

# Sandbox Respository des Open Data Teams [Robert Koch-Institut | RKI](https://rki.de)  
 Nordufer 20 13353 Berlin [Open Data Team](#Sandbox-Respository-des-Open-Data-Teams) ---  
 \*\*Zitieren\*\* Open Data Team (\*\*2023\*\*): Sandbox Respository des Open Data Teams. [DOI:  
 10.5072/zenodo.1231224](https://doi.org/10.5072/zenodo.1231224) lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum. # Einleitung lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
 (/Sandbox\_Data.tsv) # Tabelle # Lizenz Der Datensatz "Hochfrequente Mental Health  
 Surveillance" ist lizenziert unter der [Creative Commons Namensnennung 4.0 International  
 Public License | CC-BY ](https://creativecommons.org/licenses/by/4.0/deed.de). Die im  
 Datensatz bereitgestellten Daten sind, unter Bedingung der Namensnennung des Robert Koch-  
 Instituts als Quelle, frei verfügbar. Das bedeutet, jede Person hat das Recht die Daten zu  
 verarbeiten und zu verändern, Derivate des Datensatzes zu erstellen und sie für kommerzielle  
 und nicht kommerzielle Zwecke zu nutzen. Weitere Informationen zur Lizenz finden sich in der  
 [LICENSE](/LICENSE) bzw. [LIZENZ](/LIZENZ) Datei des Datensatzes.// 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"
module.exports.platform = "linux" module.exports.arch = "x64"
```

# Sandbox Respository des Open Data Teams

---

Robert Koch-Institut | RKI  
Nordufer 20  
13353 Berlin

Open Data Team

---

## Zitieren

Open Data Team (**2023**): Sandbox Respository des Open Data Teams. DOI: [10.5072/zenodo.1231224](https://doi.org/10.5072/zenodo.1231224)

lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum.

## Einleitung

---

lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem.

- ipsum lorem ipsum lorem
- ipsum lorem ipsum lorem ipsum lorem ipsum
- lorem ipsum lorem ipsum lorem

ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum  
ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum.

