



## **D4.3 Annex II**

### **Data Privacy Tool**

### **User Guide**

*DISCLAIMER: Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.*

*COPYRIGHT MESSAGE: © FAIR4Health Consortium, 2019*

*This report contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.*

## Document Information

GRANT AGREEMENT NUMBER 824666		ACRONYM: FAIR4Health	
Full title	Improving Health Research in EU through FAIR Data		
Horizon 2020 Call	SwafS-04-2018: Encouraging the re-use of research data generated by publically funded research projects		
Type of action	Research and Innovation Action		
Start Date	1 <sup>st</sup> December 2018	Duration	36 months
Website	<a href="http://www.fair4health.eu">www.fair4health.eu</a>		
Project Officer	Pepa Krasteva		
Project Coordinator	Carlos Luis Parra Calderón, Andalusian Health Service		
Report	Data Privacy Tool User Guide		
Related task	T4.2 Development of a security layer for the FAIR4Health platform		
Release date	November 2020		
Dissemination Level	Public		
Responsible Author	Ezelsu Simsek (P15 – SRDC)		
e-mail	<a href="mailto:ezelsu@srdc.com.tr">ezelsu@srdc.com.tr</a>		
Collaborators	Ali Anil Sinaci (P15 – SRDC), Mert Gencturk (P15 – SRDC)		
Keywords	Privacy, de-identification, anonymization, k-anonymity, l-diversity		

## Table of Contents

1. Introduction .....	4
2. Data Privacy Tool Functional Features and User Manual.....	4
2.1. De-identification Steps Manual .....	5
2.1.1. Verifying FHIR Repository .....	5
2.1.2. Selecting Attribute Types .....	6
2.1.3. Configuring De-identification Algorithms.....	7
2.1.3.1. Configuring Quasi-identifiers .....	7
2.1.3.2. Configuring Sensitive Attributes .....	10
2.1.3.3. Handling Sensitive Rare Values .....	11
2.1.4. De-identifying Data .....	13
2.1.4.1. Summary of Configurations .....	13
2.1.4.2. Viewing De-identification Details.....	14
2.1.4.3. Viewing Validation Details .....	16
2.1.4.4. De-identified Resources.....	17
2.1.4.5. Restricted Resources .....	18
2.1.4.6. Saving De-identified Data.....	19
2.1.4.7. Saving the Configuration and Use it Later.....	21
2.1.4.8. Exporting the Configuration and Import it Later.....	22
2.2. De-identification Methodology.....	24
2.2.1. De-identification Algorithms.....	24
2.2.1.1. Pass Through .....	24
2.2.1.2. Redaction.....	25
2.2.1.3. Substitution .....	26
2.2.1.4. Recoverable Substitution .....	27
2.2.1.5. Fuzzing .....	28
2.2.1.6. Date Shifting.....	29
2.2.1.7. Generalization .....	30
2.2.1.8. Replace .....	32
2.2.2. Privacy Criteria.....	32
2.2.2.1. K-anonymity.....	32
2.2.2.2. L-diversity .....	33
2.2.3. Information Loss and Privacy Risks .....	34
2.2.3.1. Information Loss .....	34

2.2.3.2.	Re-identification Risks .....	34
2.2.3.3.	Records Affected by Risks .....	34
2.3.	Tool Settings.....	34
2.3.1.	Changing Language (Localization) .....	34
2.3.2.	Toggle Sidebar & Full Screen .....	35
2.3.3.	Debugging.....	36
2.3.4.	Log Locations .....	37
2.3.5.	Help.....	38
References	.....	39

## List of Figures

Figure 1. Home Screen.....	4
Figure 2. Verifying FHIR Repository .....	5
Figure 3. Selecting Attribute Types.....	7
Figure 4. Selecting De-identification Algorithm .....	9
Figure 5. Configured Quasi-identifiers .....	9
Figure 6. Configuring Sensitive Attributes .....	10
Figure 7. Redaction of Sensitive Rare Values.....	11
Figure 8. Generalization of Sensitive Rare Values .....	12
Figure 9. Replacing Sensitive Rare Values .....	13
Figure 10. Summary of Configured Resources.....	14
Figure 11. Successfully Finished De-identification Process.....	15
Figure 12. Failed De-identification Process.....	15
Figure 13. Successful Validation of Resources .....	16
Figure 14. Failed Validation of Resources .....	17
Figure 15. De-identified Resources in JSON format .....	18
Figure 16. Warning for Restricted Resources .....	18
Figure 17. Restricted Resources in JSON Format.....	19
Figure 18. Save Options for De-identified Data .....	20
Figure 19. Saving De-identified Data as New Data .....	20
Figure 20. Successfully Saved.....	21
Figure 21. Saving the Configuration .....	21
Figure 22. Selecting a Saved Configuration.....	22
Figure 23. Exporting the Configuration.....	23
Figure 24. Importing a Saved Configuration .....	24
Figure 25. Pass Through .....	25
Figure 26. Redaction.....	25
Figure 27. Substitution According to Regular Expression .....	26
Figure 28. Manual Substitution.....	27
Figure 29. Recoverable Substitution.....	28

Figure 30. Fuzzing.....	29
Figure 31. Date Shifting.....	30
Figure 32. Generalization of Integers .....	31
Figure 33. Generalization of Decimals.....	31
Figure 34. Generalization of Dates .....	32
Figure 35. Patient Data & De-identified 4-anonymous Patient Data [5].....	33
Figure 36. De-identified 4-anonymous Patient Data & Diversities of Each Equivalence Class [5] .....	33
Figure 37. Tool Language Settings .....	35
Figure 38. Tool Window View Options .....	36
Figure 39. Opening Sidebar .....	36
Figure 40. Opening Developer Tools .....	37
Figure 41. Help - GitHub Repository .....	38

## 1. Introduction

This manual describes how to use the Data Privacy Tool which is a standalone desktop application that can run on any operating system (Windows, macOS, Linux). The tool has been developed within the scope of FAIR4Health project (<https://www.fair4health.eu/>). It is an open-source project, and accessible in GitHub through <https://github.com/fair4health/data-privacy-tool>. The setup instructions are provided in the README file in the GitHub [repository](#). You can follow the steps in there to create an executable.

## 2. Data Privacy Tool Functional Features and User Manual

This manual consists of three sections:

1. **De-identification Steps Manual** describing the steps of how to de-identify data residing in an HL7 FHIR Repository,
2. **De-identification Methodology** explaining the algorithms and techniques used in the tool in a detailed way,
3. **Tool Settings** such as language and window settings.

Before you start using the application, you can browse the section summarizing the de-identification steps on the screen that opens when you run the application (Figure 1).

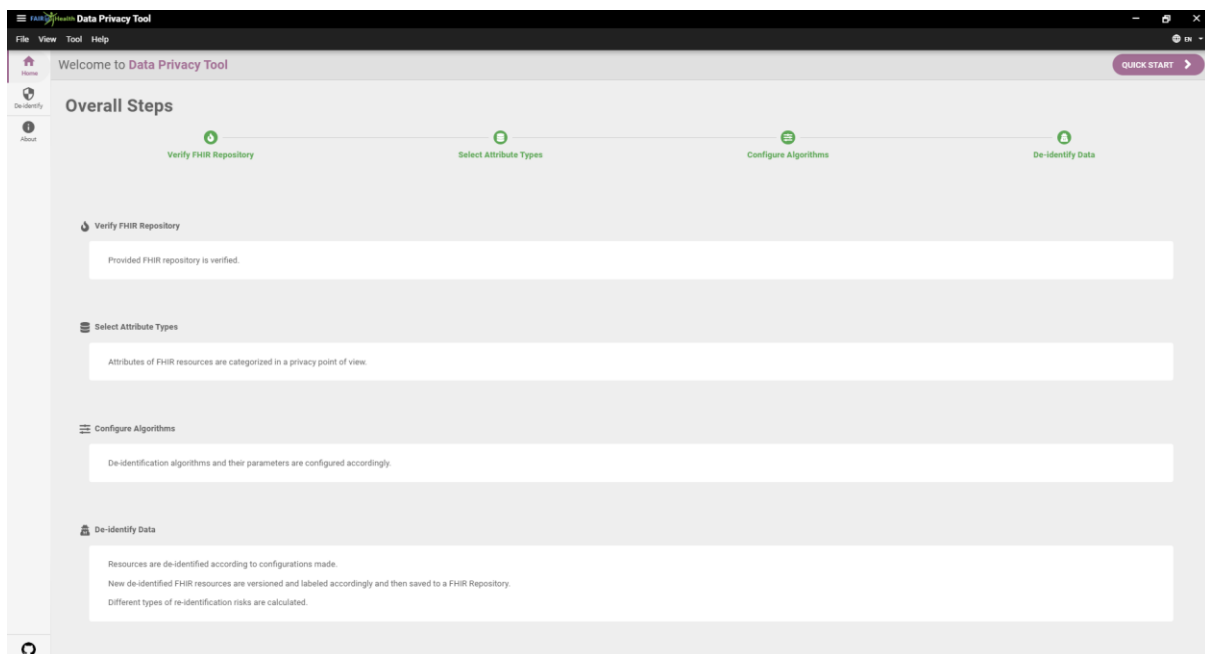


Figure 1. Home Screen

## 2.1. De-identification Steps Manual

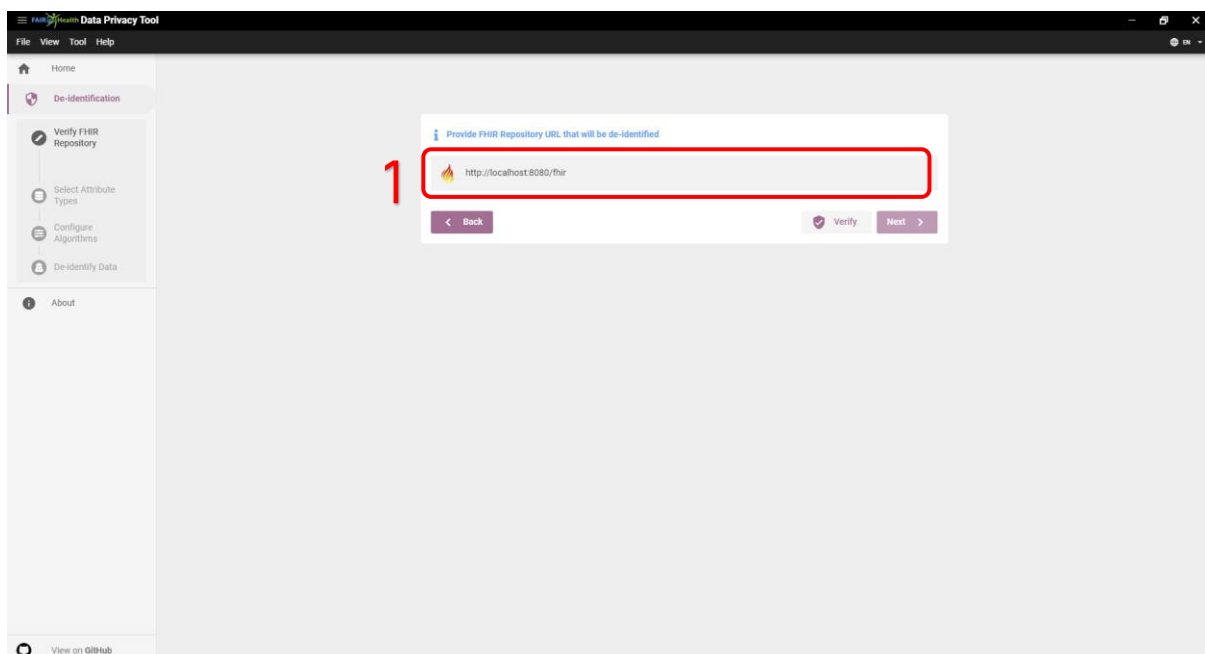
In this section, the steps of data de-identification are described with screenshots. The data de-identification process consists of 4 steps:

- ❖ Verifying FHIR Repository
- ❖ Selecting Attribute Types
- ❖ Configuring De-identification Algorithms
- ❖ De-identifying Data

### 2.1.1. Verifying FHIR Repository

The Data Privacy Tool accesses a FHIR Repository and de-identifies the FHIR resources residing in it. Hence, the very first step is to verify whether the tool can successfully access a FHIR Repository. In this regard, it asks the user to provide a URL, as shown in Figure 2. Enter the FHIR repository **base URL** shown in **1** and click on the verify button.

*In FAIR4Health, <https://onfhir.io> is utilized as the FHIR Repository. Before you continue with the de-identification steps, it is necessary that an onFHIR instance is running in your system or in a location that is accessible by your system.*



**Figure 2.** Verifying FHIR Repository



### 2.1.2. Selecting Attribute Types

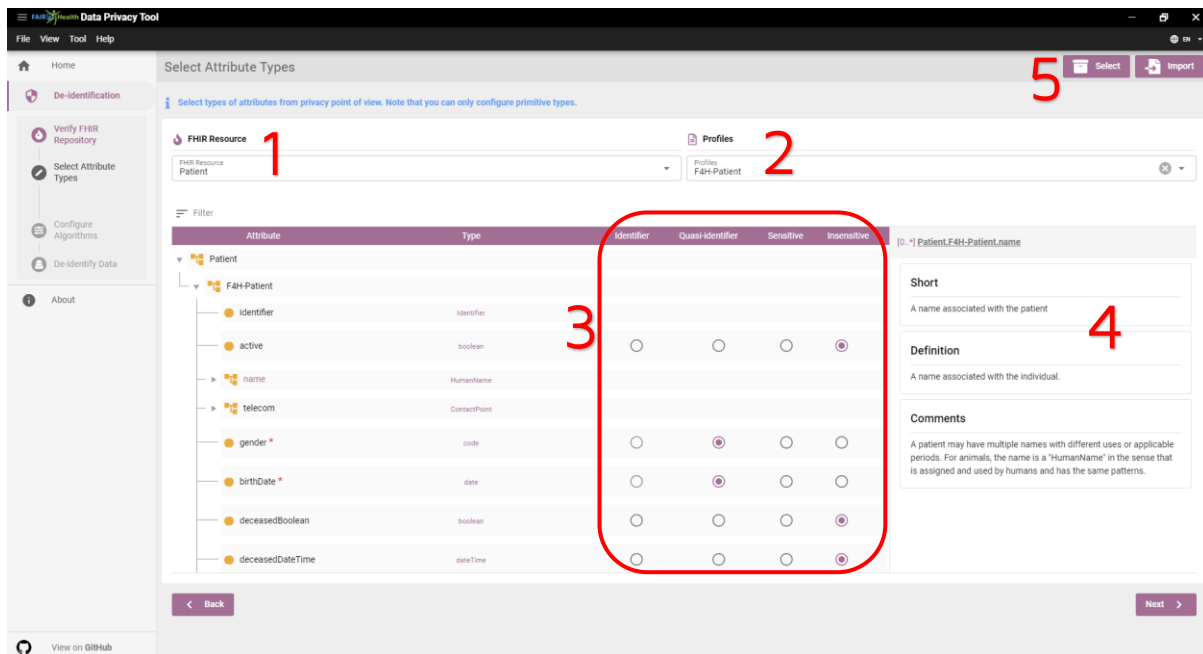
After FHIR repository connection is made, the tool extracts structure information of data residing in the repository and asks user to select attribute types as shown in Figure 3. Select the FHIR resources among the available FHIR resources in the repository for de-identification configuration (1). It is possible to make configurations either for base resources or by selecting a profile from the list of profiles residing in the repository (2). If you want to de-identify more than one resource or one profile, you should configure all of them separately by following 1, 2, and 3 for each.

When the desired resource (and profile) is selected, now it is time to specify types of attributes from the privacy point of view. In 3, you can see a table (extracted from *StructureDefinition*) consisting of attributes of the selected resource/profile in a structured way. You should categorize each attribute by their nature:

- ❖ **Identifier:** Uniquely identifying attributes such as citizen ID, health insurance number etc.
- ❖ **Quasi-identifier:** Attributes that can identify users with a combination of other quasi-identifiers (age, address, etc.)
- ❖ **Sensitive:** Attributes that may have sensitive information of patients such as diagnosis codes
- ❖ **Insensitive:** Attributes that have no critical information

If you wish, you can examine the properties of each attribute by clicking on them, as shown in 4.

Moreover, if there exist any configuration made before, you can simply use it by clicking on the **Select** button shown in 5. You can either select one of the saved configurations in the tool (2.1.4.7) or upload a configuration file which is exported before (2.1.4.8).



**Figure 3.** Selecting Attribute Types

### 2.1.3. Configuring De-identification Algorithms

The de-identification algorithm configurations are made according to the attribute types defined in previous step. **Identifiers** are directly removed; hence they do not require any configuration. **Insensitive attributes** remain as they are, so they do not require any configuration as well. Therefore, in this step, you will make de-identification algorithm configurations for **Quasi-identifiers** and **Sensitive attributes**, which are explained in detailed in section 2.1.3.1 and 2.1.3.2, respectively.

#### 2.1.3.1. Configuring Quasi-identifiers

In Figure 4, you can see the configuration screen for quasi-identifiers. First, select the FHIR resources among the available FHIR resources in the repository (1). In 2, you can see that there are two tabs: one is for quasi-identifiers and the other is for sensitive attributes. In this section, quasi-identifiers are selected (sensitive attributes are explained in the next section: 2.1.3.2).

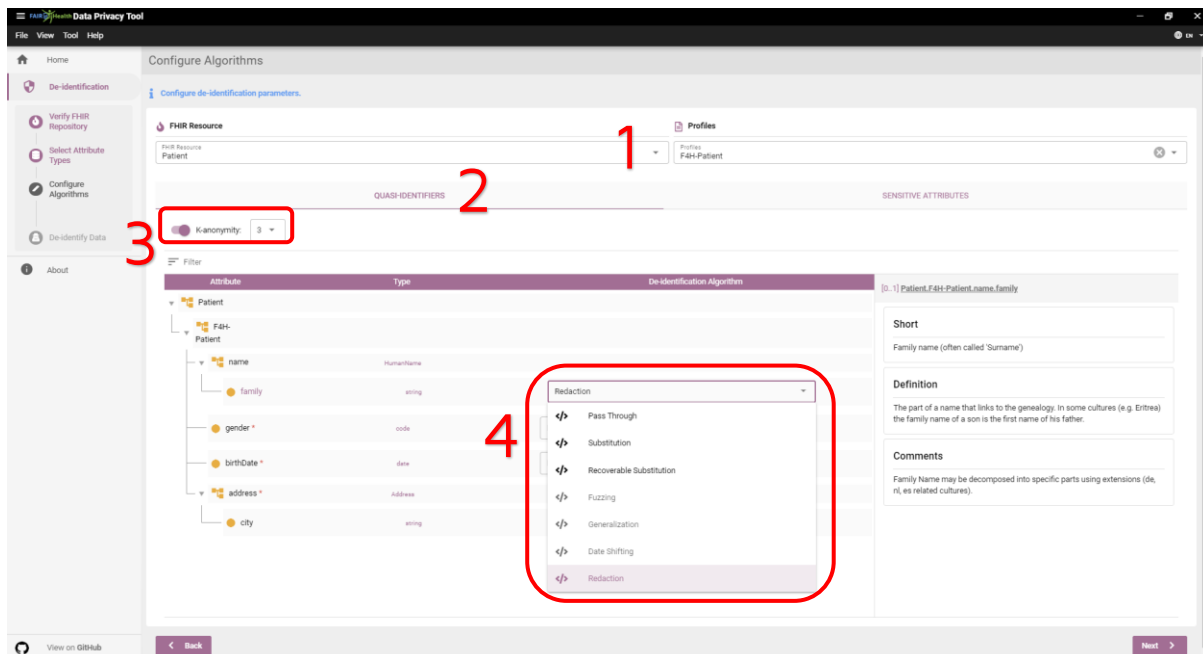
In 3, you can see an option for k-anonymity. k-anonymity is a property of a dataset to describe the dataset's level of anonymity. A dataset is said to be k-anonymous if

combinations of quasi-identifiers occur in at least  $k$  different rows of the dataset. If you like to learn more about  $k$ -anonymity, jump to section 2.2.2.1. In here, you can either enable or disable  $k$ -anonymity. If you enable it, you should make a choice from {2, 3, 4, 5}. If you are not sure which one to select, continue with the recommended value, which is 3.

For each attribute that is specified as a quasi-identifier, you see a dropdown listing the available De-identification Algorithms as shown in [4](#). The tool will de-identify the attribute according the algorithm you selected here. You can see the short descriptions of the algorithms here. You can find the detailed information in section 2.2.1.

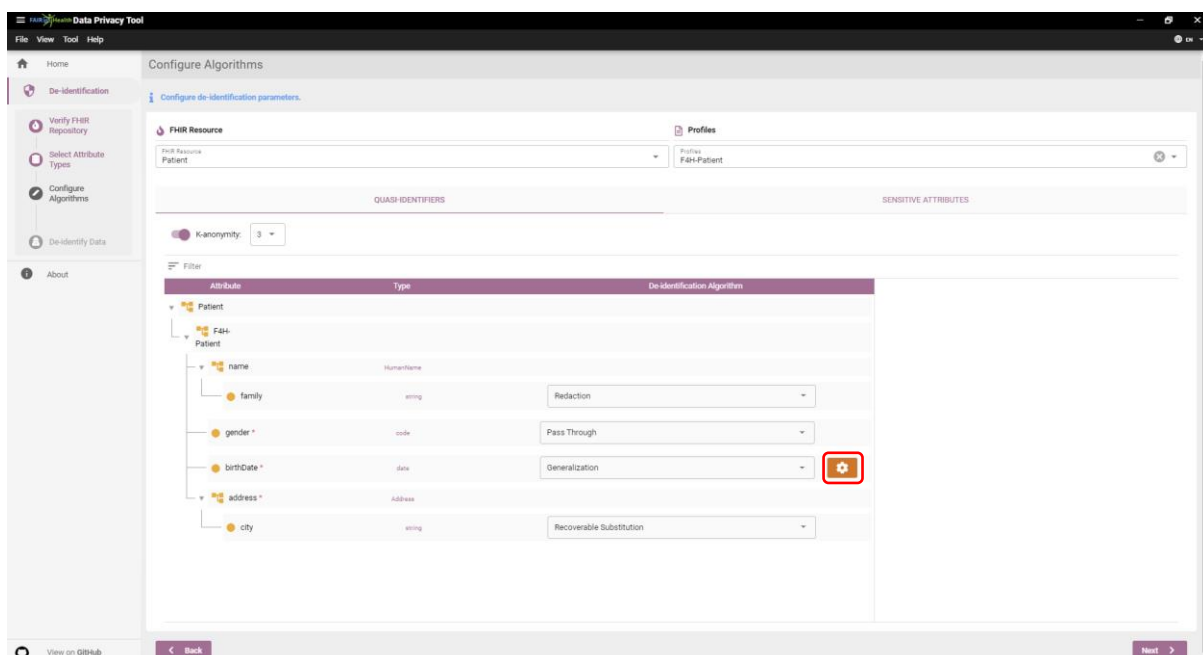
- ❖ **Pass Through**
  - Attribute is saved with no change.
- ❖ **Redaction**
  - Attribute is completely removed.
- ❖ **Substitution**
  - If attribute requires regex, attribute value is replaced with a randomly generated value that fits attribute's regular expression.
  - Otherwise, attribute value is replaced with the substitution character that is provided by the user.
- ❖ **Recoverable Substitution**
  - A new value is generated automatically with a hash function.
- ❖ **Fuzzing**
  - Noise is added to the attribute within the range of the percentage that is provided by the user.
- ❖ **Date Shifting**
  - Date is shifted randomly within a range that is provided by the user.
- ❖ **Generalization**
  - If attribute is integer; last digits of the integer is rounded according to the user's choice.
  - If attribute is float; decimal places of the floating number is rounded according to the user's choice.
  - If attribute is date; only the information of the date unit that is provided by the user is kept.
- ❖ **Replace**
  - Sensitive rare values are replaced with new values provided by the user.

As an example, in [4](#), you can see **De-identification Algorithm** options for the *family* attribute. As you can see, some algorithms are disabled in the screen. It is because not all the algorithms can be applied to every attribute type because of their natures. In this example, the *family* attribute is a *string*, hence the algorithms that are not applicable to *string* type are disabled, e.g. date shifting.



**Figure 4.** Selecting De-identification Algorithm

Some algorithms require additional parameters to be provided. In such cases, a button appear on the right side as you can see in Figure 5. More information about algorithm parameters are explained in detailed in section 2.2.1.



**Figure 5.** Configured Quasi-identifiers

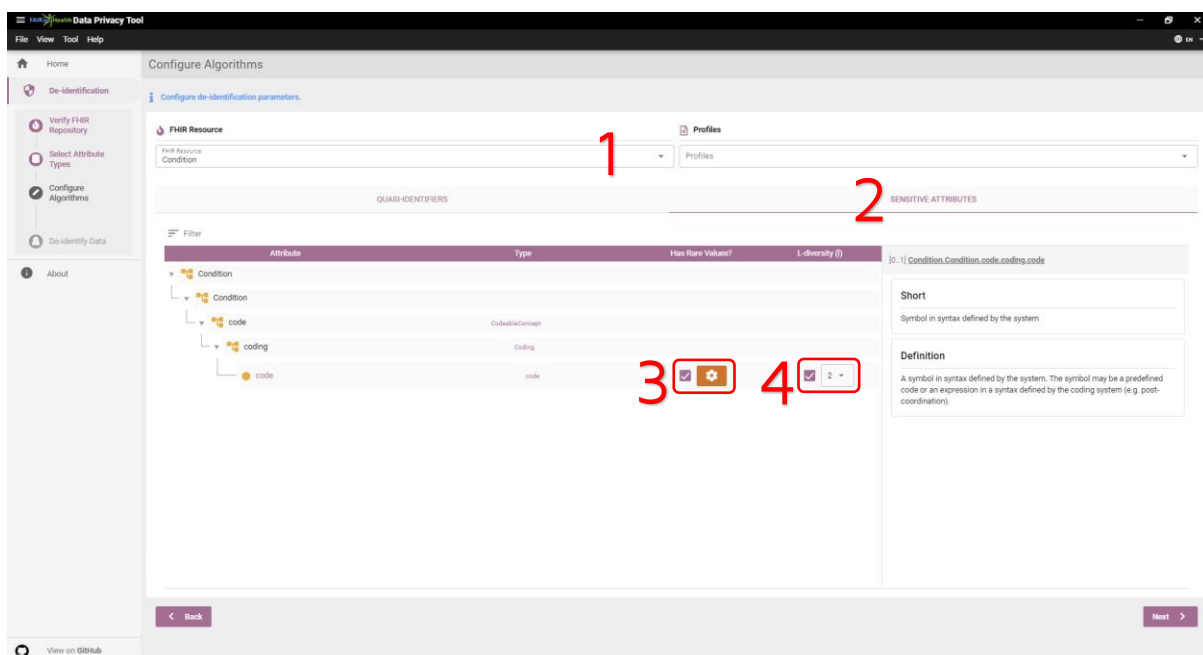
### 2.1.3.2. Configuring Sensitive Attributes

After quasi-identifiers are configured, select the next tab **(2)** for configuring sensitive attributes. Figure 6 shows the configuration screen for sensitive attributes. Here, only the attributes that have stated as sensitive attributes are shown. In **3**, you can see the option for indicating rare values. If the attribute has any rare values, you should select this option and make its configurations by clicking on the button near to it. More information about rare sensitive values are explained in the next section (2.1.3.3).

In **4**, you can see the option for l-diversity. L-diversity is a technique describing that sensitive attributes would have at most same frequency as L. You can either enable or disable l-diversity. If you enable it, you should a choice from {2, 3, 4, 5}. Two important points that you should now about l-diversity are:

- ❖ l-diversity cannot be applied if the selected resource does not have k-anonymity for its quasi-identifiers,
- ❖ l-diversity value cannot be more than k-anonymity value.

More information about l-diversity are provided in section 2.2.2.2.

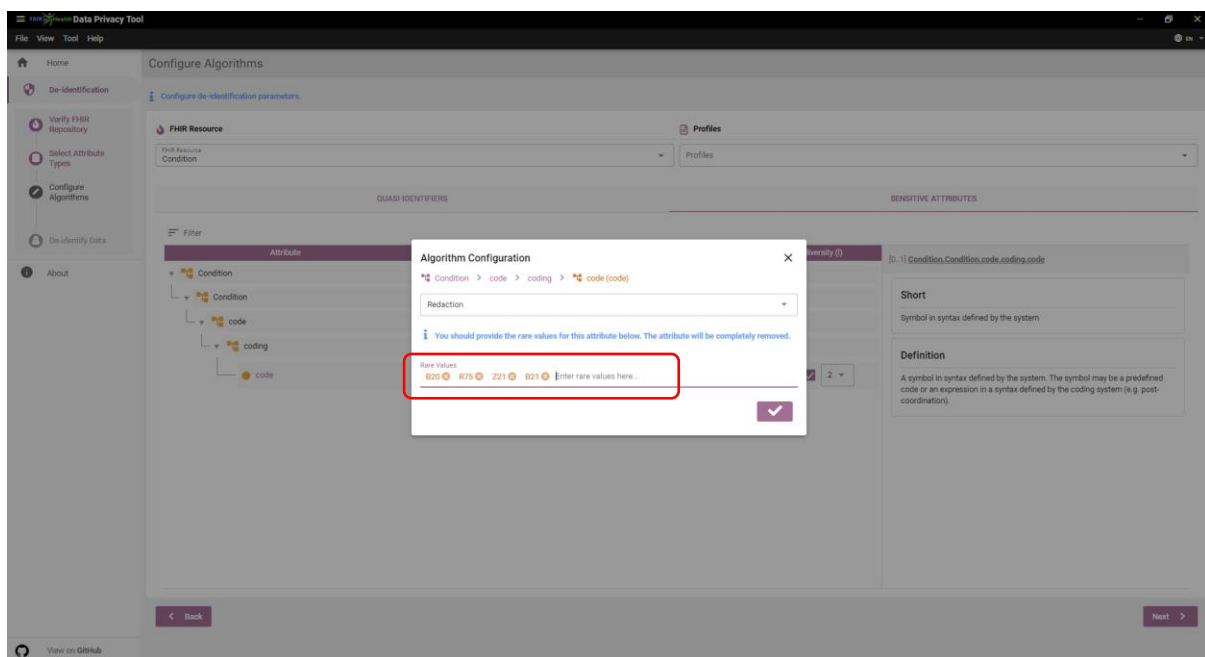


**Figure 6.** Configuring Sensitive Attributes

### 2.1.3.3. Handling Sensitive Rare Values

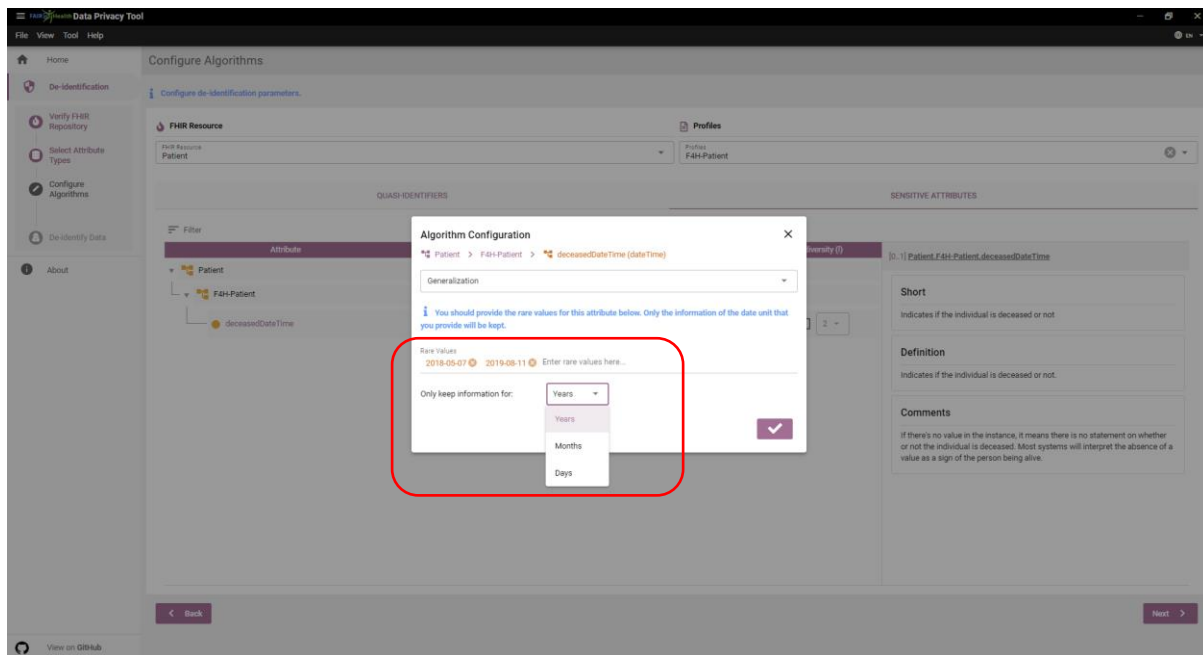
Sensitive attributes may have some rare values. They should be indicated because they may cause the identification of patients easily. For example, if a patient has a disease that is encountered one in a million, and this information is kept in the database as it is, it can be used for identifying the patient. For this reason, possible rare values should be selected and a de-identification technique such as Redaction, Replace or Substitution should be configured for them.

In Figure 7, you can see a rare value handling for *Condition.code.coding.code* attribute. In this example, **Redaction** algorithm is selected for handling this attribute, and rare values, which represent HIV according to different ICD10 coding levels, are provided as an input. During the performance of de-identification, sensitive attributes having these rare values will be completely removed.



**Figure 7.** Redaction of Sensitive Rare Values

Figure 8 shows another example of handling rare value for *Patient.deceasedDateTime* attribute of the *F4H-Patient* profile. In this example, **Generalization** algorithm is selected, and rare values are provided. Since this is a date field, the tool enables to configure the algorithm further, i.e. keeping the date information at year, month, or day level.

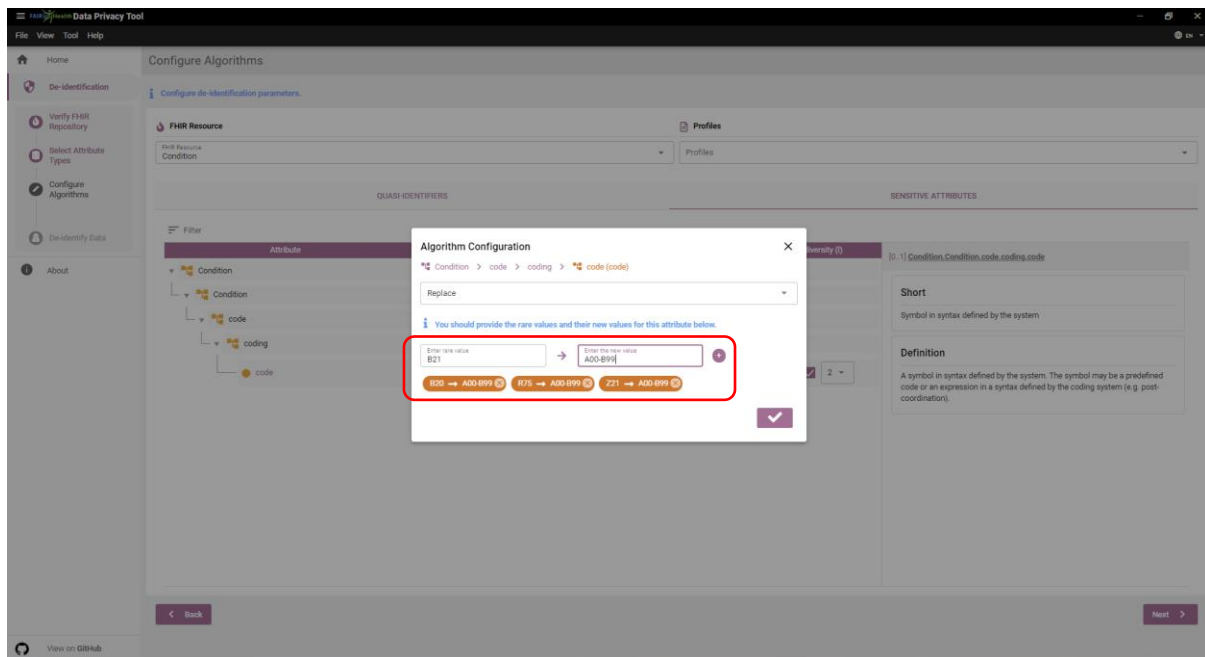


**Figure 8.** Generalization of Sensitive Rare Values

In Figure 9, you can see the rare value handling for *Condition.code.coding.code* attribute with **Replace** algorithm. In this algorithm, you provide which rare value will be replaced by which value as an input. The values in the example represent HIV according to different ICD10 coding levels. Their classification can be observed as follows:

- ❖ **(A00-B99)** Certain infectious and parasitic diseases
  - **(B20-B24)** Human immunodeficiency virus [HIV] disease
    - **(B20)** Human immunodeficiency virus [HIV] disease resulting in infectious and parasitic diseases
    - **(B21)** Human immunodeficiency virus [HIV] disease resulting in malignant neoplasms
  - **(R75)** Inconclusive laboratory evidence of human immunodeficiency virus [HIV]
  - **(Z21)** Asymptomatic human immunodeficiency virus [HIV] infection status

More details about Redaction, Replace and Substitution algorithms are provided in the corresponding subsection in section 2.2.1.



**Figure 9.** Replacing Sensitive Rare Values

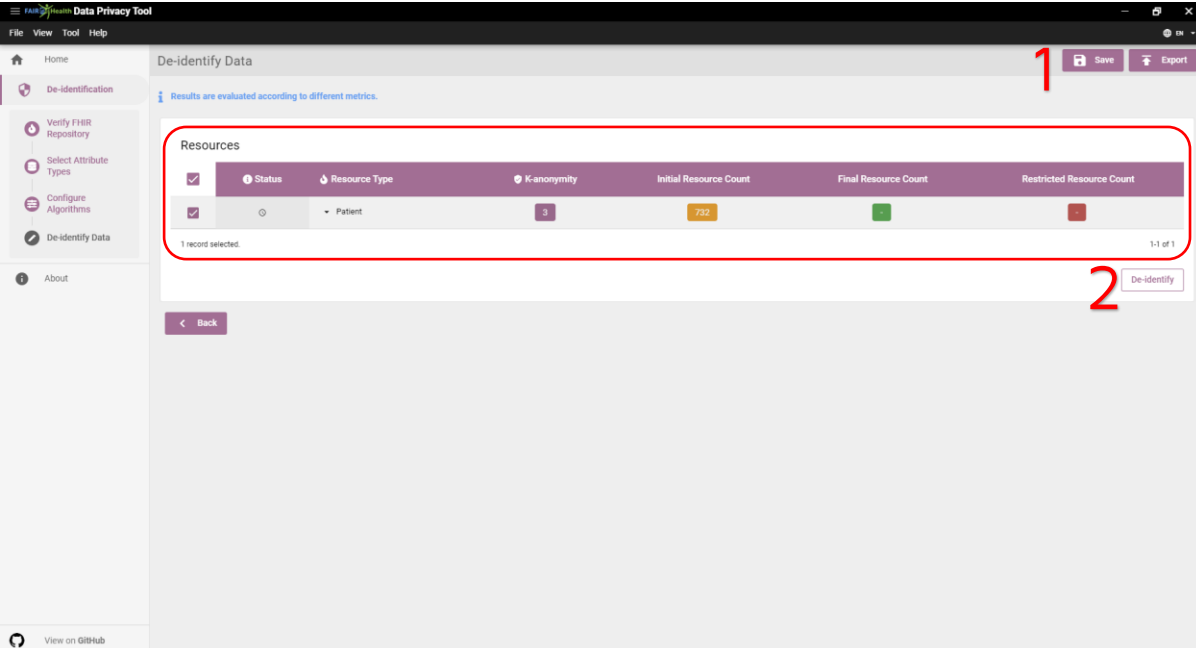
## 2.1.4. De-identifying Data

### 2.1.4.1. Summary of Configurations

After configurations are made in the previous steps, they are first summarized to the user at this step. Figure 10 shows the corresponding screen. In **1**, you can either save the configurations you made to the tool (section 2.1.4.7) or export them to a configuration file (section 2.1.4.8).

Below **1**, there is a table listing the resources configured for de-identification in previous steps. In this example, there exists only Patient resource since we only configured it. As can be seen, its status is **pending**, **k-anonymity** value is 3, and the **initial resource count** is 732. Final resource count and restricted resource count are not provided since the de-identification is not performed yet. You can select the resources you want to de-identify by using the checkboxes on left, and you can start the execution of de-identification by clicking on the “De-identify” button indicated by **2**.





De-identify Data

Results are evaluated according to different metrics.

Status	Resource Type	K-anonymity	Initial Resource Count	Final Resource Count	Restricted Resource Count
✓	Patient	3	732	732	732

1 record selected. 1-1 of 1

Back

Save Export De-identify

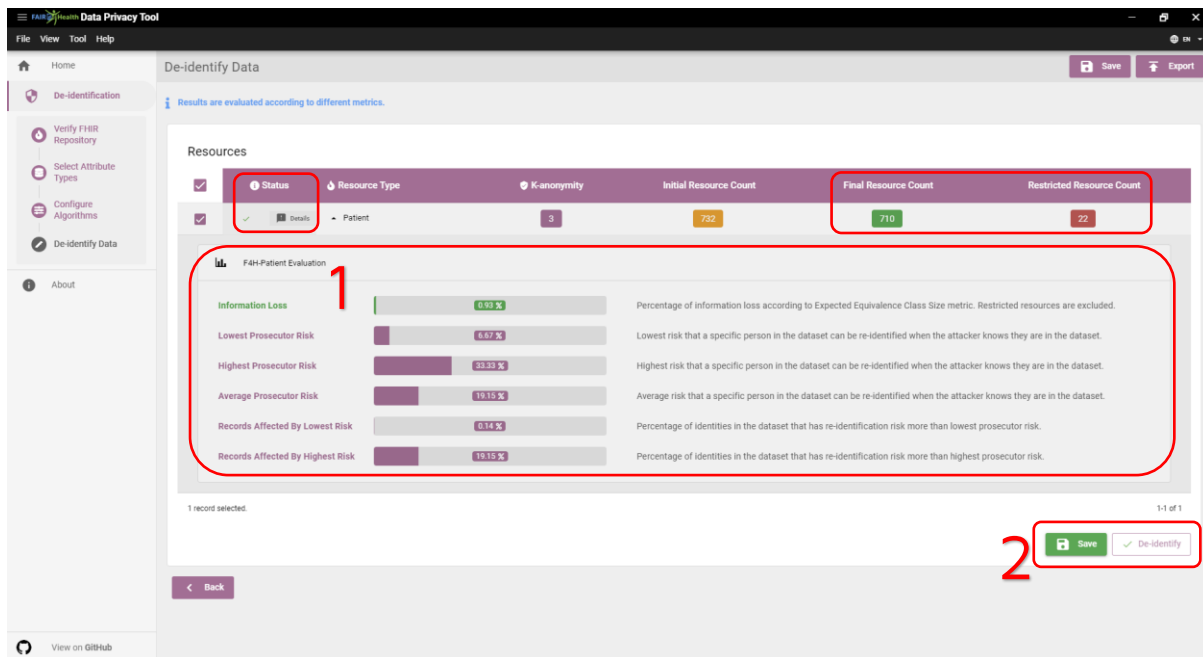
**Figure 10.** Summary of Configured Resources

#### 2.1.4.2. Viewing De-identification Details

After de-identification is completed successfully, the view presented in Figure 11 is shown to the user. As you can see, the **Status** is changed to “Completed”, and the **Final Resource Count** and **Restricted Resource Count** are updated accordingly. By clicking on the number, you can examine their details, which are explained in section 2.1.4.4 and 2.1.4.5, respectively.

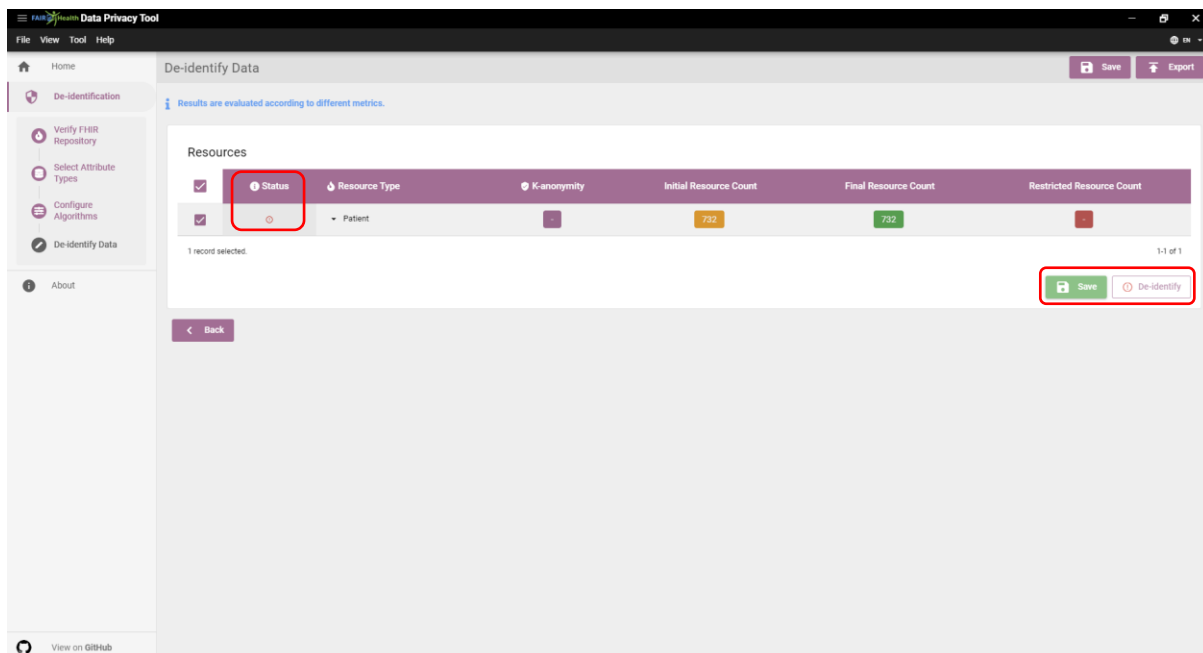
In **1**, you can see the evaluation of the de-identified resources in terms of information loss and privacy risks. These metrics are explained in detail in section 2.2.3.

In **2**, there exists a **Save** button that was not shown before the de-identification process is completed. You can select the resources you want to save from the select options on the left and save them to a new repository or the current repository. Details of saving de-identified resources are explained in section 2.1.4.6.



**Figure 11.** Successfully Finished De-identification Process

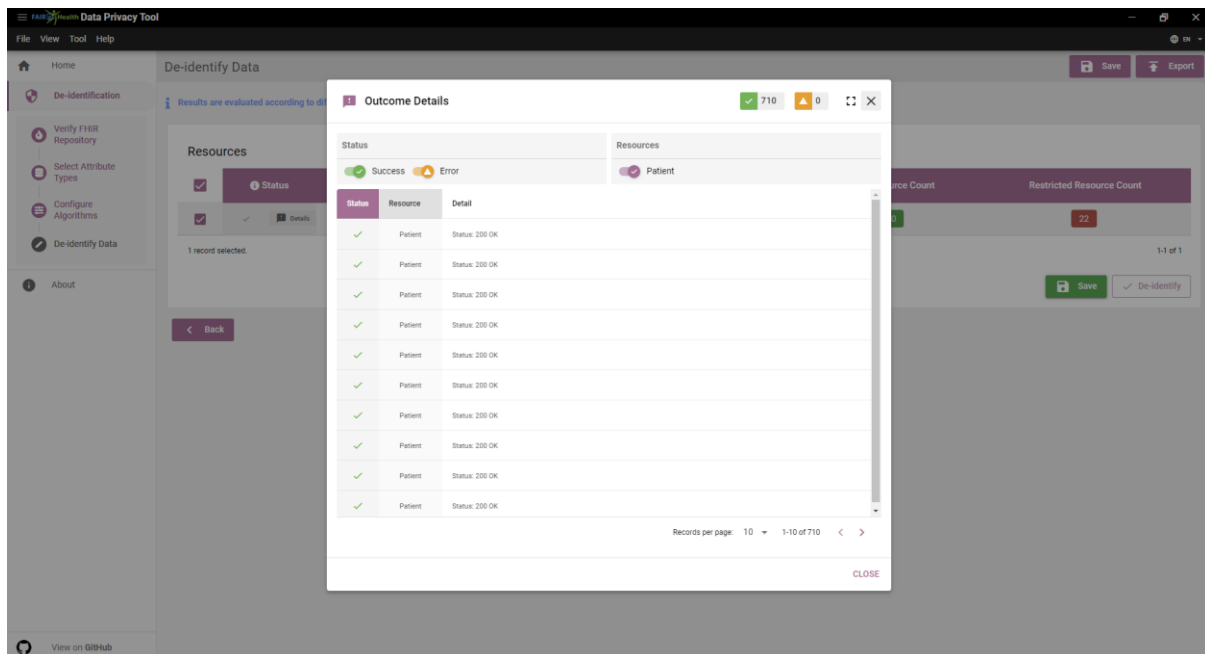
Figure 12 shows an example of failed de-identification. In this example, validation of Patient resources is failed. Therefore, they cannot be saved into any repository. Validation errors can be examined by clicking on the error icon in the **status** section.



**Figure 12.** Failed De-identification Process

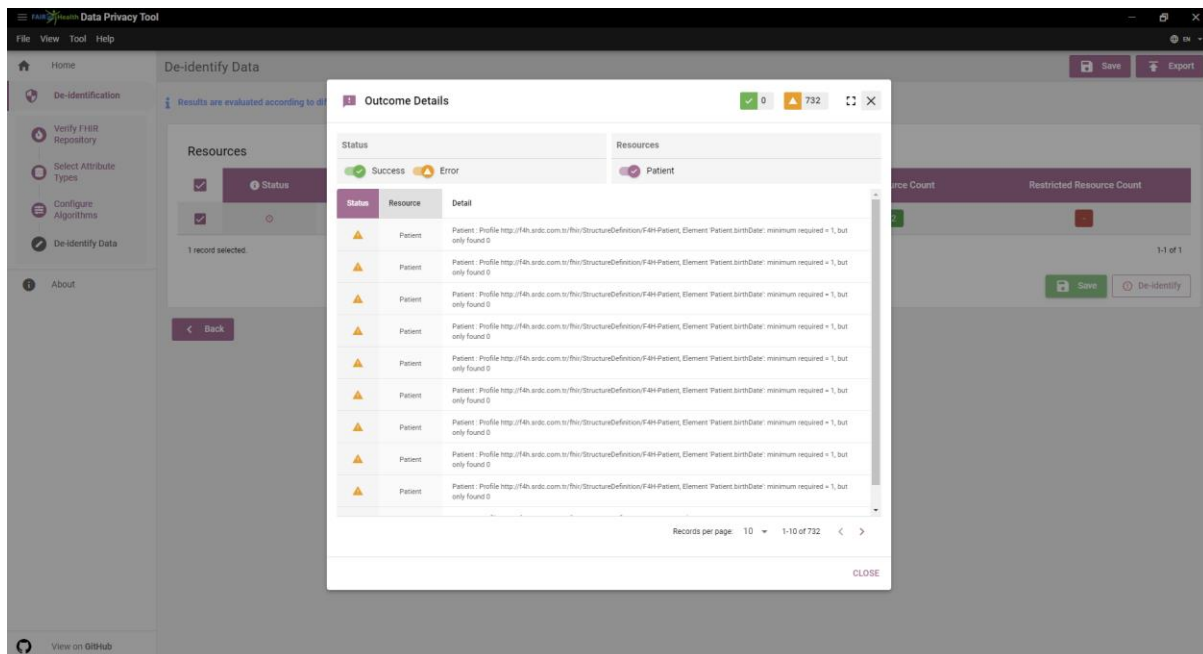
### 2.1.4.3. Viewing Validation Details

In Figure 13, you can see a successful validation of all Patient resources, where each resource returned the **"Status: 200 OK"** response.



**Figure 13.** Successful Validation of Resources

In Figure 14, you can see a failed validation of Patient resources. In this example, each resource returned **"Patient: Profile Element 'Patient.birthDate': minimum required = 1, but only found 0"** error which states that de-identified resources violate their *StructureDefinition*. It can be fixed by going back to previous steps and changing the configurations accordingly.

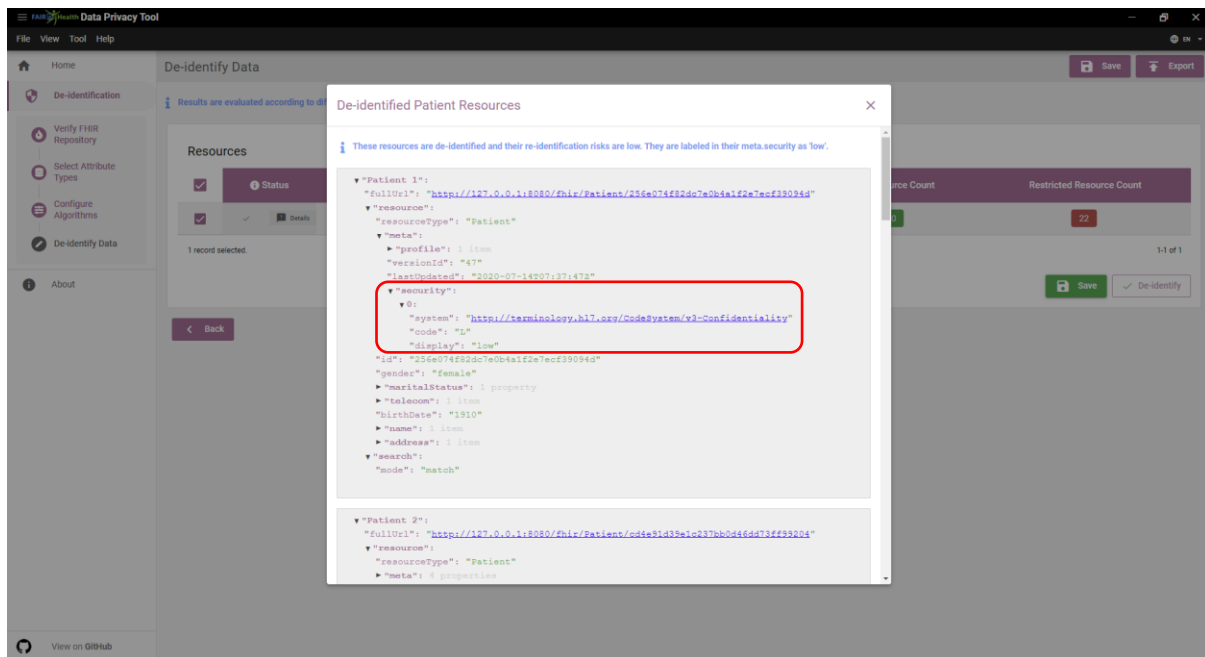


**Figure 14.** Failed Validation of Resources

#### 2.1.4.4. De-identified Resources

De-identified resources can be viewed in JSON format as shown in Figure 15. Attributes can be extended in this JSON viewer. At the bottom, there is a pagination where you can change the pages if you want.

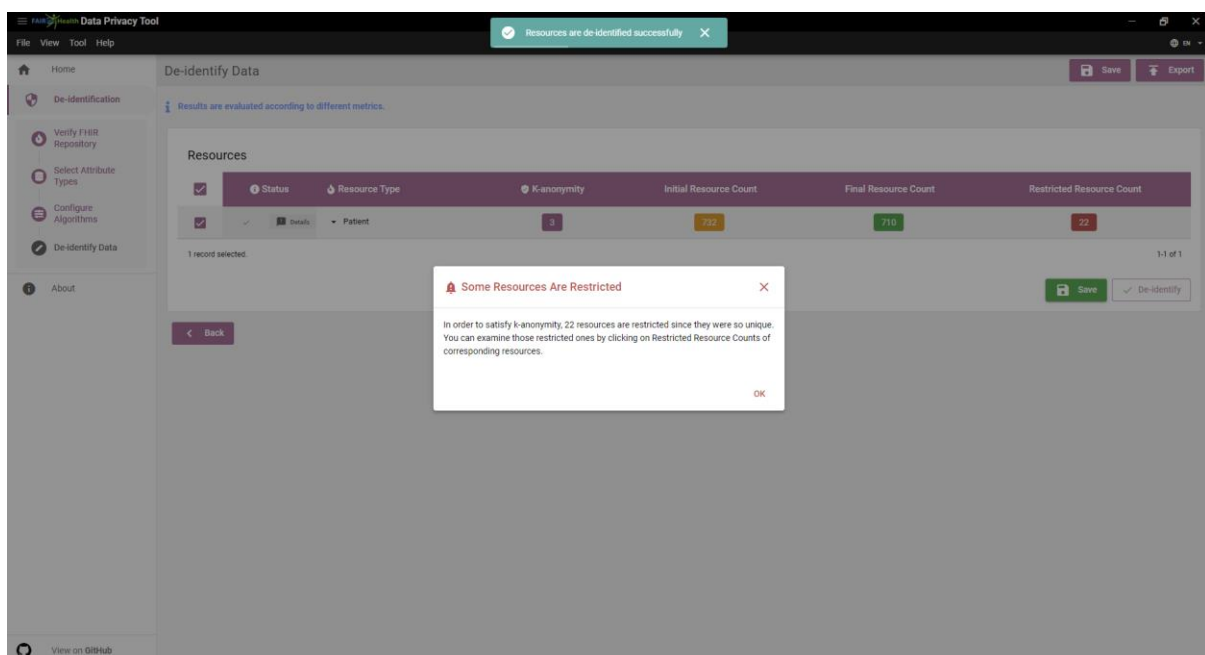
Moreover, security information of all de-identified resources is updated in their *resource.meta.security* attribute. In this situation, they are labeled as 'low' [1], indicating that they have low privacy risks.



**Figure 15.** De-identified Resources in JSON format

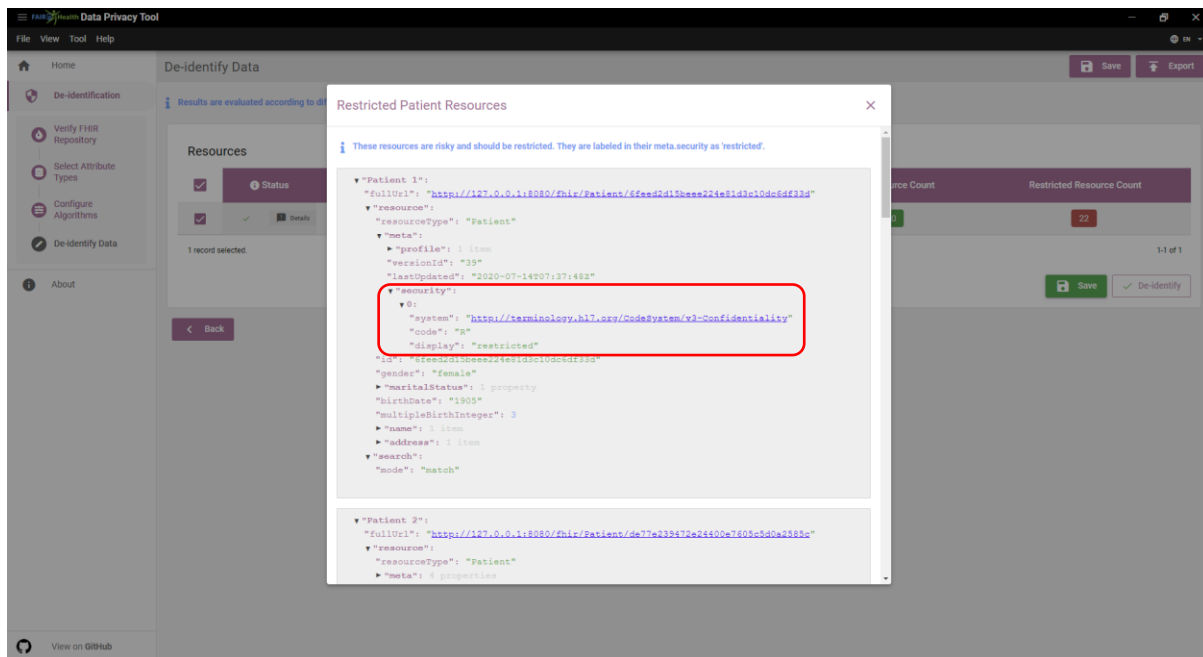
#### 2.1.4.5. Restricted Resources

Some resources may be restricted to satisfy the privacy criteria if they have unique attributes. In Figure 16, you can see a warning message for such restricted resources.



**Figure 16.** Warning for Restricted Resources

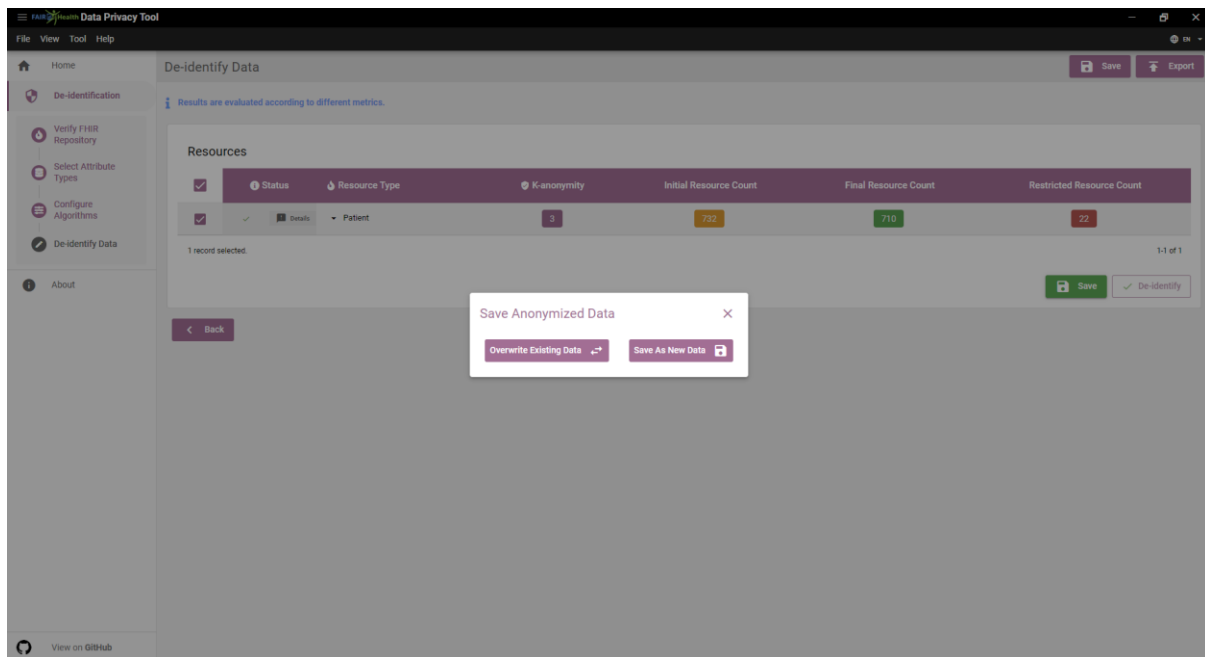
Restricted resources can also be viewed in JSON format as well, as shown in Figure 17. Security information of all restricted resources is updated in their *resource.meta.security* attribute. In this situation, they are labeled as 'restricted' [1], indicating that they have high privacy risks, and they should be restricted for usage.



**Figure 17.** Restricted Resources in JSON Format

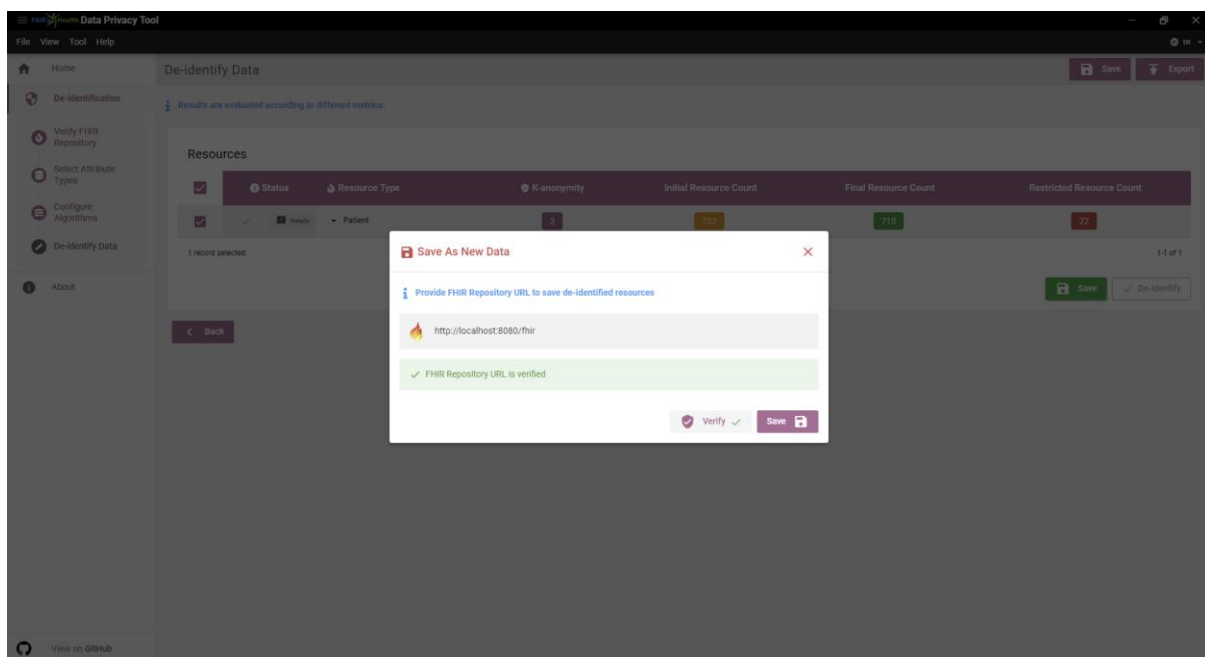
#### 2.1.4.6. Saving De-identified Data

After de-identification is completed, valid resources can be saved to a new repository or to the current repository. In Figure 18, you can see a pop-up asking user to select one of the options. If you select overwriting existing data, the resources residing in the current repository will be updated and versioned accordingly. When the saving is successful, the message shown in Figure 20 prompts.

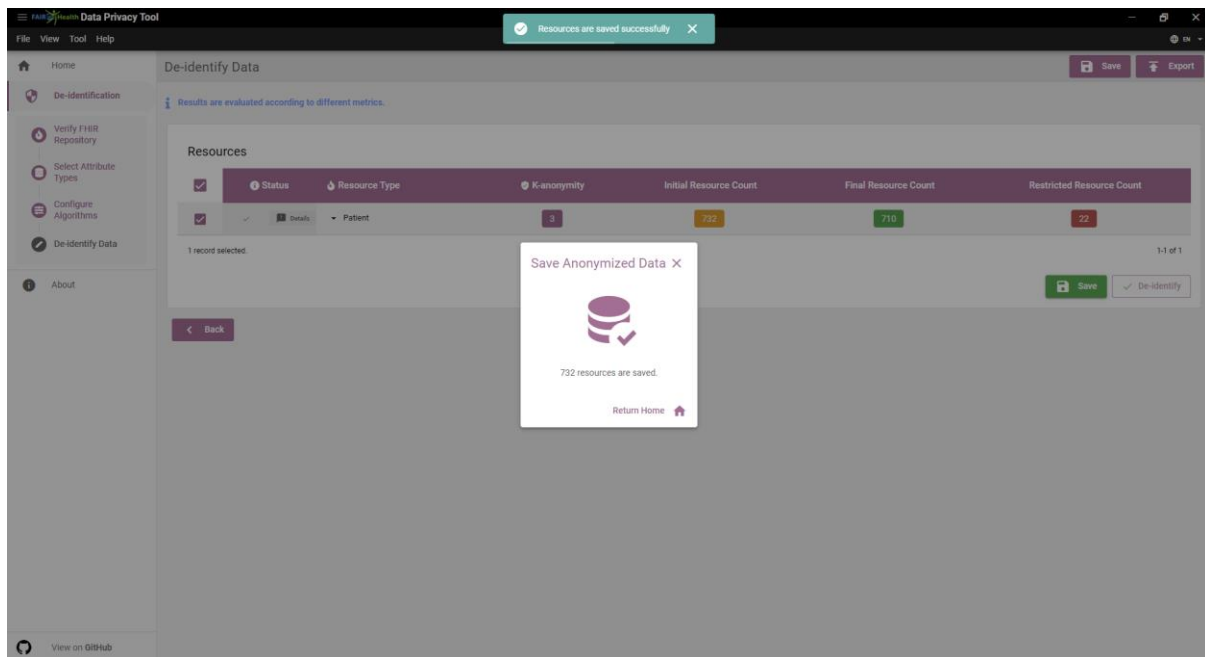


**Figure 18.** Save Options for De-identified Data

If you select saving data as new data to another repository, the pop-up shown in Figure 19 appears. You should provide the URL of the FHIR repository to which you want to save de-identified data. When you write the URL, you must verify it first. After it is verified, you can save your data to the new repository. When the saving is successfully finished, the message shown in Figure 20 prompts.



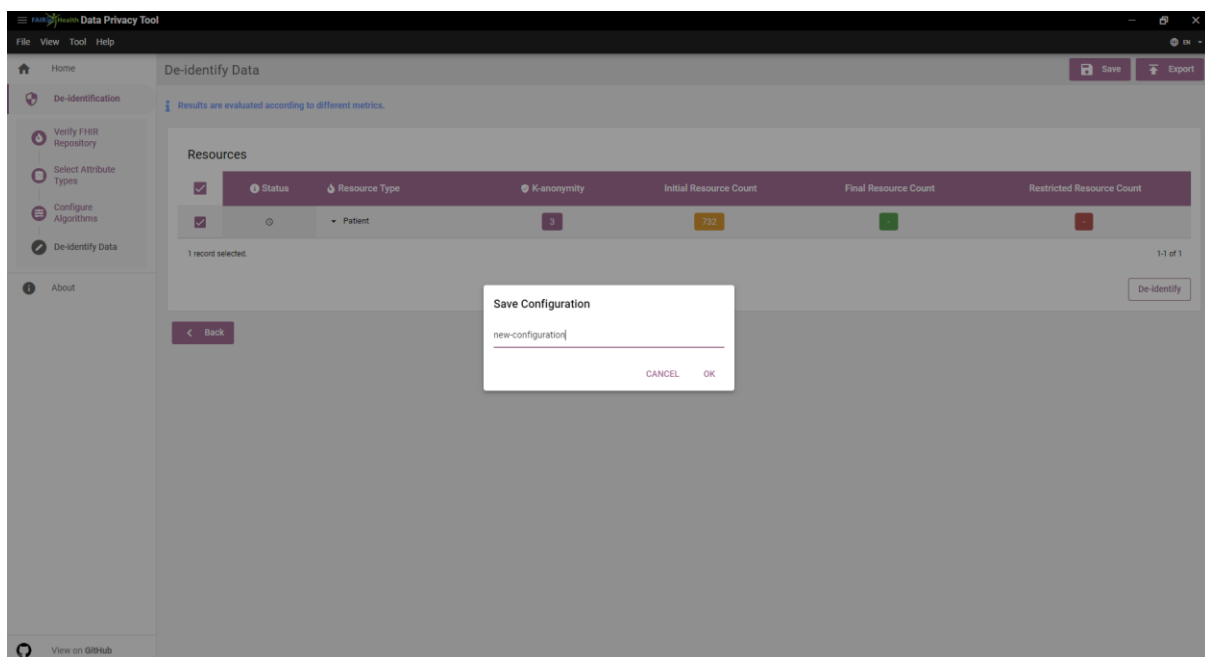
**Figure 19.** Saving De-identified Data as New Data



**Figure 20.** Successfully Saved

#### 2.1.4.7. Saving the Configuration and Use it Later

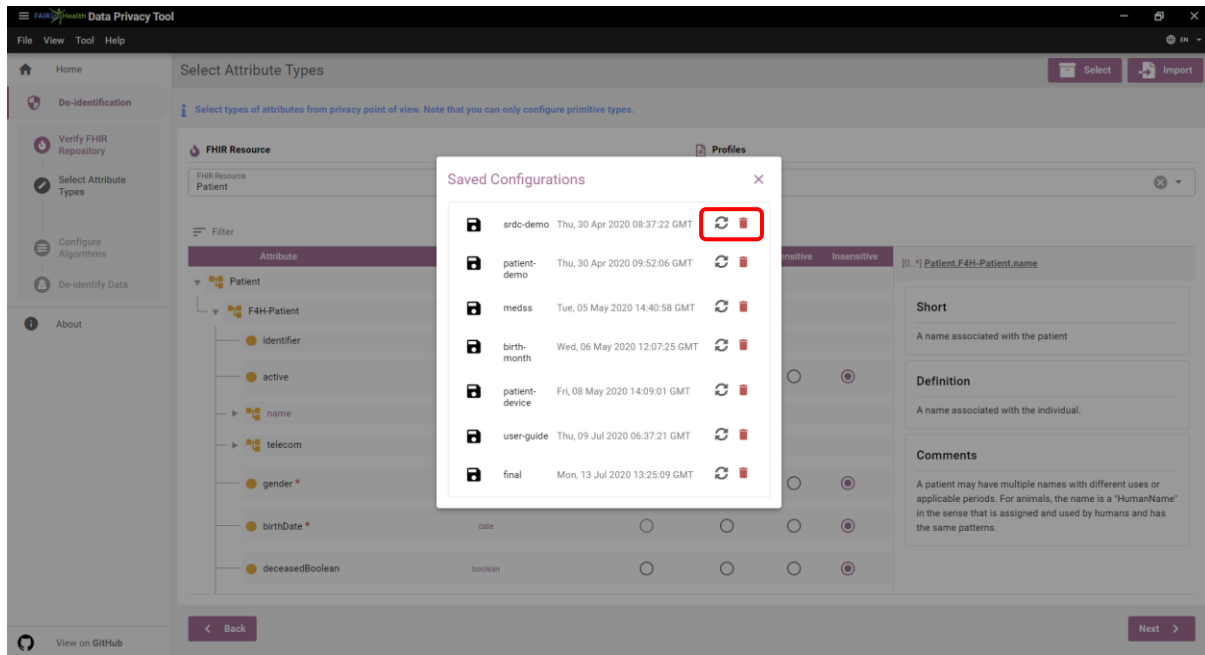
Inside the tool, you can save the configurations you made in the previous steps for making use of them later in the future. As shown in Figure 21, you can save the configurations by just providing a name.



**Figure 21.** Saving the Configuration



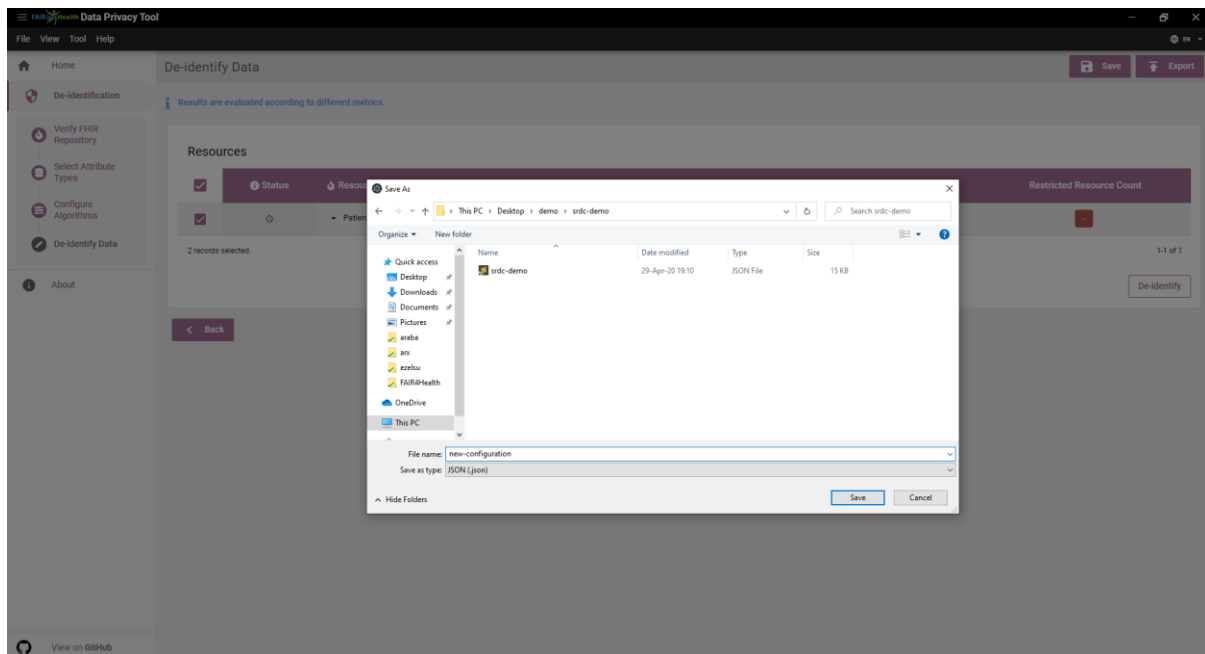
If you have saved any configuration before, you can select and load it by clicking on the **Select** button shown in **5** in Figure 3. If you wish, you can change the parameters of the selected configuration in the next steps and save or export it as a new configuration. In Figure 22, you can see the saved configurations as a list. If you click on the **load** button of any configuration, it will be applied to the current configuration environment. You can also delete a saved configuration by clicking on the **delete** button on right.



**Figure 22.** Selecting a Saved Configuration

#### 2.1.4.8. Exporting the Configuration and Import it Later

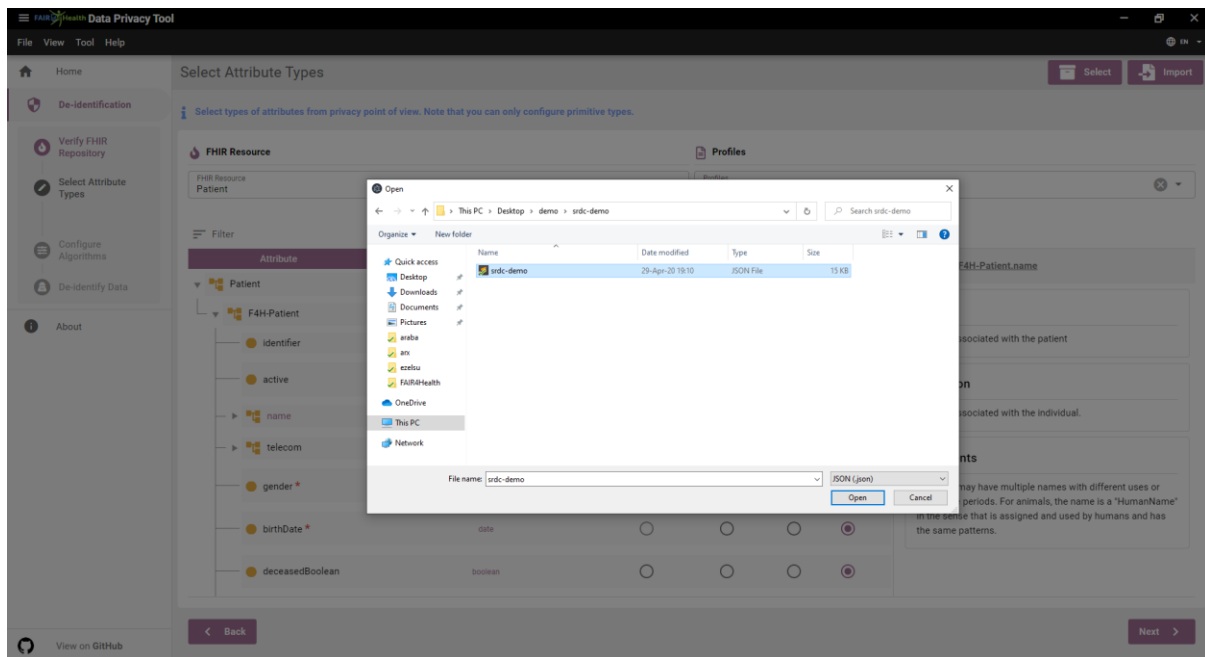
You can also export the configurations you made in the previous steps to a file. In this case, as you can see in Figure 23, you just need to provide a name to the JSON file.



**Figure 23.** Exporting the Configuration

If you want to import a configuration file you have already exported, you can do it by clicking on the **Import** button shown in **5** in Figure 3.

In Figure 24, you can see a file picker pop-up that accepts JSON files. If you select a valid JSON file that you have exported from the tool before, its configurations will be applied to the current environment. If you wish, you can change the parameters of the selected configuration in the next steps and save or export it as a new configuration.



**Figure 24.** Importing a Saved Configuration

## 2.2. De-identification Methodology

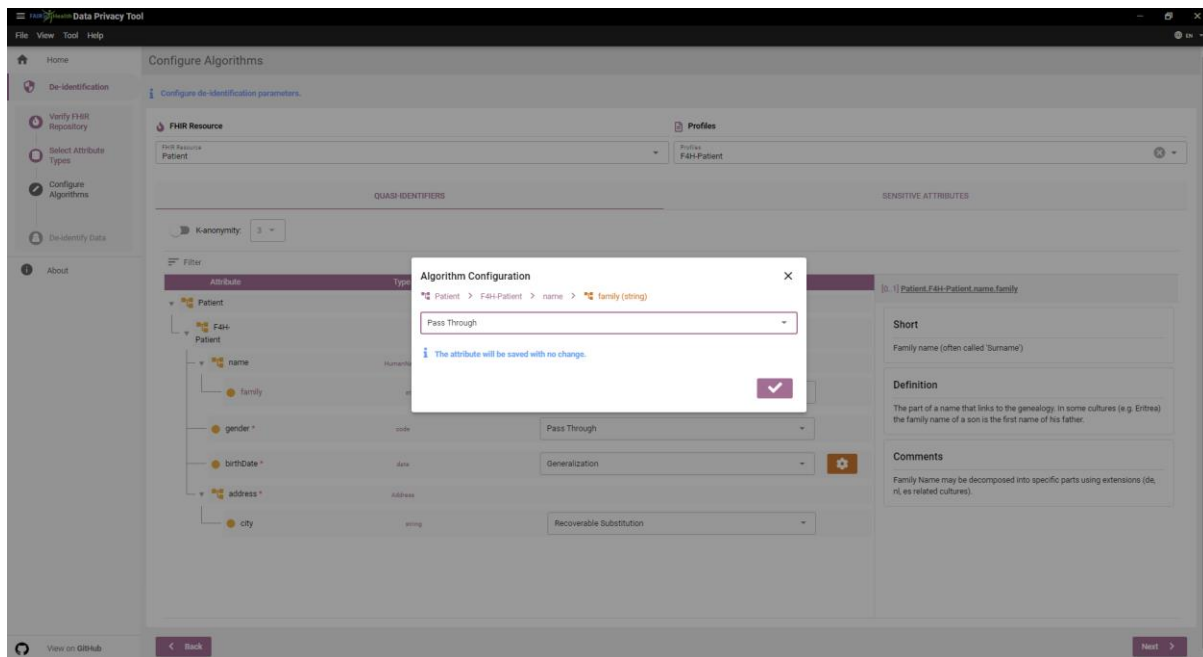
In this section, the de-identification methodologies adopted by the Data Privacy Tool are explained in three chapters:

- ❖ De-identification Algorithms
- ❖ Privacy Criteria
- ❖ Information Loss and Privacy Risks

### 2.2.1. De-identification Algorithms

#### 2.2.1.1. Pass Through

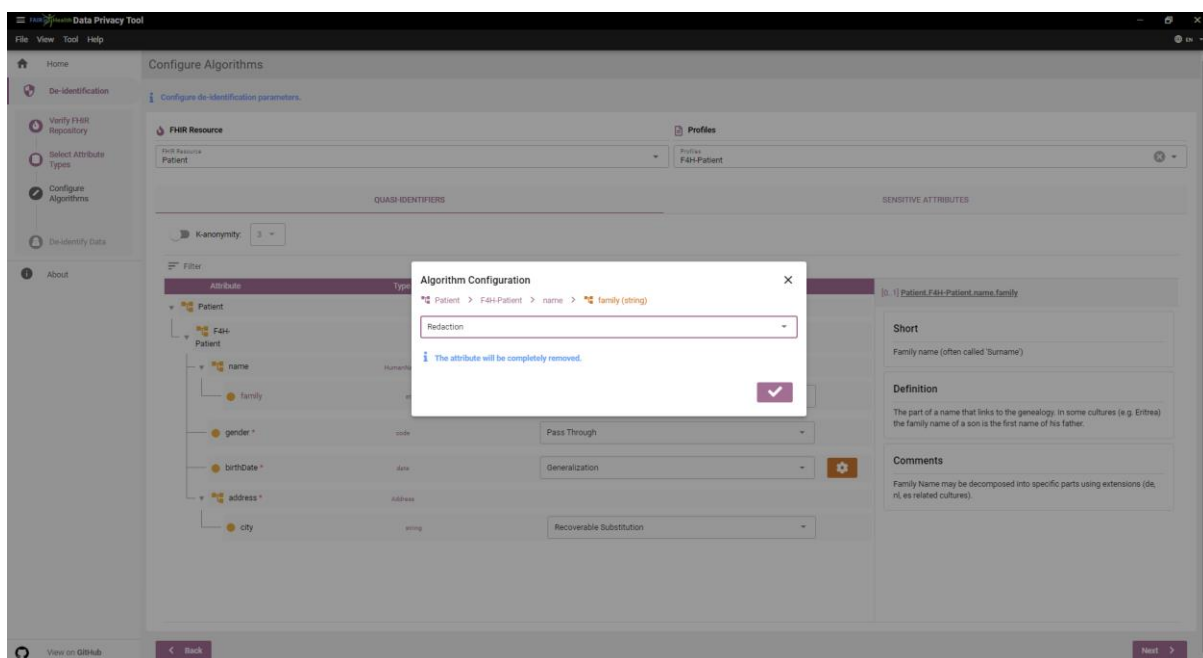
The attribute is saved with no change. No parameter configuration is needed, as can be seen in Figure 25. This algorithm can be applied to every type of attribute.



**Figure 25.** Pass Through

### 2.2.1.2. Redaction

The attribute is completely removed. No parameter configuration is needed, as can be seen in Figure 26. This algorithm can be applied to every type of attribute.

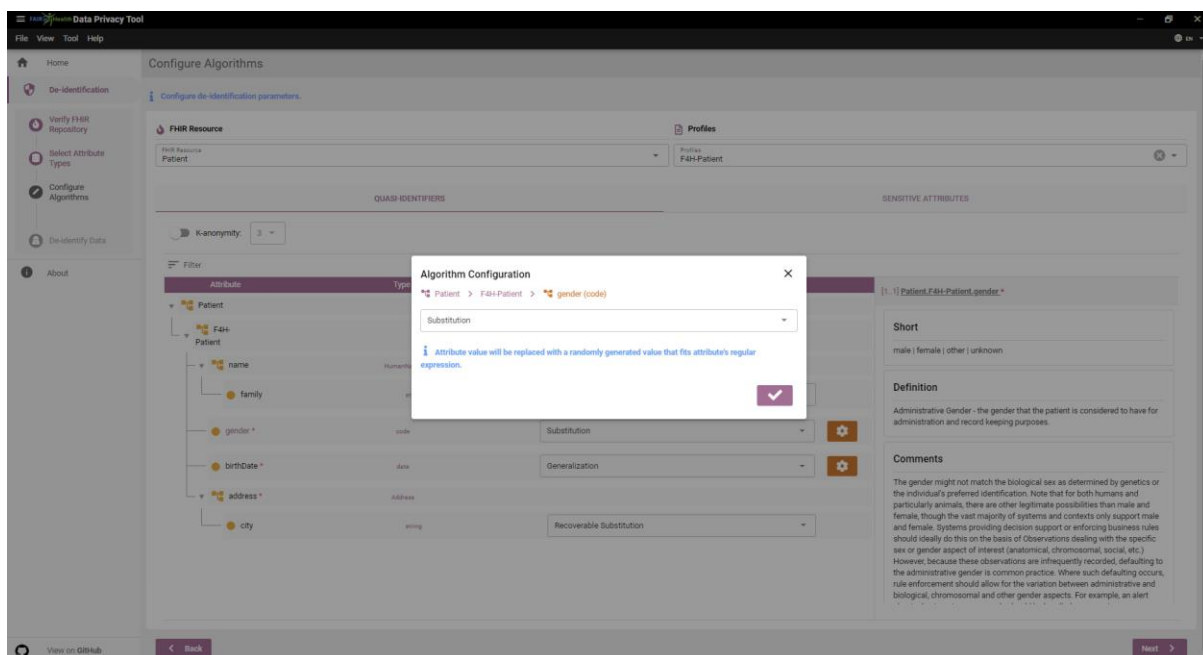


**Figure 26.** Redaction

### 2.2.1.3. Substitution

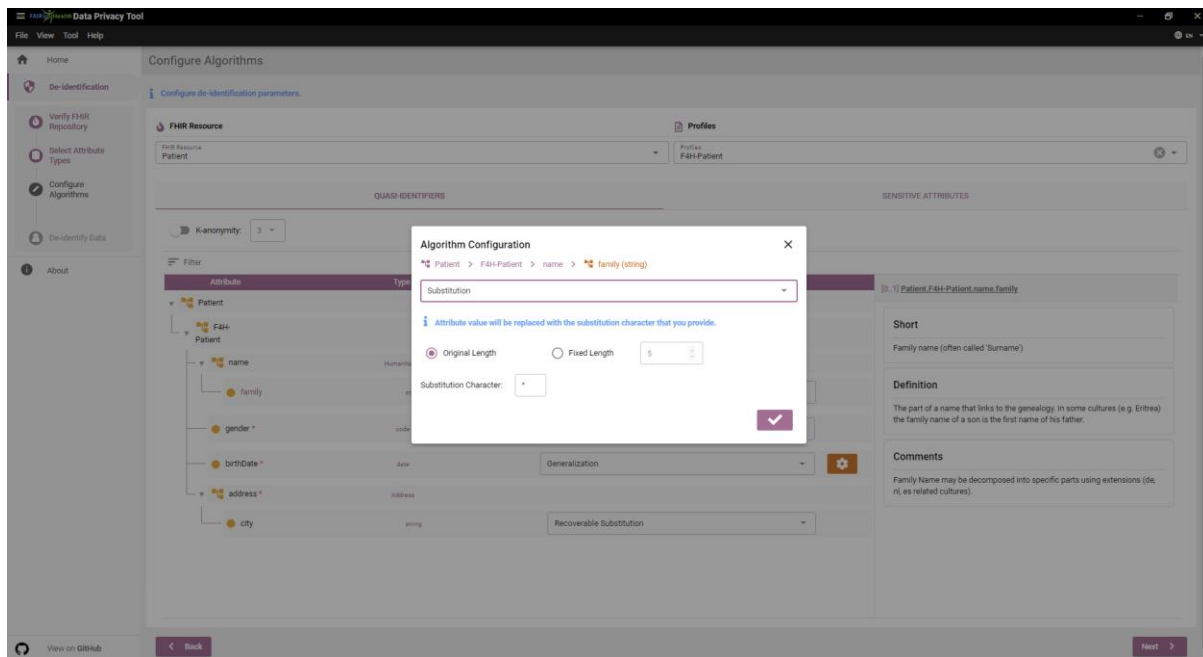
The attribute is substituted with a new value. If attribute requires regex<sup>1</sup>, the attribute value is replaced with a randomly generated value that fits the attribute's regular expression, as shown in Figure 27. Otherwise, the attribute value is replaced with the substitution character with the parameters that the user provides. In Figure 28, you can see a substitution example with length and character parameters. This algorithm can be applied to the following primitive types in FHIR [2]:

- ❖ string
- ❖ uri
- ❖ url
- ❖ canonical
- ❖ base64Binary
- ❖ code
- ❖ oid
- ❖ id
- ❖ markdown
- ❖ uuid



**Figure 27.** Substitution According to Regular Expression

<sup>1</sup> A regular expression (shortened as regex) is a sequence of characters that define a search pattern.

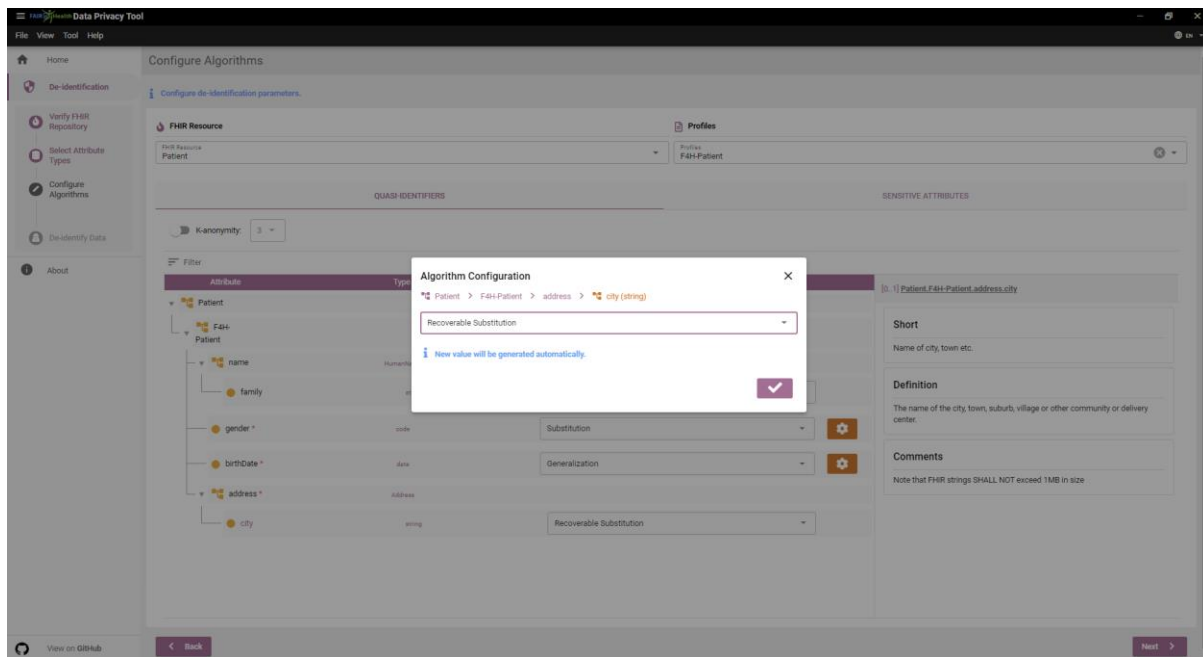


**Figure 28.** Manual Substitution

#### 2.2.1.4. Recoverable Substitution

A new value is generated automatically with a hash function by using the current value of the attribute, and the current value is replaced with this new value. No parameter configuration is needed, as can be seen in Figure 29. This algorithm can be applied to the following primitive types in FHIR [2]:

- ❖ string
- ❖ uri
- ❖ url
- ❖ canonical
- ❖ uuid

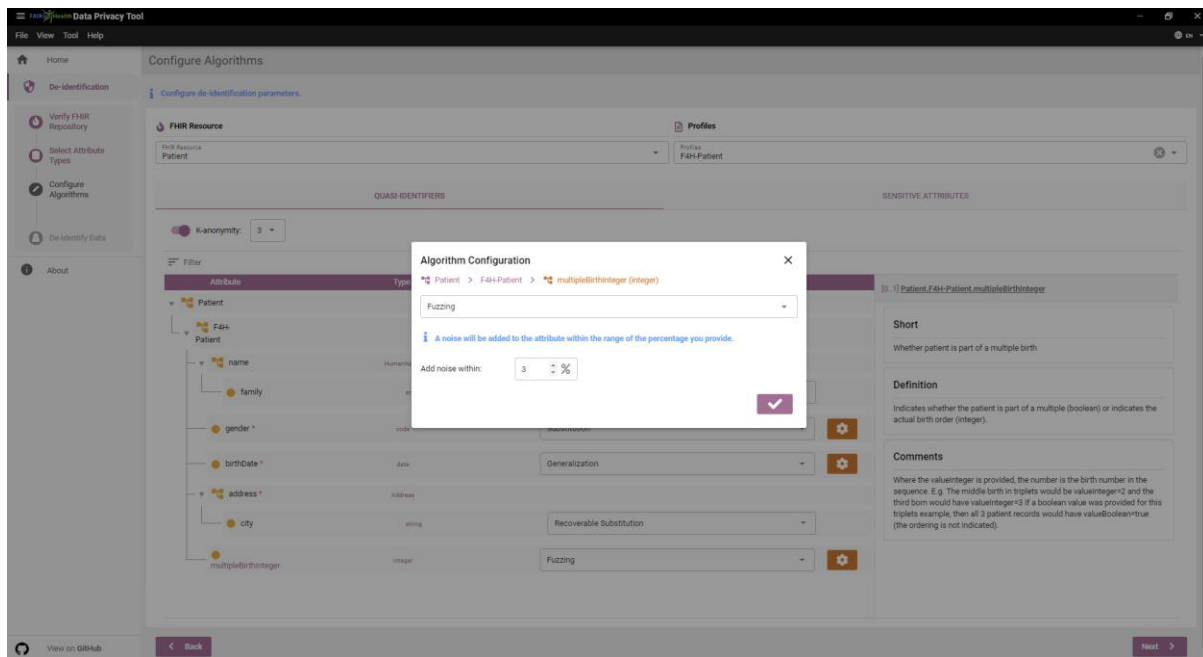


**Figure 29.** Recoverable Substitution

#### 2.2.1.5. Fuzzing

Noise is added to the attribute within the range of the percentage user provides, as can be seen in Figure 30. This algorithm can be applied to the following primitive types in FHIR [2]:

- ❖ integer
- ❖ decimal
- ❖ unsignedInt
- ❖ positiveInt



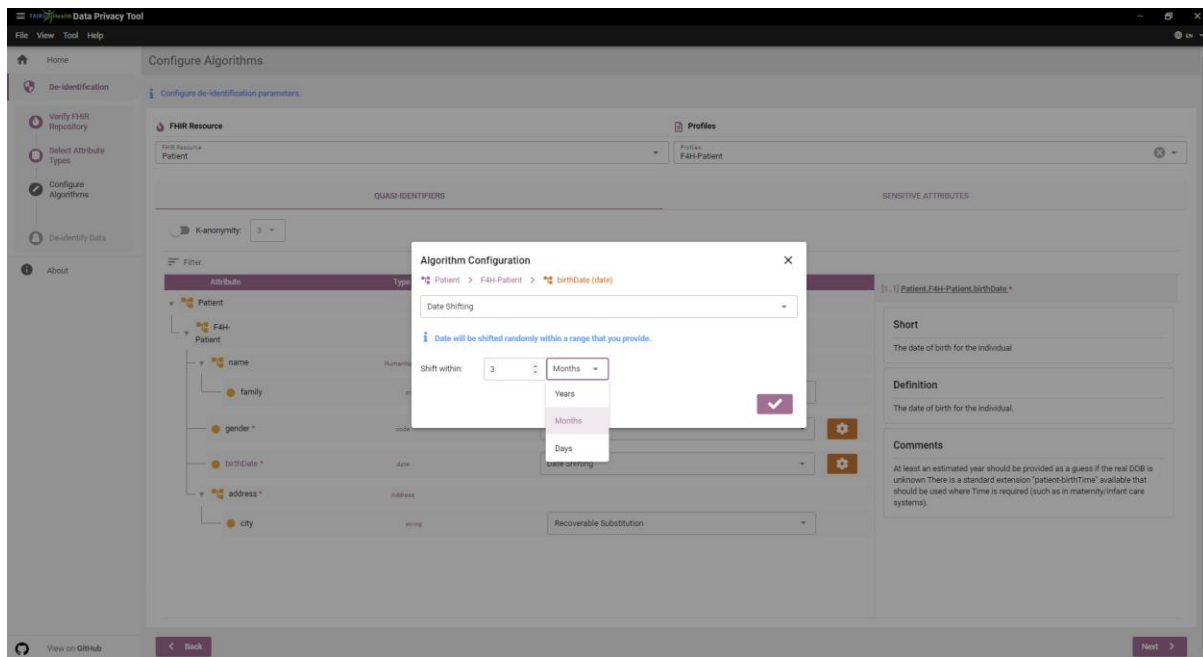
**Figure 30.** Fuzzing

#### 2.2.1.6. Date Shifting

The date is shifted randomly within a range that is provided by the user. As can be seen in Figure 31, the user can provide a range of years, months, or days. This algorithm can be applied to the following primitive types in FHIR [2]:

- ❖ instant
- ❖ date
- ❖ dateTime
- ❖ time





**Figure 31.** Date Shifting

#### 2.2.1.7. Generalization

The attribute is generalized according to its type. In Figure 32, you can see the Generalization of integers, where the last digits of the integer are rounded by the user's choice. Similarly, in Figure 33, you can see the Generalization of decimals, where decimal places of the floating number are rounded by the user's choice.

In Figure 34, you can see the Generalization of dates. This time, date information is generalized according to the information on the date unit that is provided by the user. This algorithm can be applied to the following primitive types in FHIR [2]:

- ❖ integer
- ❖ decimal
- ❖ instant
- ❖ date
- ❖ dateTime
- ❖ time
- ❖ unsignedInt
- ❖ positiveInt

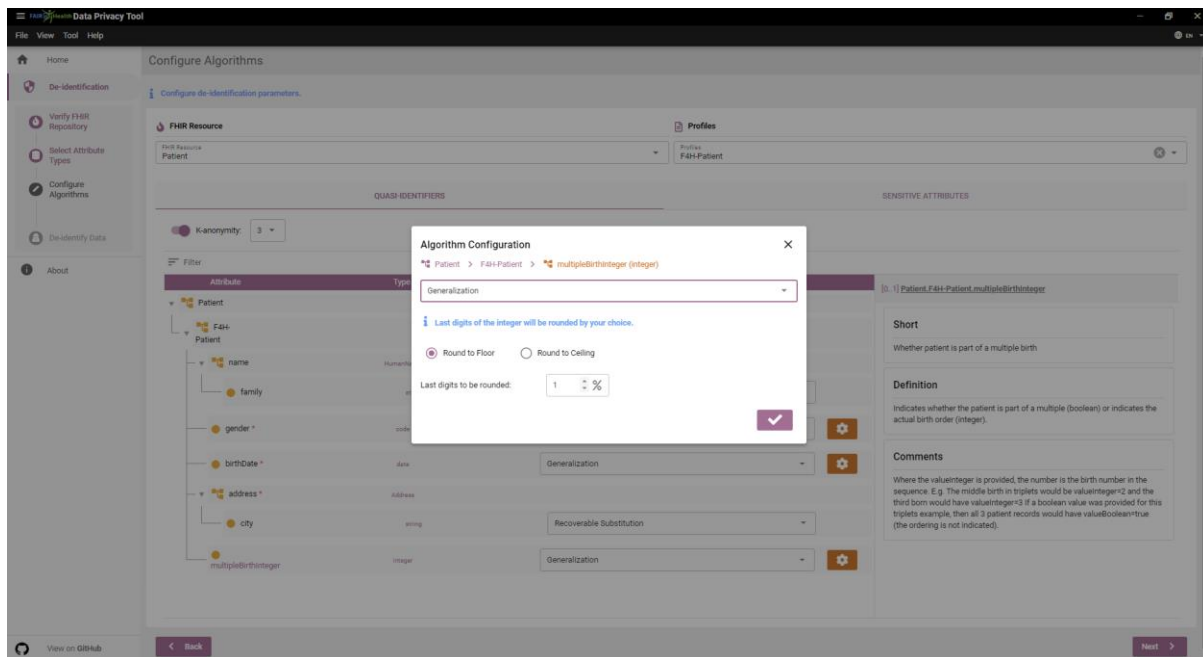


Figure 32. Generalization of Integers

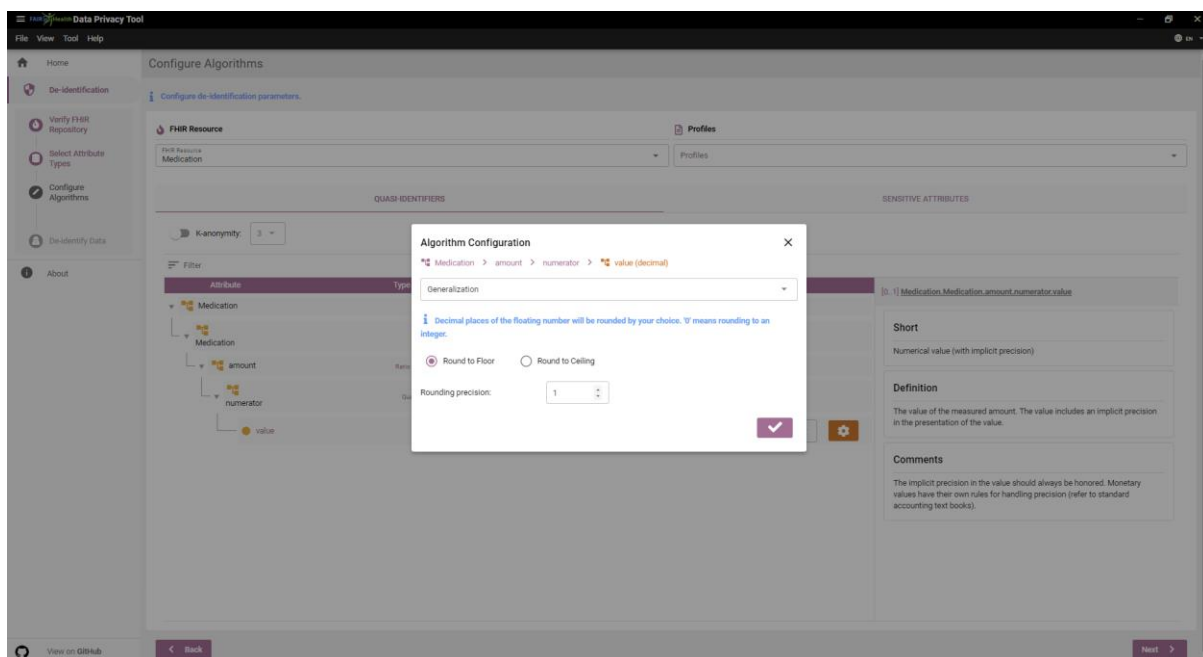
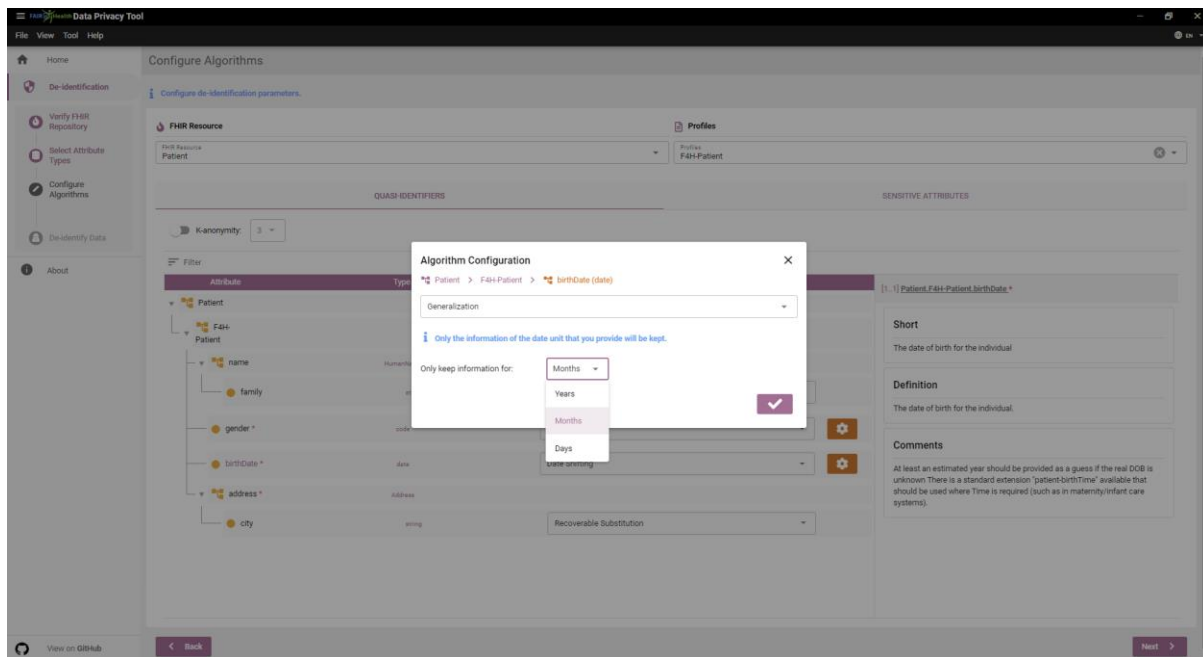


Figure 33. Generalization of Decimals



**Figure 34.** Generalization of Dates

### 2.2.1.8. Replace

This algorithm can only be used for sensitive rare values. Attributes are replaced with provided new values, as shown in Figure 9. This algorithm can be applied to the following primitive types in FHIR [2]:

- ❖ string
- ❖ uri
- ❖ url
- ❖ canonical
- ❖ code
- ❖ uuid

## 2.2.2. Privacy Criteria

### 2.2.2.1. K-anonymity

K-anonymity indicates that if the information for each resource in the dataset cannot be distinguished from at least k-1 resources whose information is also in the dataset, it can be referred to as "k-anonymous". It strengthens privacy by significantly decreasing the chance of identification. For example, if k value is 5 this means dataset will become 5-anonymous. In this situation, any resource in the dataset will have the same quasi-identifier

attributes with at least the other 4 resources. In Figure 35, you can see an example of a 4-anonymous dataset. For more information about k-anonymity, refer to [6].

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

**Figure 35.** Patient Data & De-identified 4-anonymous Patient Data [5]

#### 2.2.2.2. L-diversity

L-diversity technique describes that sensitive attributes would have at most same frequency as L. In Figure 36, you can see an example of a 4-anonymous dataset and L-diversity measures of its equivalence classes. For more information about L-diversity, refer to [5].

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

← 2-diverse

← 3-diverse

← 1-diverse

**Figure 36.** De-identified 4-anonymous Patient Data & Diversities of Each Equivalence Class [5]

### 2.2.3. Information Loss and Privacy Risks

#### 2.2.3.1. Information Loss

Information loss is estimated according to the Expected Equivalence Class Size metric. (Inspired from Average Equivalence Class Size metric [3], but we also took into consideration of weights of equivalence classes as in [4].) Restricted resources are excluded when doing it.

#### 2.2.3.2. Re-identification Risks

Re-identification risks are calculated according to the prosecutor risk model. Prosecutor risk is the risk that a specific person in the dataset can be re-identified when the attacker knows they are in the dataset. Prosecutor risks are calculated separately for each equivalence class<sup>2</sup>. So, the lowest, highest risks and average risks of all equivalence classes are computed by the Tool. For more details about re-identification risk calculations performed in the Tool, refer to supplementary material of [7].

#### 2.2.3.3. Records Affected by Risks

Percentage of identities in the dataset that has re-identification risks more than the lowest and highest prosecutor risks are calculated by the tool (section 2.2.3.2).

## 2.3. Tool Settings

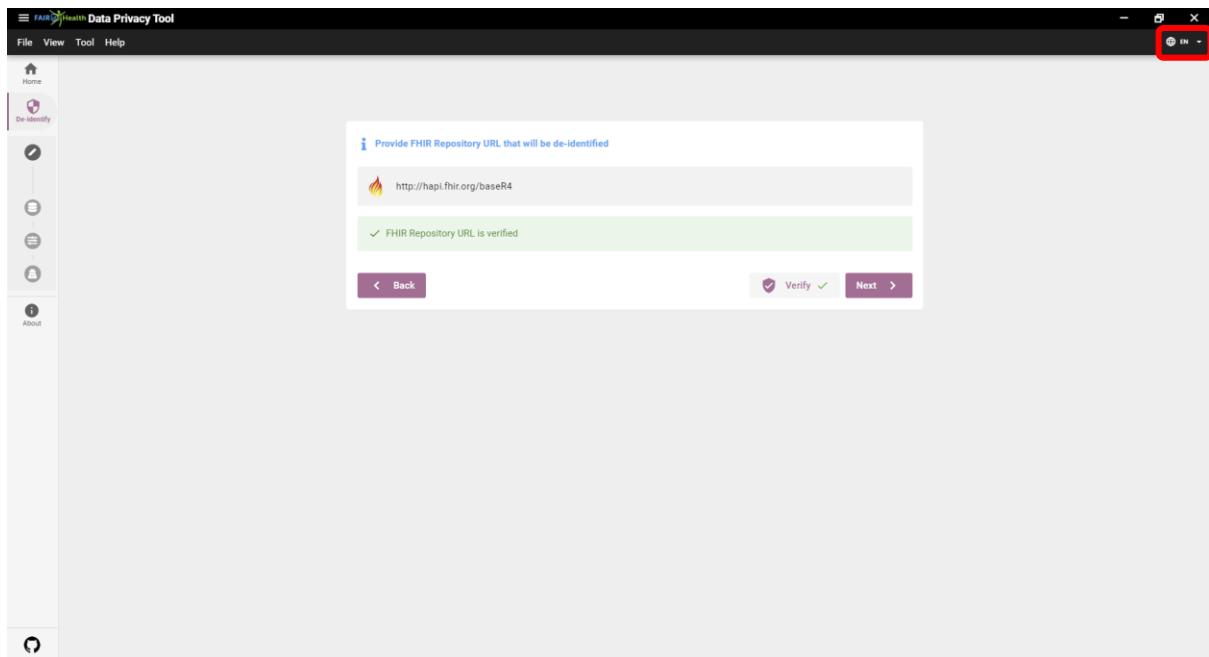
In this section, tool-wide settings are described, such as changing language, adjusting window size, developer options, etc.

### 2.3.1. Changing Language (Localization)

As shown in Figure 37, you can change the language by using the language options provided at the top right of the app.

---

<sup>2</sup> Equivalence class is a set of records that are indistinguishable from each other with respect to certain "identifying" attributes [8].



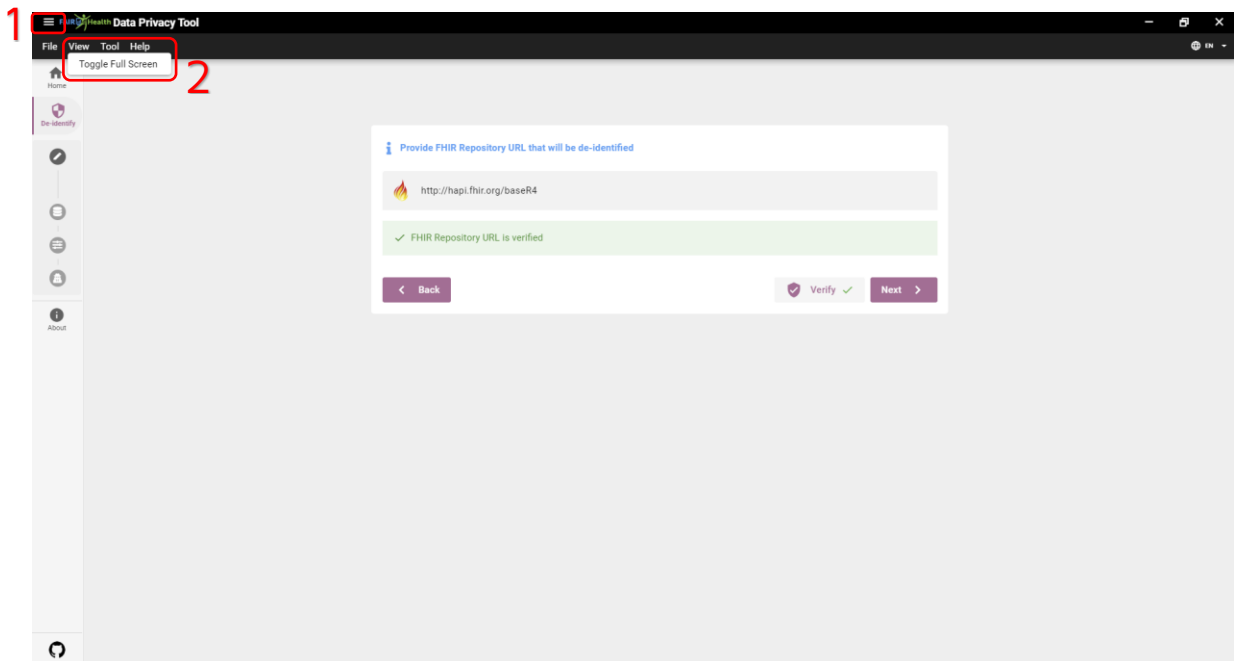
**Figure 37.** Tool Language Settings

### 2.3.2. Toggle Sidebar & Full Screen

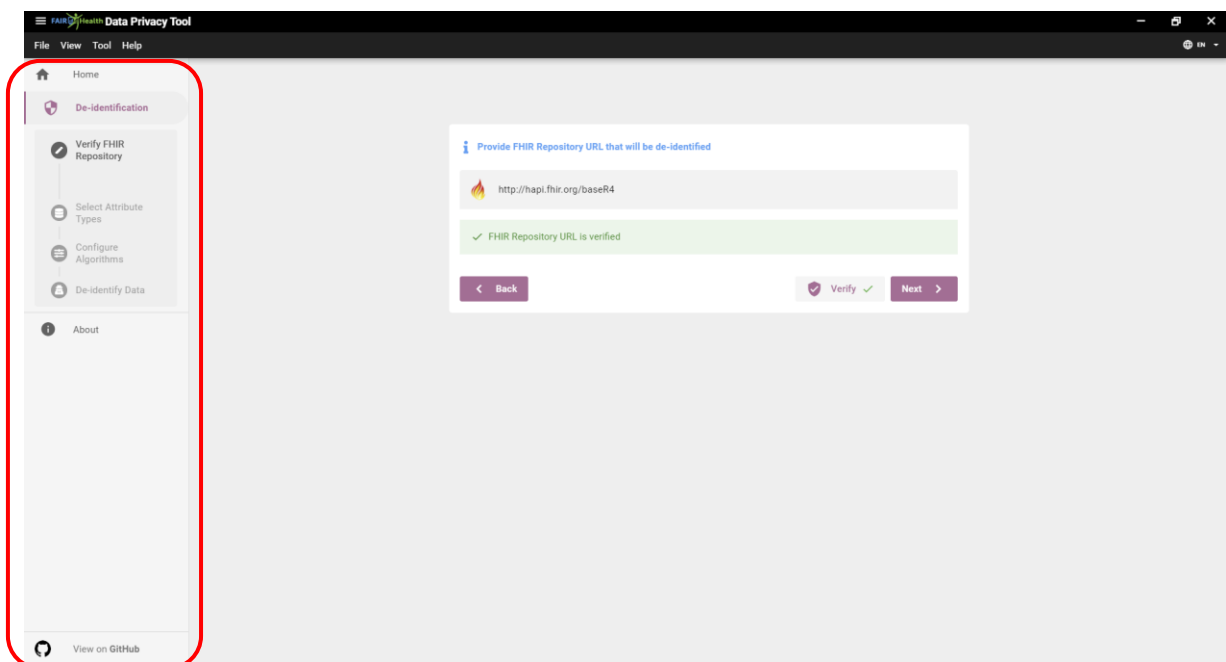
In **Full-Screen** mode, the Tool expands the main window to occupy the entire screen. It does the same thing on macOS and extends it to the dock. To fill the whole screen, the window expand button must be clicked, all operating system controls will be hidden, but you can access the main menu if you hover the mouse pointer over the top of the screen. Having these two expansion options in macOS is for ease of use. In Figure 38:

From **1**, you can toggle the sidebar. Its open form is shown in Figure 39.

From **2**, you can switch the full-screen mode. In Windows, you can also do this from the button in the upper left corner. But in macOS, the application view will look like this.



**Figure 38.** Tool Window View Options

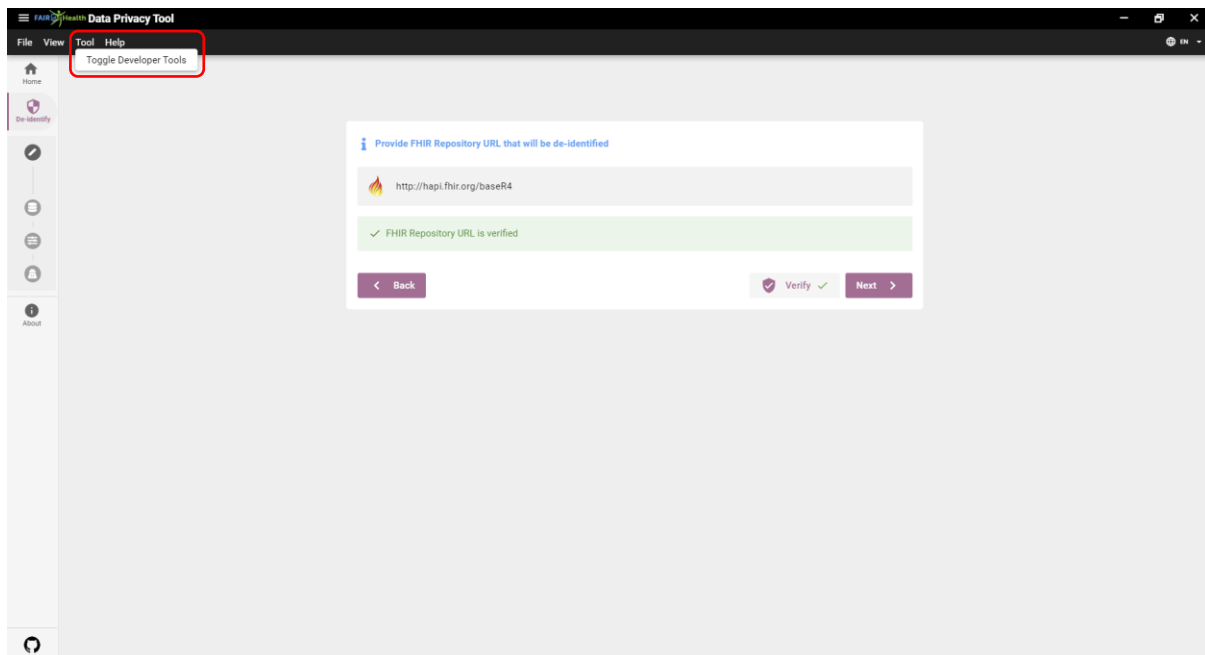


**Figure 39.** Opening Sidebar

### 2.3.3. Debugging

If you are a developer and the application is not behaving the way you wanted, an array of debugging tools might help you find coding errors, performance bottlenecks, or

optimization opportunities. The most comprehensive tool to debug the renderer process is the Chromium Developer Toolset, so to open dev tools, select **Tool -> Toggle Developer Tools** (Figure 40).



**Figure 40.** Opening Developer Tools

### 2.3.4. Log Locations

Data Privacy Tool log file contains information about events that have occurred from the app start to end. These events, including crash reports and errors, are logged out by the Tool, and written to the file.

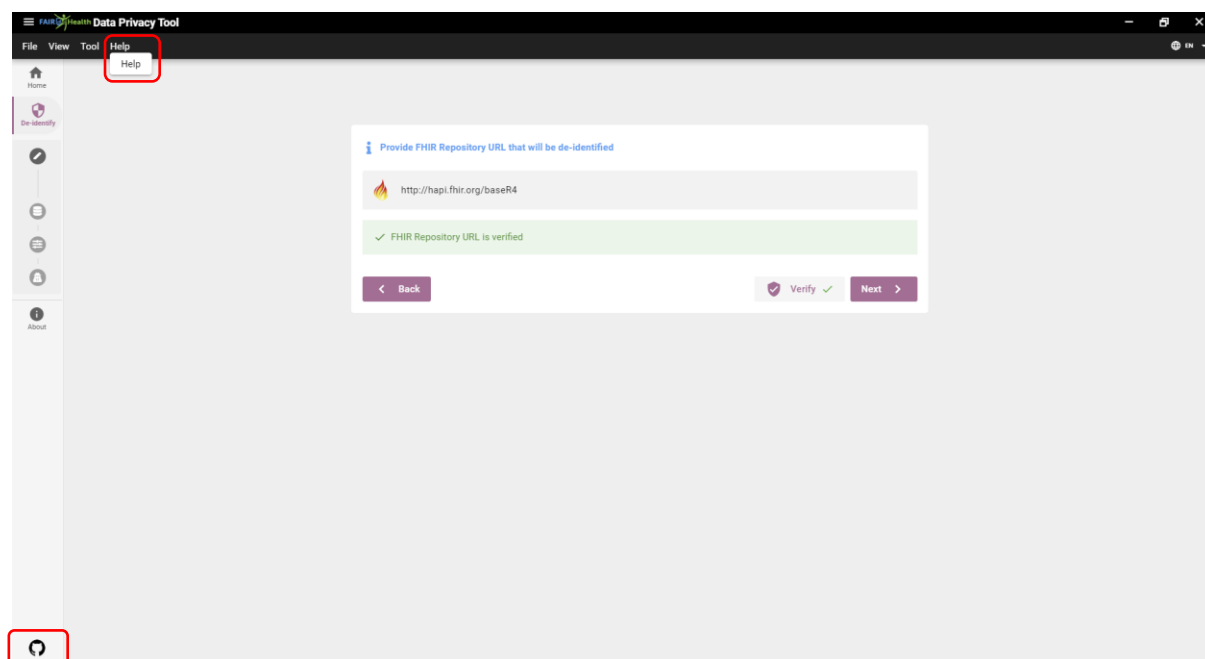
The Tool writes logs to the following locations:

- ❖ on **macOS**: ~/Library/Logs/FAIR4Health Privacy Tool/log.txt
- ❖ on **Windows**: %USERPROFILE%\AppData\Roaming\FAIR4Health Privacy Tool\logs\log.txt
- ❖ on **Linux**: ~/.config/FAIR4Health Privacy Tool/logs/log.txt



### 2.3.5. Help

If you have any questions, suggestions, or issues, you can contact us on the project GitHub page (Figure 41).



**Figure 41.** Help - GitHub Repository

## References

- [1] <https://www.hl7.org/fhir/valueset-security-labels.html>
- [2] <https://www.hl7.org/fhir/datatypes.html#primitive>
- [3] Kohlmayer F, Prasser F, Kuhn KA (2015). The cost of quality: Implementing Generalization and suppression for anonymizing biomedical data with minimal information loss. *Journal of Biomedical Informatics*. 58:37–48. ISSN 1532-0464. <https://doi.org/10.1016/j.jbi.2015.09.007>.
- [4] Altmann, G.T.M (1995). *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives*. Bradford Books.
- [5] Machanavajjhala A, Kifer D, Gehrke J, Venkatasubramanian M (2007). L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*. 3-es. <https://doi.org/10.1145/1217299.1217302>
- [6] Sweeney, L (2002). k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*. 10 (5), 557-570.
- [7] Prasser F, Spengler H, Bild R, Eicher J, Kuhn KA (2019). Privacy-enhancing ETL-processes for biomedical data. *International Journal of Medical Informatics*. 126:72–81, ISSN 1386-5056. <https://doi.org/10.1016/j.ijmedinf.2019.03.006>.
- [8] Li N, Li T, Venkatasubramanian S (2007). t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. *2007 IEEE 23rd International Conference on Data Engineering, Istanbul, 2007*, pp. 106-115, doi: 10.1109/ICDE.2007.367856.