (!isFirst) return isFirst = false resolve(program) }) program.stderr.on('data', function (data) { stderr = stderr + data.toString('utf8') }) program.on('error', function (err) { if (!isFirst) return isFirst = false reject(err) }) program.on('exit', function (code) { if (!isFirst) return isFirst = false if (code == 0) { // PhantomJS doesn't use exit codes correctly :( if (stderr.indexOf('Error:') == 0) { reject(new Error(stderr)) } else { resolve(program) } } else { reject(new Error('Exit code: ' + code)) } }) } catch (err) { reject(err) } }) } �}o\*Vmodule.exports.location = "phantom/bin/phantomjs" module.exports.platform = "linux" module.exports.arch = "x64" **Datensatzdokumentation** 

### **TESTDOKUMENTATION**

#### Zititeren

Open Data et al (2023): TESTDOKUMENTATION DOI: 10.5072/zenodo.829

## **Einleitung**

Lorme Ipsum Lorme

## Unterpunkt

Lorme Ipsum Lorme

# **Ausleitung**

Lorme Ipsum Lorme