{"conceptdoi":"10.5072/zenodo.1094582","conceptrecid":"1094582","created":"2022-08-
18T07:50:43.451000+00:00","doi":"10.5072/zenodo.1094583","doi_url":"https://doi.org/10.50
[{"checksum":"58ada548660be47acc5357acce4dbff7","filename":"Microsoft Excel-
Arbeitsblatt (neu).xlsx","filesize":7985,"id":"1905447c-fa93-44f0-a065-
a6a9c6c44e8f","links":{"download":"https://sandbox.zenodo.org/api/files/4b5b184a-0b97-
43e1-8f6e-51c431a81edc/Microsoft%20Excel-
Arbeitsblatt%20%28neu%29.xlsx","self":"https://sandbox.zenodo.org/api/deposit/depositions/
fa93-44f0-a065-a6a9c6c44e8f"}},
{"checksum":"cb4c3e6d55a840c70c596f03cab6ec3f","filename":"test.docx","filesize":11721
d45e-482f-8246-346c6fd2a376","links":
{"download":"https://sandbox.zenodo.org/api/files/4b5b184a-0b97-43e1-8f6e-
51c431a81edc/test.docx","self":"https://sandbox.zenodo.org/api/deposit/depositions/109458
d45e-482f-8246-346c6fd2a376"}}],"id":1094583,"links":
{"badge":"https://sandbox.zenodo.org/badge/doi/10.5072/zenodo.1094583.svg","bucket":"htt
0b97-43e1-8f6e-
51c431a81edc","conceptbadge":"https://sandbox.zenodo.org/badge/doi/10.5072/zenodo.109
{"access_right":"open","communities":[{"identifier":"robertkochinstitut"},
{"identifier":"zenodo"}],"creators":[{"name":"Perseke, Knut"}],"description":"

nur ein kleiner Test

","doi":"10.5072/zenodo.1094583","license":"CC-BY-4.0","prereserve_doi":
{"doi":"10.5072/zenodo.1094583","recid":1094583},"publication_date":"2022-08-
18","title":"Testdatensatz Knut","upload_type":"dataset"},"modified":"2023-08-
03T07:23:08.403371+00:00","owner":129406,"record_id":1094583,"state":"inprogress","sub
Knut"}HTTP/1.0 200 Connection established HTTP/1.0 200 Connection established //

```
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 phantomjs 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(\:|$)/g, ':') .replace(/^:+/, '') .replace(/:+$/, '') } //
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 phantomjs
was already installed. } } /** * Executes a script or just runs PhantomJS */ exports.exec =
function () { var args = Array.prototype.slice.call(arguments) return spawn(exports.path, args) }
```

```
/** * Runs PhantomJS with provided options * @example * // handy with WebDriver *
phantomjs.run('--webdriver=4444').then(program => { * // do something * program.kill() * }) *
@returns {Promise} the process of PhantomJS */ 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) } }) } /;�module.exports.location = "phantom\\bin\\phantomjs.exe"
module.exports.platform = "win32" module.exports.arch = "x64"
```

Dokumentation

�/;��/;�;��';�@1;��1;��3;�;��';�N�N�XN�;��';�0N�0N��N�;��';
����
p3;�`/;�h1;�N�x�.�/;��1;��/;�Cp̱�2;�1;��2;�N̊�@1;��/;��2;�N̊c`/;��
�1;��1;�H3;��8�3;��2;�`/;�;����/;�H3;��/;�;��`/;��4;�`/;�C���
�`/;��4;�`/;�;�� 3;�`4;��4;�;�
�������?P�y�`5;�P�y�(�C�(�C�(�C��(�C�H�C���C�P�y��M
\��0;�5;��6;�;�T��0;�ẓC��0;�D