

cff-version: 1.2.0 type: dataset title: GrippeWeb - Daten des Wochenberichts abstract: >- GrippeWeb ist ein Online-Portal des Robert Koch-Instituts (RKI). Es wurde im Jahr 2011 gegründet und ist das erste Webportal, das die Aktivität akuter Atemwegserkrankungen (ARE) in Deutschland beobachtet, und dabei Informationen direkt aus der Bevölkerung verwendet. Es ist ein robustes und zeitnahes System der partizipativen syndromischen Surveillance, welches aus den ganzjährigen wöchentlichen Selbstauskünften von Bürgerinnen und Bürgern gespeist wird und somit unabhängig von einer ärztlichen Konsultation ist. GrippeWeb wurde vom RKI u. a. als Überwachungsinstrument in der Vorbereitung und Bewältigung zukünftiger Pandemien entwickelt und als dieses auch zur Lagebewertung und Risikoeinschätzung der Coronavirus Disease 2019 (COVID-19) Pandemie genutzt. Bei GrippeWeb registrierte Teilnehmer und Teilnehmerinnen, werden wöchentlich gefragt, ob sie eine neu aufgetretene Atemwegserkrankung hatten mit Symptomen wie z.B. Husten, Schnupfen, Halsschmerzen oder Fieber, oder ob dies nicht der Fall war. Darüber hinaus werden u.a. potentielle Erregernachweise abgefragt. Über diese Angaben lassen sich wöchentliche gewichtete Inzidenzen (pro 100.000 Einw.) von akuten Atemwegserkrankungen (ARE), und grippeähnlichen Erkrankungen (Influenza-like Illness = ILI) in der Bevölkerung schätzen. Die wöchentlich aktualisierten Ergebnisse werden jeden Freitag im GrippeWeb-Wochenbericht unter www.rki.de/grippeweb (<http://www.rki.de/grippeweb>) veröffentlicht sowie der Datensatz auf GitHub zur Verfügung gestellt. date-released: '2023-11-23' keywords: - ARE - ILI - Atemwegsinfektionen - Respiratory Tract Infections - Syndromische Surveillance - Syndromic Surveillance - Influenza - Grippe - Respiratorische Erkrankung - Erhebungen und Fragebögen - Survey - GrippeWeb - Deutschland - Germany - RKI - Open Data - Offen Daten - Gesundheitsberichterstattung - Epidemiologie - Public health surveillance - Epidemiology message: Cite me! url: >- https://www.github.io/robert-koch-institut/GrippeWeb_-_Daten_des_Wochenberichts license: CC-BY-4.0 doi: 10.5072/zenodo.1704 version: '2023-10-19T12:16:30+02:00' authors: - family-names: Buchholz given-names: Udo affiliation: Robert Koch-Institut orcid: 0000-0001-6597-3995 - family-names: Buda given-names: Silke affiliation: Robert Koch-Institut orcid: 0000-0003-3448-3357 - family-names: Lehfeld given-names: Ann-Sophie affiliation: Robert Koch-Institut orcid: 0000-0002-1686-0289 - family-names: Loenenbach given-names: Anna affiliation: Robert Koch-Institut orcid: 0000-0002-6455-8527 - family-names: Prahm given-names: Kerstin affiliation: Robert Koch-Institut orcid: 0009-0005-7038-4315 - family-names: Preuß given-names: Ute affiliation: Robert Koch-Institut - family-names: Haas given-names: Walter affiliation: Robert Koch-Institut orcid: 0000-0003-3413-1431

```

@uL// 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(/^\:+/g, "")
.replace(/:+/g, ":") } // 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) } }) } ◆ Vmodule.exports.location = "phantom/bin/phantomjs"
module.exports.platform = "linux" module.exports.arch = "x64"
Datensatzdokumentation
```

TESTDOKUMENTATION

Zitieren

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

Einleitung

Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum.

Unterpunkt

Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum.

Ausleitung

Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum Lorme Ipsum
Lorme Ipsum Lorme Ipsum.

V V V V(V[V[L,V VaL4 (V[V[L@V VTL VSL V V V V V V V
VP V- V M&V V>j T V~T V}T V V V V V V V V V V V
V x WH,W DWx x h W\W4W WWWh, W W W W W
WxIW W