

PHI Hide utility

GeekMustHave

Version 2.1e, 11/06/2019: Overview

Table of Contents

Security	1
Installation	1
Structure	1
Postman Run fileDoc	1
phi-hide.json file	2
Node app	4
CLI Libraries	4
Additional libraries	5
Code tricks	5
Console.Log shortened	5
Global replace	5
Passing parameters	5
Development files	6
Using PHI-HIDE	7
Compiling PHI-Hide	7
Document History	7

PHI Hide is a simple utility to help in post processing of the Postman **run** JSON file to replace PHI information from a list of replacements in a separate JSON file.

To view or download the PDF version of this document please click on this [link](#)

Security

While this document and project is about Protected Health Information PHI great care has been take to make certain no PHI exist in the documents or code.

Installation

After the GIT repository has been cloned or copied the installation is required to get all the Node dependancies.

```
npm init
```

This may take some time to complete.

Structure

Postman **Run** fileDoc

The Postman test output **Run** file has beneficiary information that needs to be replace with non-identifying information.

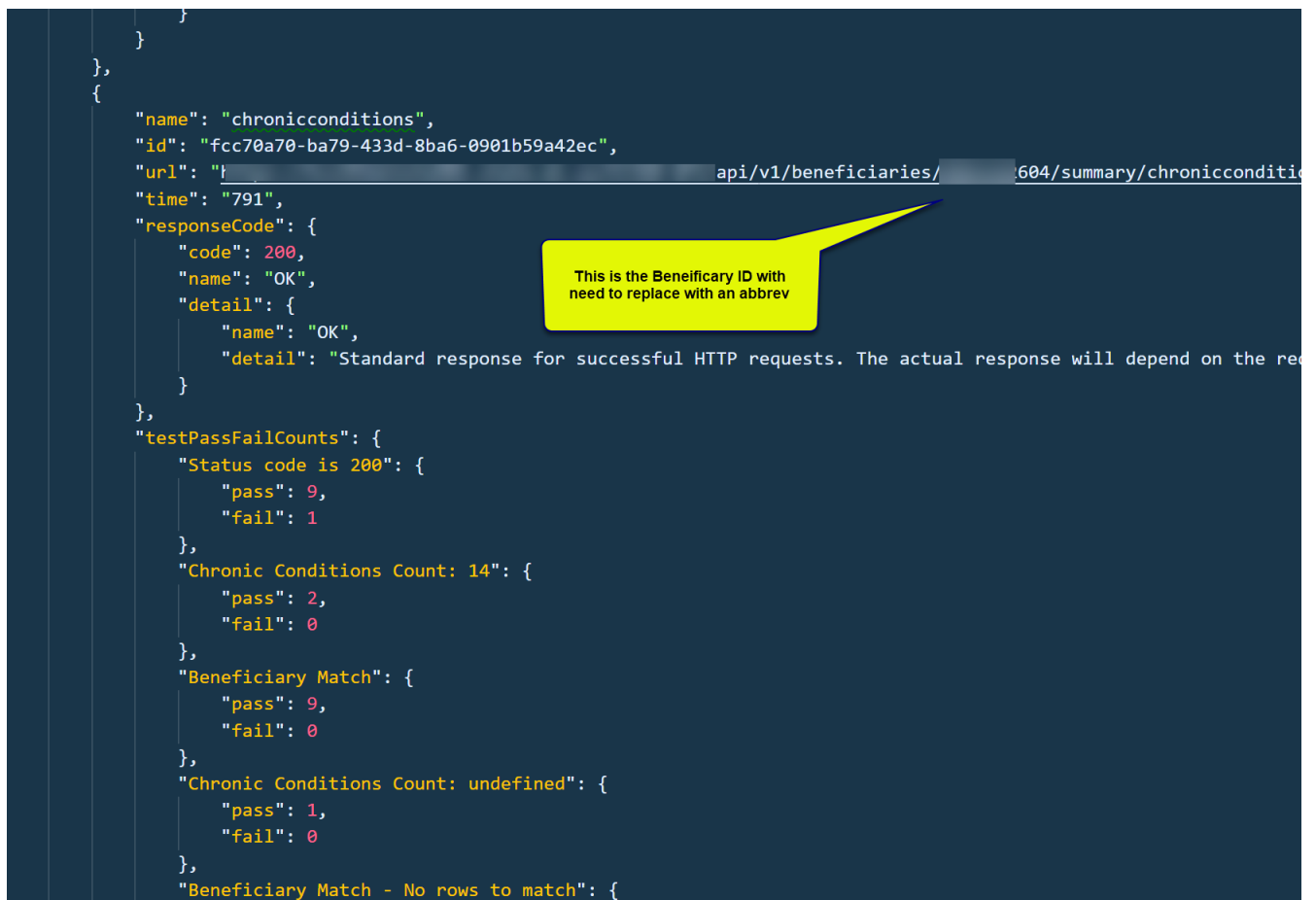


Figure 1. Postman run example

phi-hide.json file

This `phi-hide.json` file is used to drive the Postman run process to run (x) interactions based on the number of values in this file.

```
[
  {
    "beneID": "[REDACTED]604",
    "abbrev": "{604}"
  },
  {
    "beneID": "[REDACTED]848",
    "abbrev": "{848}"
  },
  {
    "beneID": "[REDACTED]069",
    "abbrev": "{069}"
  },
  {
    "beneID": "[REDACTED]621",
    "abbrev": "{621}"
  },
  {
    "beneID": "[REDACTED]590",
    "abbrev": "{590}"
  },
  {
    "beneID": "0018189085",
    "abbrev": "{085}"
  },
  {
    "beneID": "[REDACTED]799",
    "abbrev": "{799}"
  },
  {
    "beneID": "[REDACTED]006",
    "abbrev": "{006}"
  },
  {
    "beneID": "[REDACTED]484",
    "abbrev": "{484}"
  },
  {
    "beneID": "[REDACTED]929",
    "abbrev": "{929}"
  }
]
```

Figure 2. *phi-hide* file

The definitions are

1. beneID - 10 digit file identifying a beneficiary (PHI)
2. abbrev - What will be displayed in the beneId's place

This file need to hid the PHI beneficiary ID and provide a way to identify the test using the abbrev.



This PHI hide file contains PHI and should not be distributed.

Node app

This app will be call **phi-hide** it is a simple replacement utility.

There can be 3 optional arguments

1. **--inFile** - pathname to the Postman test output **Run** file
2. **--outFile**- pathname to the Revised test output **Run** file
3. **--phiFile**- pathname of the PHI replacement values



Argument names are double-hyped **--inFile**

CLI Libraries

Following are standard libraries I use for CLI utilities. You can install them manually using NPM or install them with the project. The manual installation commands are shown below.

```
npm install --save chalk clear figlet inquirer
```

The dependencies are

Dependencies

```
"dependencies": {  
  "chalk": "^2.4.1", ①  
  "clear": "^0.1.0", ②  
  "figlet": "^1.2.0", ③  
  "inquirer": "^6.0.0" ④  
}
```

- ① Added color to text
- ② Clears the terminal window
- ③ Draws big text ASCII characters
- ④ Ask questions saves responses (Future)

Additional libraries

I have used the following libraries in this project. You can install this library manually using NPM or install it with the project. The manual installation commands are shown below.

```
npm install --save minimist pause
```

minimist - Used to process the arguments text after the **node phi-hide** **pause** - Provides a pause function with message

Code tricks

Console.Log shortened

The **console.log** statement can be shortened to just **log** with

console.log shortened

```
// -- Abbreviation for console.log  
const log = console.log;
```

Global replace

The standard node **replace** function only replaces the first occurrence.

To get **global** replace to work the **RegExp** regular expression function is used. The **g** option is what sets the global

Global replace

```
inputText = inputText.replace(new RegExp(thisText, 'g'), thatText);
```

Passing parameters

To quickly interpret the parameters string the **minimist** library was used.

The code to extract an object with the parameters in it

Parameters Code

```
// --- Minimist is a parameter parsing function slice(2) is that arguments after  
0=node, 1=program name  
//      argv function for all parameters  
const parms = require('minimist')(process.argv.slice(2));
```

The result for the code is an object named **parms**.

minimist Result

```
parms: {
  "_": [],
  "inFile": "postman-test-run.json",
  "phiFile": "phi-list.json",
  "outFile": "hhh.txt"
}
```

The last step is to get the **values** for the three input parameters.

The **fail** variable is used later in the code to display help message for **phi-hide**.

Get Parameter values

```
// --- Handle arguments send as --inFile abc.json --outFile abc.lll --phiFile
alpha.json
if (!parms.hasOwnProperty('inFile')){fail=true}else{inFile=parms['inFile']};
if (!parms.hasOwnProperty('outFile')){fail=true}else{outFile=parms['outFile']};
if (!parms.hasOwnProperty('phiFile')){fail=true}else{phiFile=parms['phiFile']};
```

Development files

1. **inFile**: ./postman-test-run.json
2. **outFile**: ./postman-test-run.out.json
3. **phiFile**: ./phi-hide.json

The **phiFile** JSON file needs to be in a special format.

```
1 [
2   {
3     "beneID": "██████99",
4     "abbrev": "001"
5   },
6   {
7     "beneID": "██████08",
8     "abbrev": "002"
9   },
10  {
11    "beneID": "██████19",
12    "abbrev": "003"
13  }
14 ]
```

Figure 3. *phiFile* format

Using PHI-HIDE

```
PS F:\users\jschust2\Dropbox\myDev\phi-hide> node phi-hide --inFile postman-test-run.json --phiFile phi-list.json --outFile hhh.txt

[PHI-HIDE]

Input File: postman-test-run.json
Output File: hhh.txt
PHI File: phi-list.json

Input file: postman-test-run.json loaded.
PHI file: phi-list.json loaded.

Processing...
Abbrev: {604} replace will be done 2 times
Abbrev: {848} replace will be done 1 times
Abbrev: {069} replace will be done 1 times
Abbrev: {621} replace will be done 0 times
Abbrev: {590} replace will be done 0 times
Abbrev: {085} replace will be done 0 times
Abbrev: {799} replace will be done 0 times
Abbrev: {006} replace will be done 1 times
Abbrev: {484} replace will be done 0 times
Abbrev: {929} replace will be done 1 times

Output file written: hhh.txt

[PHI-HIDE]
```

Figure 4. Running phi-hide

Compiling PHI-Hide

In order to run PHI-Hide without **Node** the Node application can be compiled.

The **pkg** Node command is used to perform the compilation.

Installation of the **pkg** compiler

pkg installation

```
npm install pkg -g ①
```

① The **-g** Indicates **pkg** to be installed globally

The **PHI-Hide-win.exe** can be copied and or renamed, to any system with or without **Node** and it will run.

Document History

Table 1. Document History

Date	Version	Author	Description
11/06/2019	V2.1e	JHRS	Updated to prep for GitHub (OJ)
12/06/2018	v2.1d	JHRS	Updated with PKG complier information

Date	Version	Author	Description
07/04/2018	V2.1c	JHRS	much of the app finished
06/29/2018	V2,1b	JHRS	Initial version