***TSB Profile Generator***

By MalsKippetje | SneakerSven #1037

*Insta: @sneaker.sven*

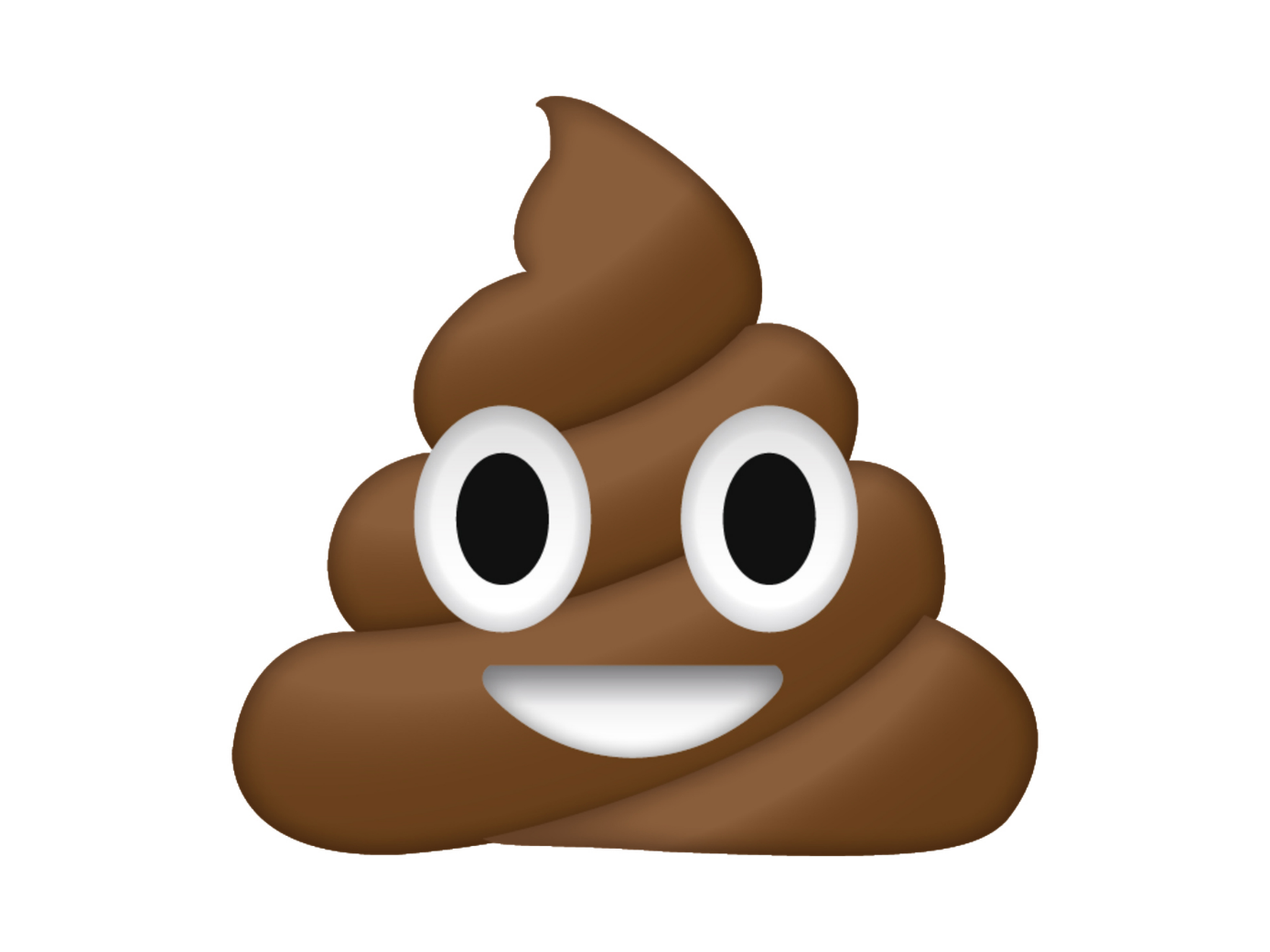
****

Table of contents

[Introduction 3](#_Toc65594919)

[Retrieve credit card details 4](#_Toc65594920)

[Generate profiles 5](#_Toc65594921)

[Address 1 jigging patterns and examples 5](#_Toc65594922)

[Jigging settings 7](#_Toc65594923)

[Address details and jigging 8](#_Toc65594924)

# Introduction

Welcome to the TSB Profile Generator!

With this generator you can create thousands of *The Shit Bot* ready profiles in mere seconds. All the program needs is a text file containing your creditcard details, which you can extract from your current TSB profiles export using this generator, too!

This guide will show you the principles by which the generator jiggs your address, and how to use the generator. First, the functionality for retrieving your credit card details from your TSB export will be explained. Secondly, the profile generating process will be explained. The program itself will also guide you through the different processes. This guide is to provide more detailed information about the generator.

Your creditcard details are untouched and will not be tampered with. To ensure the transparancy of the generator, the code is open source and anyone can clone the repository. You are free to clone the code and make any changes as you see fit. A link to the repository can be found in the *#welcome* channel of the TSB Profile Generator discord.

If you have any problems using the generator, dont hesitate to contact the devs on the discord channel!

Good luck pooping! 💩

# Retrieve credit card details

The program can retrieve your credit card details from your profiles export from TSB. Exporting your profiles within TSB will create a .json file. You have to place this file in the input folder of the generator. This folder is located at *C:/TSB Profile Generator/input*. This folder will be created when you first run this functionality. The program will ask you to place your TSB export into the folder, and to rerun the program after you’ve done that.

The .txt that gets generated by this functionality can be used for generating TSB profiles. The generating of the profiles and the jigging of the addresses will be explained further in the next chapter of this guide.

*The generated .txt will be located in the C:/TSB Profile Generator/output folder.*

If you don’t have a TSB profiles export ready to use, that is no problem! You can type the credit card details yourself in the .txt file by the following pattern:

*Pattern:*

< PROFILENAME > ; < PHONE NUMBER> ; < CC NUMBER > ; < EXPIRY DATE > ; < CVV >

*Example:*

REVOLUT 1;+31678654312;0000 0000 0000 0000;11 / 25;053

Dont forget the ; in between each field, don’t add spaces surrounding them. Only one credit card per line!

# Generate profiles

This chapter touches the jigging of your address. This functionality takes your credit card details in a .txt file as input, and will give you a .json file containing TSB ready profiles as output. The input .txt can also be generated using your TSB profiles output. For generating the input file, refer to the previous chapter.  
  
The program will ask you about some jigging settings, and it will ask for your address details. The jigging patterns, jigging settings and address details will be explained in the following subchapters.  
  
*The generated .json file will be located at C:/TSB Profile Generator/output.*

## Address 1 jigging patterns and examples

Let’s start by explaining the jigging patterns. The program will first ask you about your jigging settings. First you choose one of the 6 Address 1 jigging patterns, or a mix of all patterns. Below is a table showing all available jigging patterns.

*Note that the ‘TSBPG’ in the examples stands for a randomized piece of text, and the ‘534’ for a randomized number.*

For the examples below, the user input is “Streetname 1”.

|  |  |
| --- | --- |
| **Pattern no#** | **Address 1 jigging patterns** |
| 1 | TSBPG Streetname 1 |
| 2 | TSBPG Streetname 1 534 |
| 3 | TSBPG Streetname 1 534 TSBPG |
| 4 | TSBPG Streetname 1 TSBPG |
| 5 | Streetname 1 534 TSBPG |
| 6 | Streetname 1 TSBPG |
| 7 | Randomized mix of all of the patterns above |

*Table 1: Address 1 jigging patterns*

Do you have any suggestions for another jigging pattern? Please let us know in the *#suggestions-and-todo-list* channel in the discord!

The address 2 field will also get jigged by the program. This has a 75% chance of happening, so that not every profile uses an address 2. If input anything into address 2, it will not get jigged. If you leave it empty, it will randomly choose a prefix from the list of prefixes below, and add a random suffix. The pattern for this suffix is 1 random number between 1 and 5, and a random letter between A and E. See the table below for a list of all the available prefixes, and for address 2 jigging examples.

|  |
| --- |
| **Address 2 prefixes** |
| Appt., Appartement, Floor, Verdieping, Suite, Room, Kamer |
| **Address 2 examples** |
| Appt. 3A  Suite 5E  Kamer 1C  Appartement 2D  *etc.* |

*Table 2: Address 2 jigging patterns*

## Jigging settings

After choosing your jigging pattern, the program will ask you for some more jigging settings. Provided below is a list of all settings the generator asks input for.

|  |  |
| --- | --- |
| **Setting no#** | **Description** |
| **1** | Use randomized delivery phone numbers. This is a yes or no question. |
| **1a** | (Only shown if setting no# **1** is yes) Beginning of phone number to use for valid random phone numbers.  *EG: +316 for Dutch phone numbers.* |
| **1b** | (Only shown if setting no# **1** is yes) Amount of random numbers that have to be added to the beginning of each phone number to generate valid phone numbers.  *EG: 8 for Dutch phone numbers.* |
| **2** | Use randomized first and last names for each profile. If you choose no, you will have to put in the first and last name to use at the address details. This is a yes or no question.  *(for now these names are Dutch first and last names)* |
| **3** | Add 0 to 2 random characters to the end of city field, with a space in between them. This is a yes or no question. |
| **4** | Use shipping address as billing address. This is a yes or no question. |
| **5** | Amount of times that the program has to generate a jigged address for each provided credit card. |

*Table 3: Jigging settings*

## Address details and jigging

After you set up the jigging settings, the program will ask for your address details. If you want a field to be empty, you can type nothing and just hit enter. If you leave address 2 empty, it will be jigged as shown in the subchapter called ‘Jigging patterns and examples’. Provided below is a table showing all the address details the program will ask input for.

|  |  |
| --- | --- |
| **Address field no#** | **Description** |
| **1** | First name  *Will only be shown when you don’t want random first and last names* |
| **2** | Last name  *Will only be shown when you don’t want random first and last names* |
| **3** | Address 1  *This will be jigging using the patterns shown in the subchapter called ‘Jigging patterns and examples’* |
| **4** | Address 2  *This will be jigging using the patterns shown in the subchapter called ‘Jigging patterns and examples’* |
| **5** | Zip  *This field will not be jigged by the program* |
| **6** | City  *This will be jigging using the patterns stated in the subchapter called ‘Jigging settings’* |
| **7** | Country  *This field will not be jigged by the program* |
| **8** | State  *This field will not be jigged by the program* |

Table 4: Address details

# Example of jigging

This chapter shows a few examples of jigging your address using the TSB Profile Generator. The example address we are going to use to jigg, will be stated first. As will the settings that get used.

## Example input

|  |  |
| --- | --- |
| **Jigging setting no#** | **Input** |
| 1 | Y |
| 1a | +316 |
| 1b | 8 |
| 2 | Y |
| 3 | Y |
| 4 | Y |
| 5 | 3 |

|  |  |
| --- | --- |
| **Field** | **Input** |
| **Address 1** | Colosseum 1 |
| **Address 2** |  |
| **Zip** | 1213NL |
| **City** | Hilversum |
| **Country** | Netherlands |
| **State** |  |

The input .txt we are going to use looks like this:

*REVOLUT 1;+31612345678;0000 0000 0000 0001;3 / 26;001*

*REVOLUT 2;+31612345678;0000 0000 0000 0002;3 / 26;002*

*REVOLUT 3;+31612345678;0000 0000 0000 0003;3 / 26;003*

*REVOLUT 4;+31612345678;0000 0000 0000 0004;3 / 26;004*

*REVOLUT 5;+31612345678;0000 0000 0000 0005;3 / 26;005*

## Example output

The output that is shown below will be copied from the output .json file.

TODO: Jigging pattern in jigging settings zetten, eerst jigging settings daarna jigging patterns tonen