ArtComputer

AI ROBOT

Automated Test with the Robot

AI Robot

Contents

Foreword

This user manual will give you information on how to use the AI Robot to generate the generic patterns to be used with the function detectGUI.

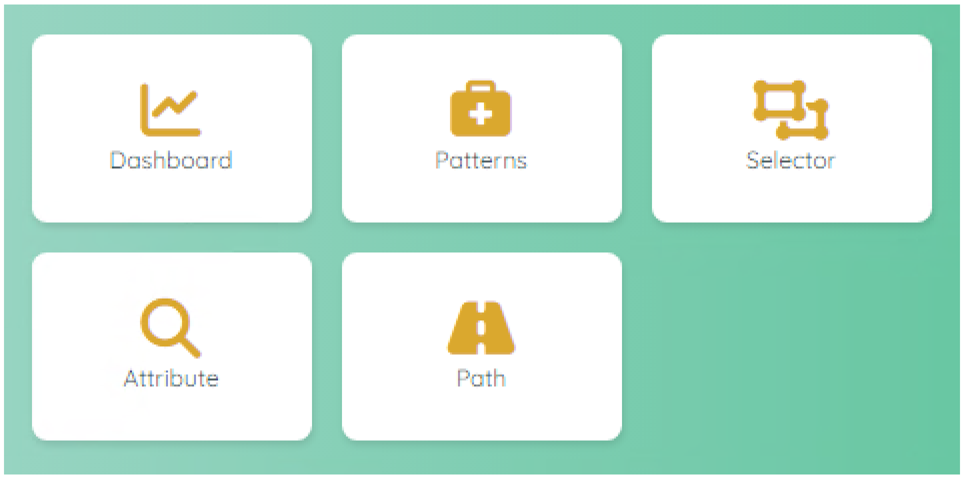
Warning: the job of the Admin will be to train the robot to detect an element.
This job requires a good skill in html and xpath. Admin will acts as 'The Expert'.
A bad expert advice will generate bad patterns!
The minimum skill required is:

- Be able to use the inspect (tool included in the Chrome or Firefox browser)
- Be confident with all the html tags
- Understand the concept of frame and iframe (if your application uses it)
- Be confident in the generation of a xpath (advanced level)

AI Robot:

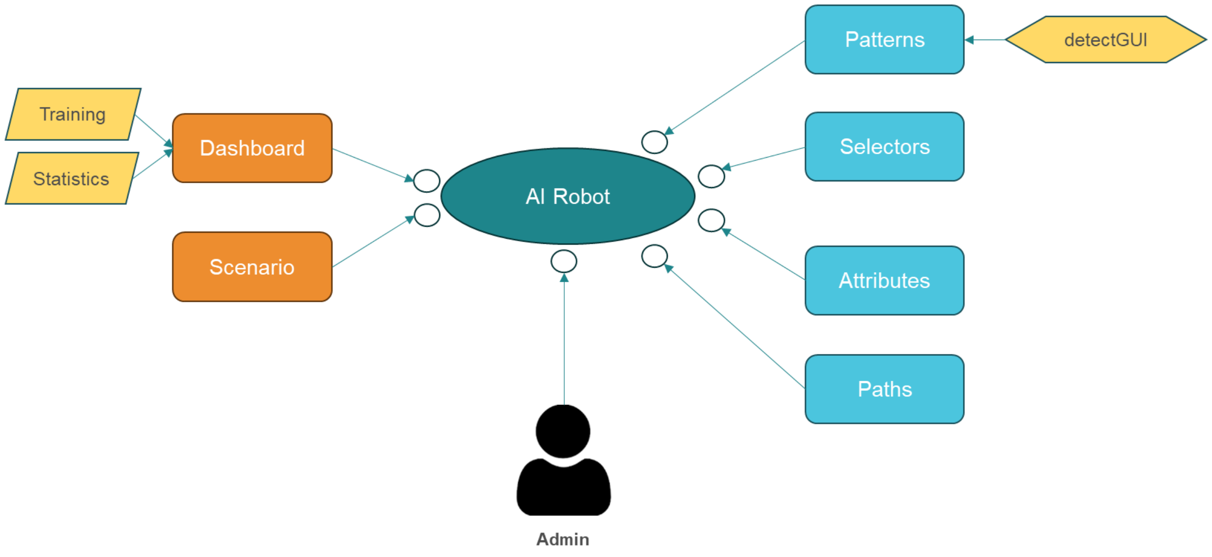
The AI Robot give you access to:

- The dashboard to train the Robot to detect a new selector.
- The patterns to manage the generic patterns generated during the training session.
- The selector to manage the html element (button, field…)
- The attribute to manage what element behavior will be important during the attribute.
- The path to manage the possible paths to explore during the training.



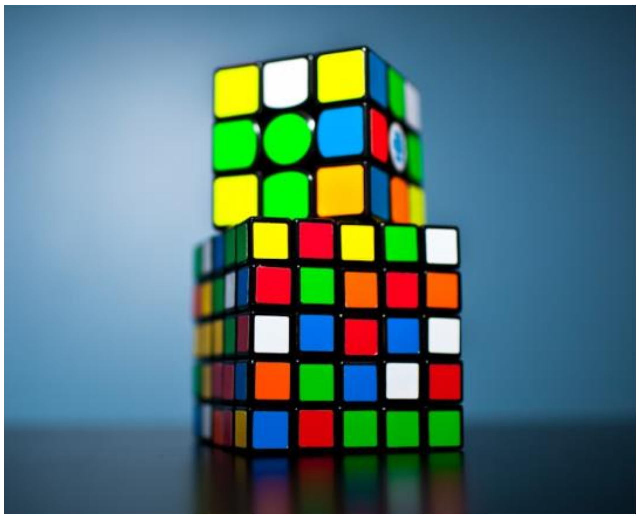
Architecture

To use the AI Robot, it’s important to understand. The concept behind each component of the robot.
The robot uses Artificial Intelligence and Statistics to generate generic patterns.
In the next section we will see what are the goals of each component.



A non-technical example

To better understand the concepts, we will use a common object to detect the elements.



What do you see on this picture?

- 2 Rubik's cubes
- A lot of colours
- A Rubik cube is composed of smaller elements (cubes, cylinders and rounded cubes)
- A black table
- A big cube and a small one
- The big cube is on the table, the small cube is on the big cube

Ok, now with all these information, we can build a strategy to detect the 2 Rubik's cube and the table.

The Paths

The path will give the robot the limit of the exploration.
In our example, we have a small cube above a larger cube, itself placed on a table (we have 3 levels to explore). But the table itself is placed on the floor of an apartment in a city, a country, a continent...

The robot is very brave and if you don't limit it, it will work for days to check if a cube can be placed on a chair. Is it still a cube if I change country...?

The objective of the path is to limit the robot's investigations.
This is the big difference between using AI to detect something that we know and using AI to research for an unknown concept (like a virus for instance).

In our case, we want to limit the number of level to explore (maximum 3) to speed up the learning process.

The Selectors

The selector will give the robot the objects that we want to analyse.
In our example, we have 2 selectors: a Rubik cube and a table

The Attributes

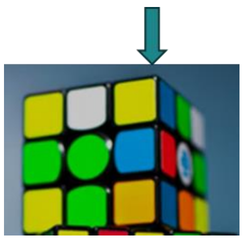
To be able to analyse elements, we must provide behaviours to the robot.
The attributes are generic, so for some selector an attribute can be not applicable.
Example of attributes could be: colour, size, surface, position, shape…

The Training

We are now ready to start our first training with the table.
Depending of the number of paths, the AI Robot will provide us multiple solutions.
For instance, a table can be seen at the first level (we have a table with a cube above) or we have a cube with below another cube and below a table (level 3).
The robot will use the 'distance to the object' to define the best solution. In our case, the first proposal (level 1) is better than the second proposal (level 3).
However, YOU are the EXPERT and it's up to you to decide what is the best proposal.
So you can now understand that if you don't have the right skill, you cannot generate the best pattern!

The Statistics

If we train the robot with only one picture, he will understand that a small Rubik cube contains always a cube yellow/blue at the right/top corner.



To avoid this problem, you need to train again the robot, with other pictures.
When he will discover another picture with another colours, he will remove the colour as a valid attribute to recognize an element.
In fact, there are two ways to work:

a) Robot discovers another colour and removes the attribute as a valid one
b) Robot discovers another colour and adds this new colour in a list of valid attributes (E.g.: a square can be red, yellow, blue, green or white)

To reduce the number of training sessions, you can use your skill to remove the unnecessary attributes.
Again, be careful but you are the EXPERT, right?

The Patterns

Finally after a few training sessions, we have now generic patterns.
Sometimes, you can have multiple patterns for the same selector. In this case, you can update the 'weight' to indicate your preference (E.g.: pattern 1: 100%, pattern 2: 90%)
Now the Designer is able to use the function detectGUI to very quickly and easily detect a selector.
Example: detectGUI (table) or detectGUI (Rubik, small)

User interface for the editing

User Interface

We use the same user interface when editing a record.
To avoid repeating the same explanation all the time, you will find here a generic explanation on how to use the edit interface.
When you have the permission to edit a record, you will see the following icons:

🗑 ✎ ⧉ ⤡ ☐

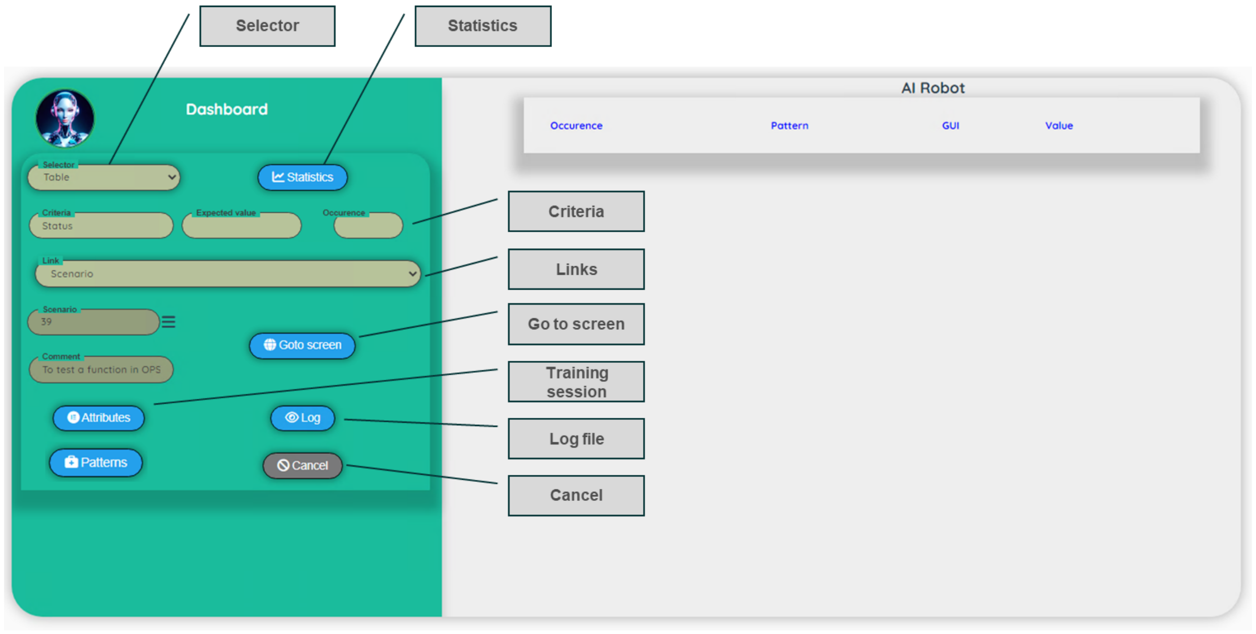
Topic	Icon	Comment
Delete	🗑 or ☑ 🗑	Delete a record (a confirmation is required!) To delete multiple records, select the records first.
Edit	✎	Go to a screen to update the record.
Copy	☑ ⧉	First select the record to copy and then click on the copy icon. Record(s) will be set after the copy icon.
Move	☑ ⤡	First select the record to move and then click on the move icon. Record(s) will be set after the move icon.
Select / Unselect	☑ / ☐	Select or unselect a record

Depending of the context, some extra icon(s) can be available.
The extra icon(s) will be explain in the section dedicated to the context.

Dashboard

User Interface

The dashboard will allow you to train the robot.



Topic	Comment
Selector	Select an existing selector.
Criteria	Indicate to the Robot how to detect the selector. Criteria is very often the label of the selector (or the header of a table). Expected value (optional) if the value of the selector is known (E.g.: a title) Occurrence (optional): for Analysis only to detect a specific occurrence (by default all the occurrences are detected)
Links	An URL or an existing scenario (use the hamburger to select the scenario)
Go to Screen	Open the URL or execute the scenario. At the end, the browser is still open and you can navigate to the point that you want to analyse.
Statistics	Run the statistics to generate patterns for the selected selector based on the previous training session.
Training session (Attributes)	Review all the training session and access to the attributes detected during the training.
Log file	Log file of the execution of the scenario, the analyse and the training
Patterns	Go to the patterns
Cancel	Return to the Control Panel

Example of a training session

For this example, we will train the Robot to detect a title.
We suppose that all the data are correctly configured (paths, selector, attributes…)

Open your application in the browser

1. **Selector**
   We need to select a selector. For this example, we use the selector 'Title'

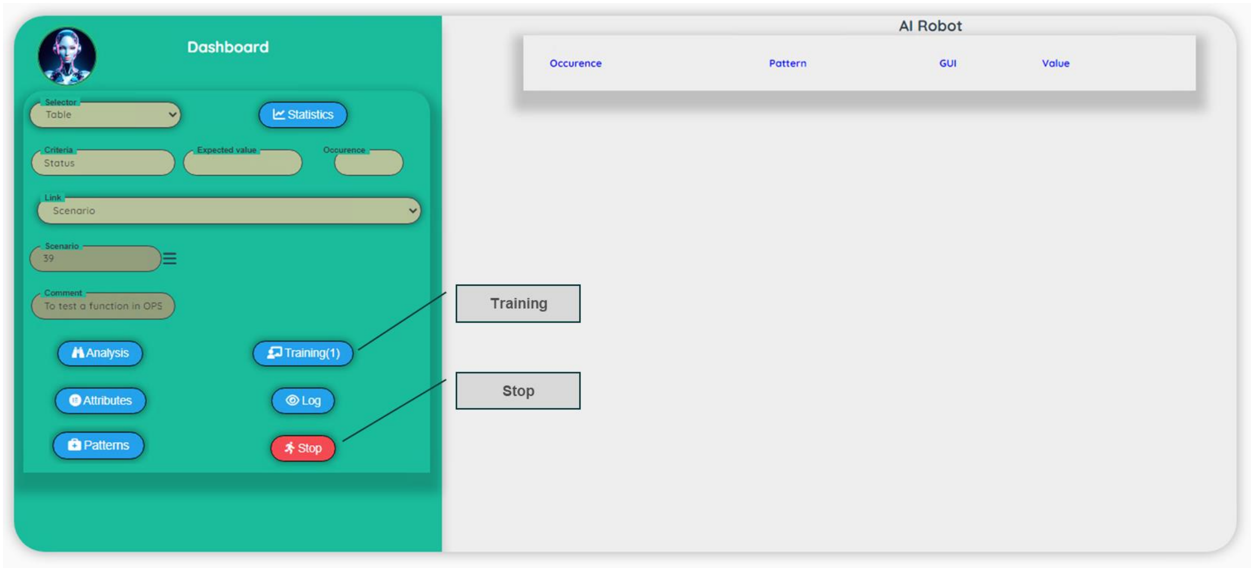
2. **Criteria**
   We are looking for a title with the value 'Portfolio'.
   We could key the Expected value ('Portfolio') to limit the number of possible solution.
   For this exercise, we will let the Expected value to blank.
   We don't need to fill the occurrence (it's only useful for the Analysis)

3. **Link**
   We select an existing scenario (using the hamburger). You can see the scenario ID and comment.

4. **Got o Screen**
   Click on the Go to screen to execute the scenario.
   At the end, the Robot displays: screen is ready.
   You can now, use your application to move to the correct screen (the screen with the title Portfolio)

Train the Robot

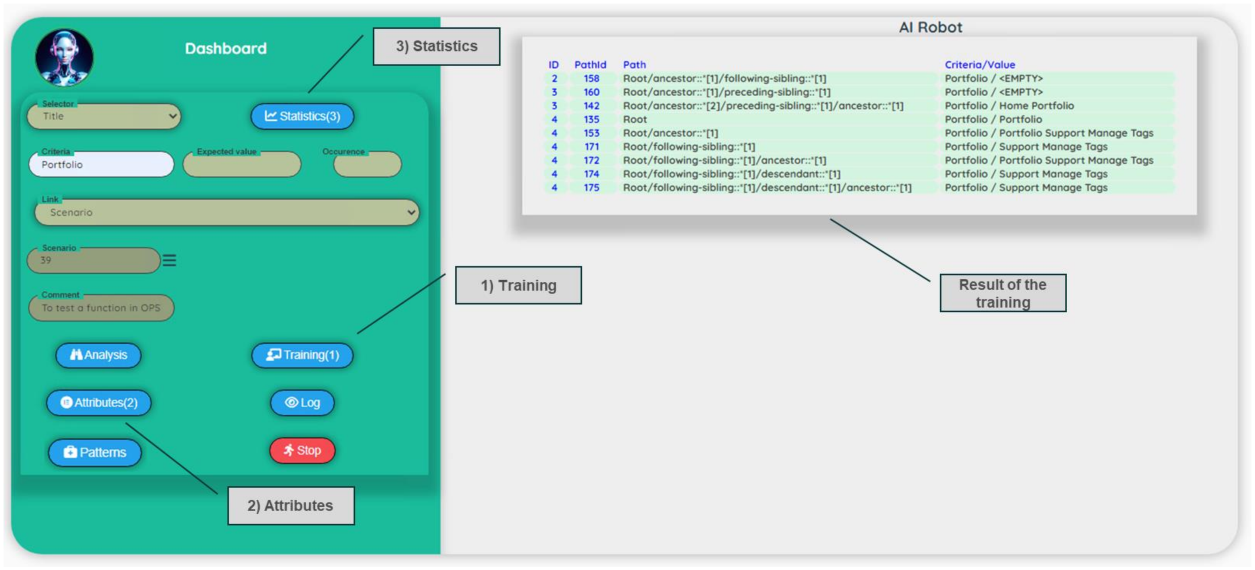
Now the screen looks like that



We have a new button Training and a button stop to close the browser.

1. **Training**
   Click on the Training button

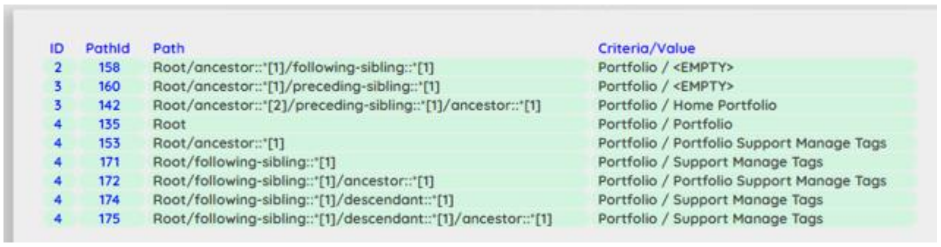
2. **Result of the training**
   After a few seconds, the screen looks like that



ID	PathId	Path	Criteria/Value
2	158	Root/ancestor::*[1]/following-sibling::*[1]	Portfolio / <EMPTY>
3	160	Root/ancestor::*[1]/preceding-sibling::*[1]	Portfolio / <EMPTY>
3	142	Root/ancestor::*[2]/preceding-sibling::*[1]/ancestor::*[1]	Portfolio / Home Portfolio
4	135	Root	Portfolio / Portfolio
4	153	Root/ancestor::*[1]	Portfolio / Portfolio Support Manage Tags
4	171	Root/following-sibling::*[1]	Portfolio / Support Manage Tags
4	172	Root/following-sibling::*[1]/ancestor::*[1]	Portfolio / Portfolio Support Manage Tags
4	174	Root/following-sibling::*[1]/descendant::*[1]	Portfolio / Support Manage Tags
4	175	Root/following-sibling::*[1]/descendant::*[1]/ancestor::*[1]	Portfolio / Support Manage Tags

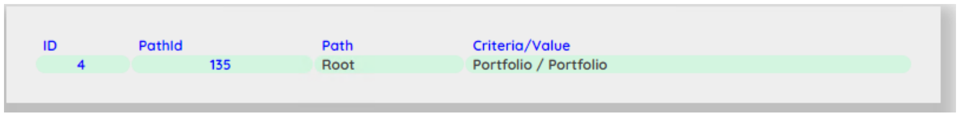
As you can see, the Robot as detected a lot of potential candidates.
This is due to the fact that we not key an expected value.

Without Expected value:



ID	PathId	Path	Criteria/Value
2	158	Root/ancestor::*[1]/following-sibling::*[1]	Portfolio / <EMPTY>
3	160	Root/ancestor::*[1]/preceding-sibling::*[1]	Portfolio / <EMPTY>
3	142	Root/ancestor::*[2]/preceding-sibling::*[1]/ancestor::*[1]	Portfolio / Home Portfolio
4	135	Root	Portfolio / Portfolio
4	153	Root/ancestor::*[1]	Portfolio / Portfolio Support Manage Tags
4	171	Root/following-sibling::*[1]	Portfolio / Support Manage Tags
4	172	Root/following-sibling::*[1]/ancestor::*[1]	Portfolio / Portfolio Support Manage Tags
4	174	Root/following-sibling::*[1]/descendant::*[1]	Portfolio / Support Manage Tags
4	175	Root/following-sibling::*[1]/descendant::*[1]/ancestor::*[1]	Portfolio / Support Manage Tags

With Expected value ('Portfolio')



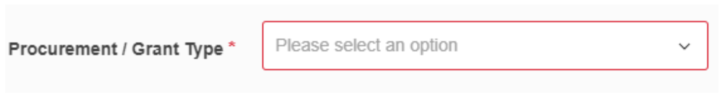
ID	PathId	Path	Criteria/Value
4	135	Root	Portfolio / Portfolio

We have only one proposal.
For the purpose of this exercise, we will work with the first training session (with all the potential candidates)

3. **Close the browser**
We don't need to keep the browser open, you can click on the stop button to close the browser.

<u>Note</u>: If the label contains a / (slash), the analysis and the training will not be able to discover a value. In this case, please take the left or the right part of the label.

Example:

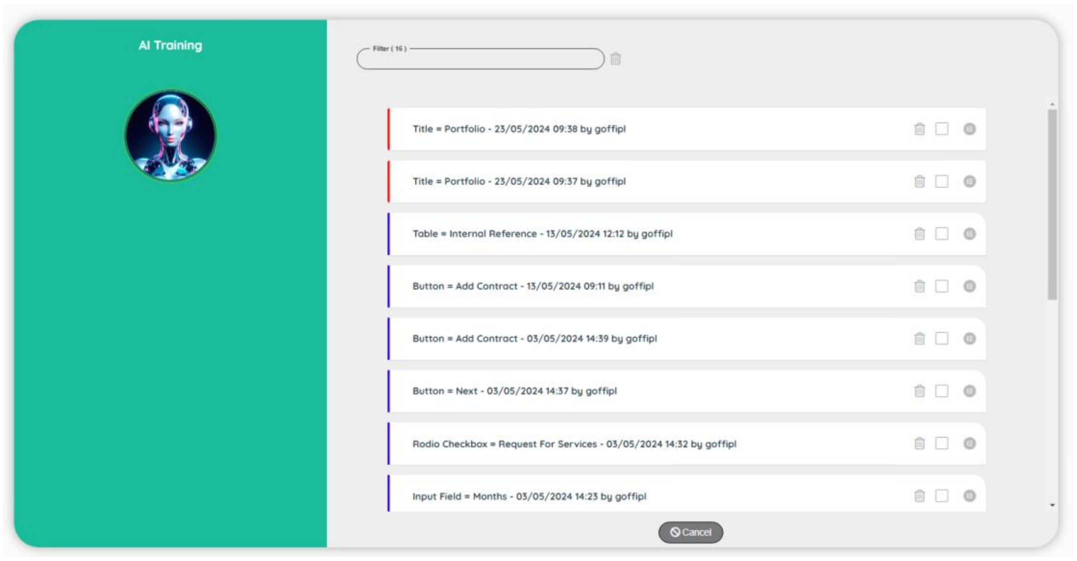


Procurement / Grant Type *

Please select an option

In this case, search for the label 'Procurement' or 'Grant Type'

Result of the training session



Title = Portfolio - 23/05/2024 09:38 by goffipl
Title = Portfolio - 23/05/2024 09:37 by goffipl
Table = Internal Reference - 13/05/2024 12:12 by goffipl
Button = Add Contract - 13/05/2024 09:11 by goffipl
Button = Add Contract - 03/05/2024 14:39 by goffipl
Button = Next - 03/05/2024 14:37 by goffipl
Rodio Checkbox = Request For Services - 03/05/2024 14:32 by goffipl
Input Field = Months - 03/05/2024 14:23 by goffipl

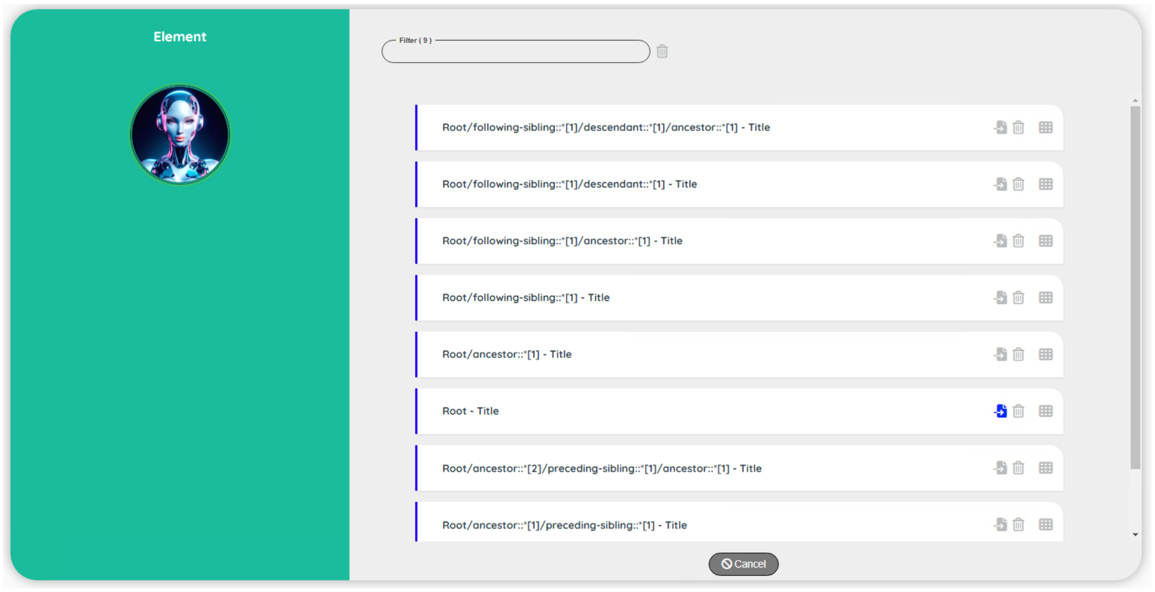
We have now a list with all the training sessions.
You can see that the first two sessions are with a red border. This means that the session is not completed. Only the completed session will be taken into account to generate patterns.

The first record correspond to the last session with only one proposal.
For the purpose of this exercise we will work with the second record (with all the potential candidates)

<u>Note:</u> As you can see, at this stage, we don't need any more to have the browser open.

Topic	Icon	Comment
Status	🟥 🟦	Processed / Not processed
Description		Selector / Criteria / date and user of the session.
Delete	🗑	Delete a previous session (including the attributes of the session).
Select to delete	☐	Select multiple session to delete.
View paths	●	View paths detected during the training session.

Analyse the paths of the session



Root/following-sibling::*[1]/descendant::*[1]/ancestor::*[1] - Title
Root/following-sibling::*[1]/descendant::*[1] - Title
Root/following-sibling::*[1]/ancestor::*[1] - Title
Root/following-sibling::*[1] - Title
Root/ancestor::*[1] - Title
Root - Title
Root/ancestor::*[2]/preceding-sibling::*[1]/ancestor::*[1] - Title
Root/ancestor::*[1]/preceding-sibling::*[1] - Title

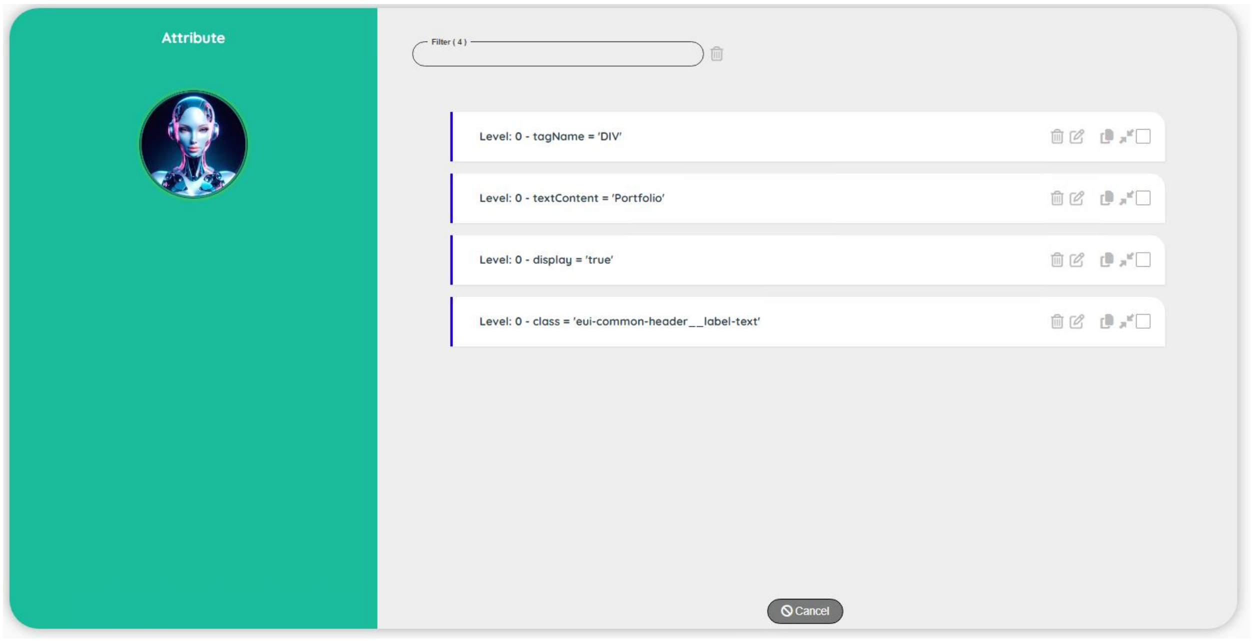
Topic	Icon	Comment
Path - Selector		Path and selector
Select	⇥ ⇥	Select a path that will become the pattern to use to generate
Delete	🗑	Delete a path (if you select a path, this operation will automatically done on all the unselected paths)
View attributes	●	View specific attribute values detected during the training session.

<u>Note:</u> You can see a specific blue selector for the path Root.
This is due to the fact that we already train the robot before and we used this path as the best pattern to use. In 99% of the cases this is the path to select.

If it's your first training for a selector, all the 'select' icons will be grey and you will have to select the best pattern.
A first consideration is the complexity of the path. It's always good to check if the simplest path is specific enough to do a correct job. Sometimes, you will prefer to have a more complex path that will give you specific attributes to detect a selector unambiguously.

To continue with our investigation, we need to check the attributes of the path (in this case, we will examine the attributes of the path Root – Title)

Analyse the attributes of a path



Level: 0 - tagName = 'DIV'
Level: 0 - textContent = 'Portfolio'
Level: 0 - display = 'true'
Level: 0 - class = 'eui-common-header__label-text'

Topic	Icon	Comment
Level – Attribute - Value		The level (0: root, 1: ancestor 1, 2: ancestor 2…) The attribute and its value.
Delete	⊘ or ✓ ⊘	Disable attribute(s) – The value, will be replaced by ?? and it will not be included in the pattern
Edit	✎	Go to a screen to update the record.
Copy	✓ ⧉	First select the record to copy and then click on the copy icon. Record(s) will be set after the copy icon.
Move	✓ ⤡	First select the record to move and then click on the move icon. Record(s) will be set after the move icon.
Select / Unselect	✓ / ☐	Select or unselect a record

Note: The level indicates the hierarchy in the path.
Level 0 means the Root. Root correspond to the search for the criteria
Level 1 means the parent of the root
Level 2 means the parent of the level 1
Level 3 means ….

The analysis of the path can be done form the root to the preceding element (down to up) or from the root to the ancestor (up to down)

Simplify an attribute

At this stage we can delete or simplify an attribute.

Disabling an attribute.

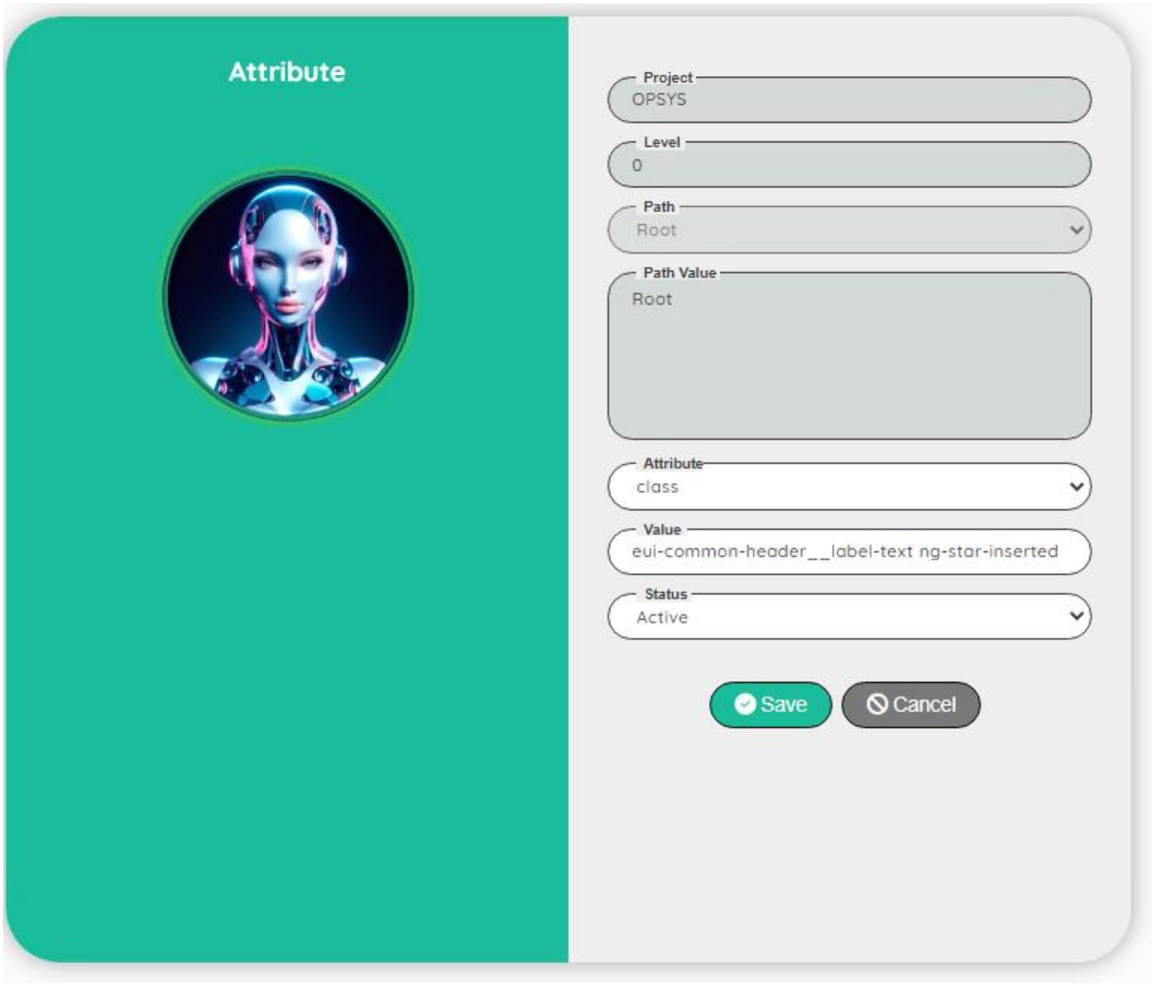
If you think that the attribute is not relevant for the detection of a selector, you can remove it (disable it).
Example: disable the attribute ID (if it is declared on your project) to avoid to have a very specific pattern.
Very often this is something that will perform on the 'class'.
Example: class = "ng-star-inserted" has no added value, we can disable it.

Note:
When you disable an attribute for one session, it will be effective for all the other sessions (with the same path and the same selector, of course)

Simplify an attribute.

Sometimes with the 'class', we have extra information not very useful and this extra information can even be problematic for the future detection as it will not be there all the times. In this case, we can simplify a class.

Click on the 'Edit' button to simplify the values of the attributes: Level 0 – class (the last record)



Project	OPSYS
Level	0
Path	Root
Path Value	Root
Attribute	class
Value	eui-common-header__label-text ng-star-inserted
Status	Active

The value: 'eui-common-header__label-text ng-star-inserted' contains 2 classes.
If the first one seems to be relevant: 'eui-common-header__label-text', the second class 'ng-star-inserted' is not relevant so we will edit the value to remove the second class to obtain: eui-common-header__label-text'.

<u>Note 1</u>: you can only simply by removing the beginning or the end of the value… not the middle.

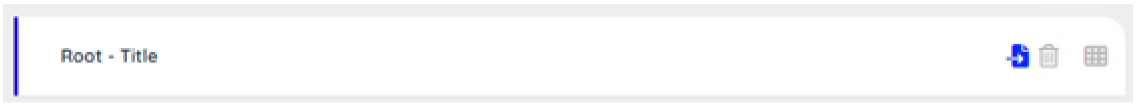
Example: class1 class2 class3
Valid classes are: 'class1', 'class1 class 2' or 'class2', 'class3', 'class2 class3"
But the class: 'class1 class3' is invalid!
(the robot will build a xpath with the function contains. Example: //*[contains(@class, 'class1')]

Note 2: It is not mandatory to simplify attributes, but it will drastically improve the speed of the training. If you don't simplify yourself, you need to let the Robot do the job with different training session (at least 3 or 4)

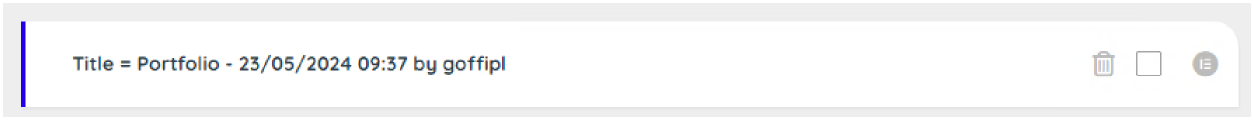
Note 2: The type of a button is set as 'submit' by default but if the attribute is not explicitly defined in the html, the pattern will not work properly! And you have to deactivate this attribute in the attributes of the training session.

Select the path to be the path to use to generate the patterns

Save your update (click on Save). Leave the attribute screen (click on Cancel). In the list of potential candidate paths, select the Root – Title (the one with a blue select icon).


Root - Title

Confirm your choice (click on Yes). You have now only one candidate selected
Leave the path screen (click on Cancel) to return back to the training session.
You have now the session with a blue bar.

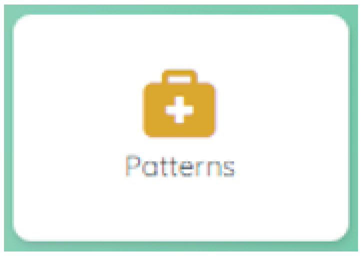

Title = Portfolio - 23/05/2024 09:37 by goffipl

Leave the training session screen (click on Cancel) to return back to the dashboard.

Statistics to generate the patterns

The statistics are a process to analyse the different training session in order to provide generic patterns for a selector.
Statistics are always linked to a specific selector.
Check that the 'Title' selector is still selected and click on the button 'Statistics'.
Congratulations, after a few seconds, your new generic patterns are ready!

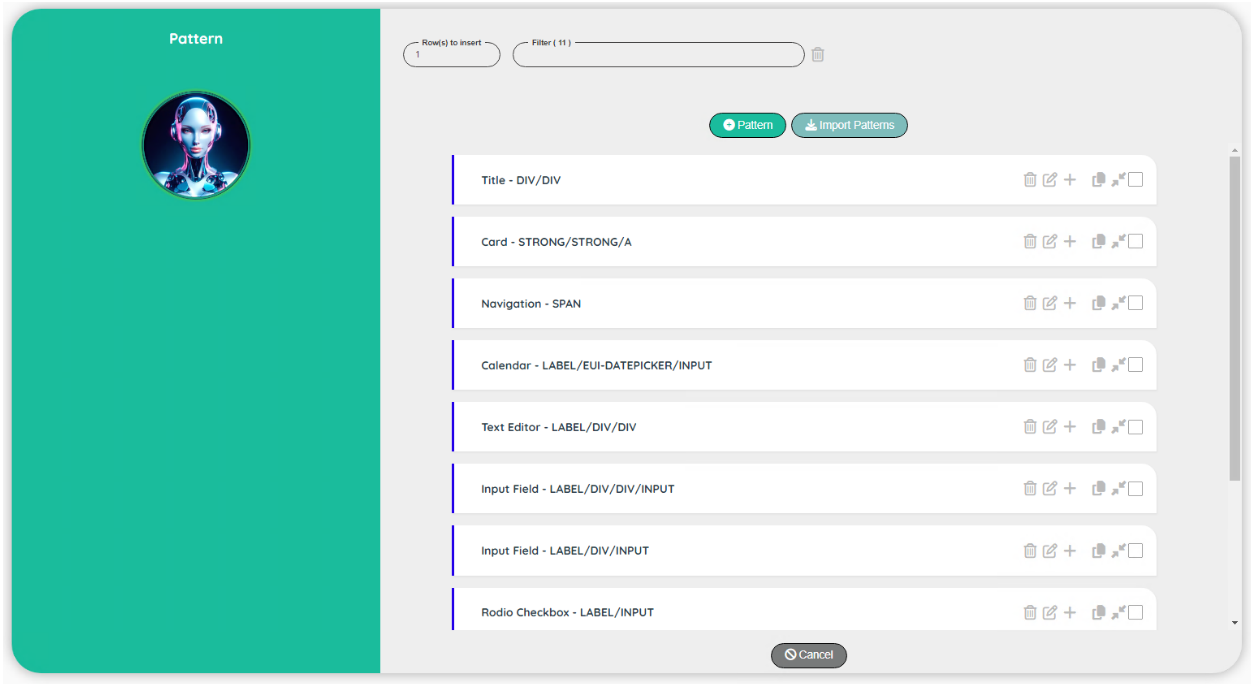
You can view the patterns via the menu patterns in the AI Robot screen.
This topic will be discuss in detail in the next section.



Patterns

User Interface

The patterns are used by the detectGUI function to discover an element on a webpage.
Patterns are generated by the action 'Statistics' in the dashboard (see previous section)



Title - DIV/DIV

Card - STRONG/STRONG/A

Navigation - SPAN

Calendar - LABEL/EUI-DATEPICKER/INPUT

Text Editor - LABEL/DIV/DIV

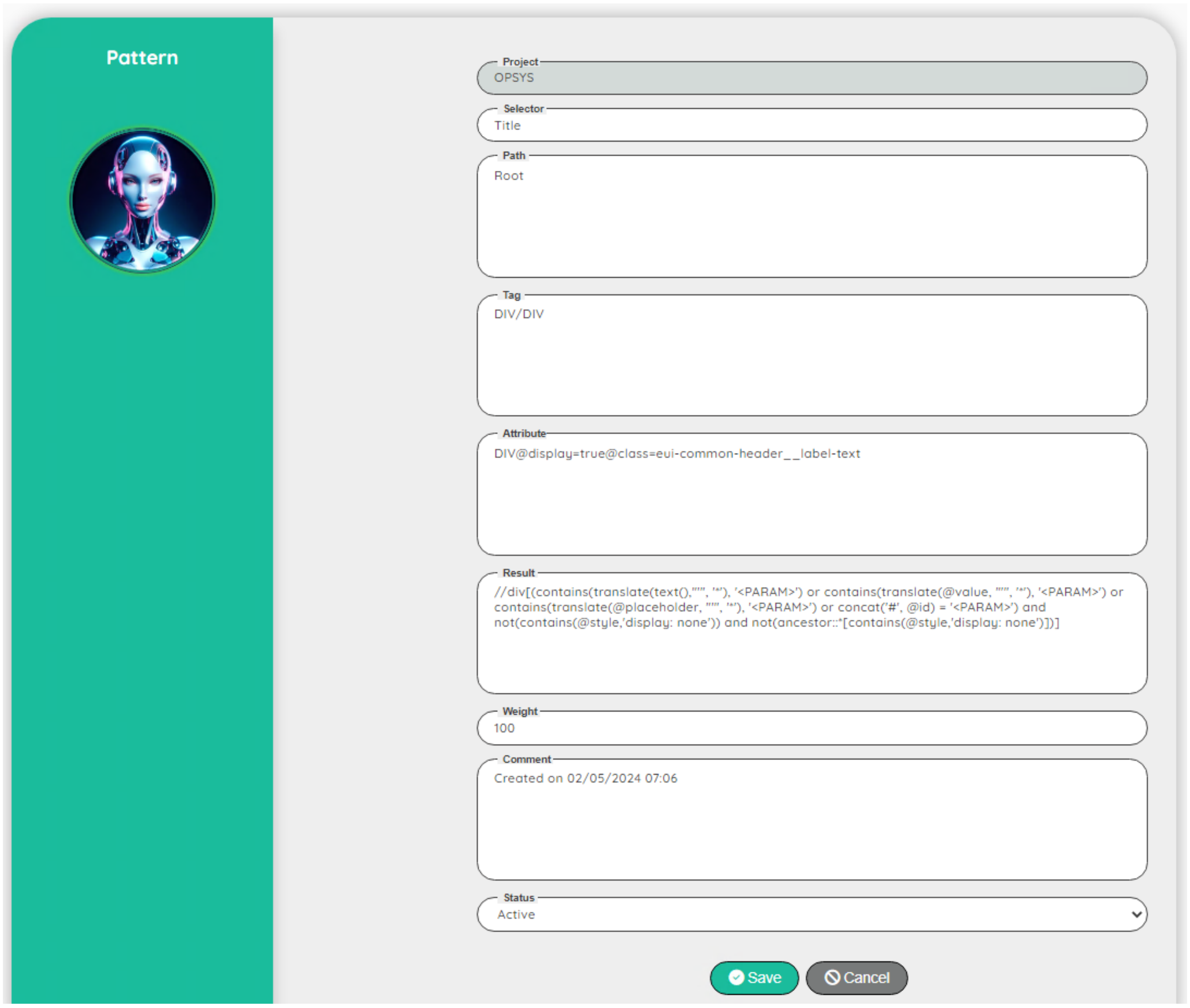
Input Field - LABEL/DIV/DIV/INPUT

Input Field - LABEL/DIV/INPUT

Rodio Checkbox - LABEL/INPUT

User Interface- Edit

The only field that you could be interested to change is the weight of the pattern.
When the detectGUI function is used, all the possible patterns for a selector are tested.
Depending of the context, the robot will reduce the original weight to get the best result.
However, it's possible that at the end two potential candidates have the same weight!
In this case, the detectGUI stops with an error 'Ambiguous patterns detected!: id:xxx and id: yyy'
A quick way to fix this issue is to go on the patterns and update the weight of one of them.
Example: id:xxx → weight: 100 and id:yyy → weight: 90



Project
OPSYS

Selector
Title

Path
Root

Tag
DIV/DIV

Attribute
DIV@display=true@class=eui-common-header__label-text

Result
```
//div[(contains(translate(text(),"", '*'), '<PARAM>') or contains(translate(@value, "", '*'), '<PARAM>') or
contains(translate(@placeholder, "", '*'), '<PARAM>') or concat('#', @id) = '<PARAM>') and
not(contains(@style,'display: none')) and not(ancestor::*[contains(@style,'display: none')])]
```

Weight
100

Comment
Created on 02/05/2024 07:06

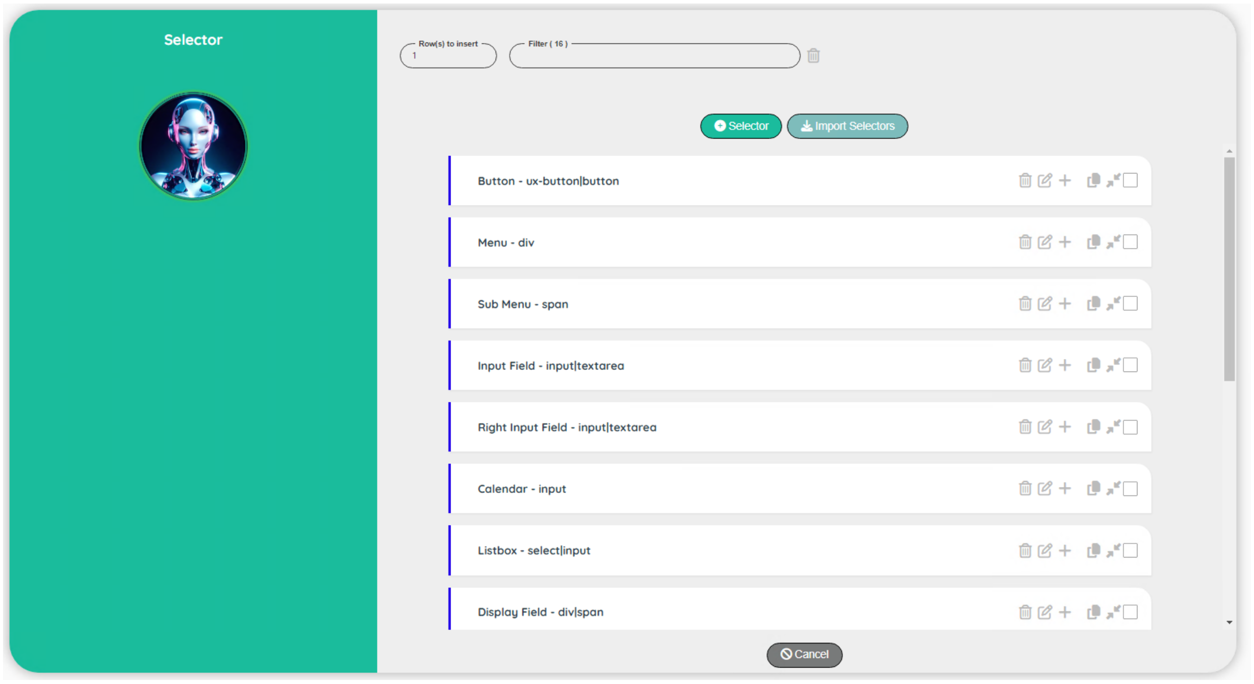
Status
Active

Selectors

User Interface

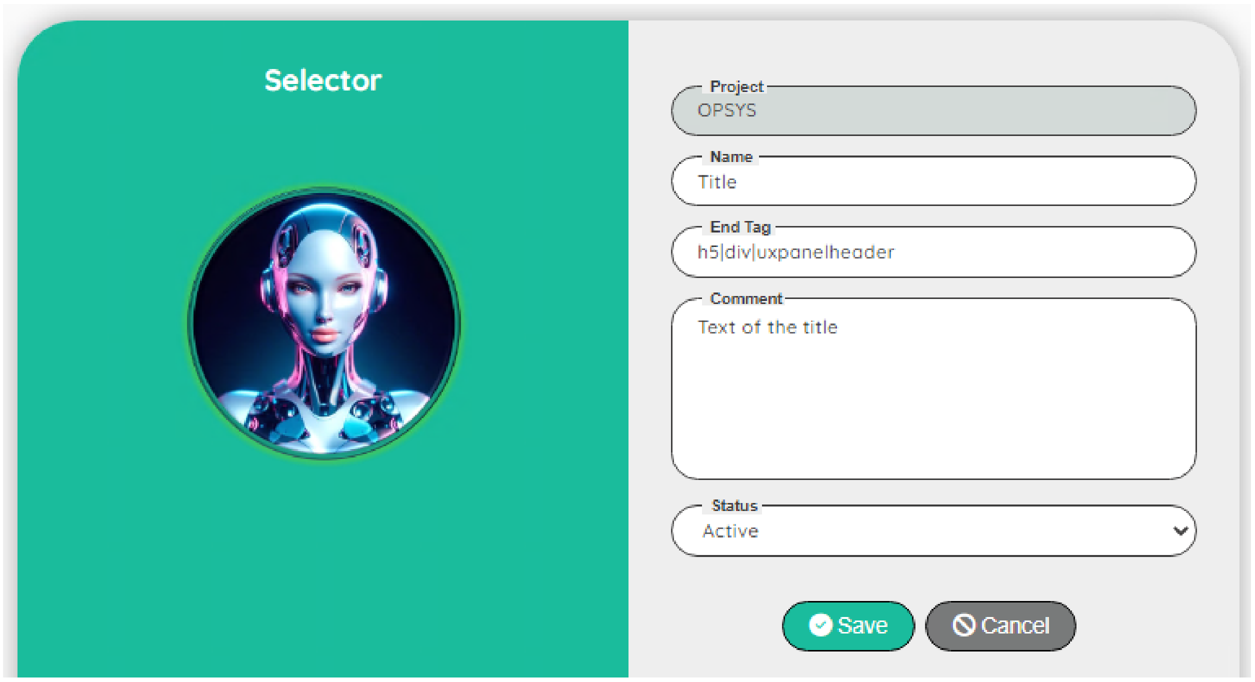
The selectors depend of your need on your project.
For instance, you might need a display field selector and an input field selector.
May be you might need to have a special selector for a rich text field…

If you start a new project, you can import a basic template that you can adapt to your need



Button - ux-button|button
Menu - div
Sub Menu - span
Input Field - input|textarea
Right Input Field - input|textarea
Calendar - input
Listbox - select|input
Display Field - div|span

User Interface- Edit



Project: OPSYS
Name: Title
End Tag: h5|div|uxpanelheader
Comment: Text of the title
Status: Active

Topic	Comment
Project	For info: Name of the project
Name	Identifier of the selector
End tag	A list of tag (separated by a |) that must be present at the end of the path
Comment	Comment on the selector
Status	Active / Not Active

Note: End tag is used to filter the possible candidates during the training

Example: for a title the last tag must contain h5 or div or uxpanelheader
For instance suppose that during the training we have something like that:

- Proposal 1: <h5>Portfolio</h5>
- Proposal 2: <span>Portfolio</span>

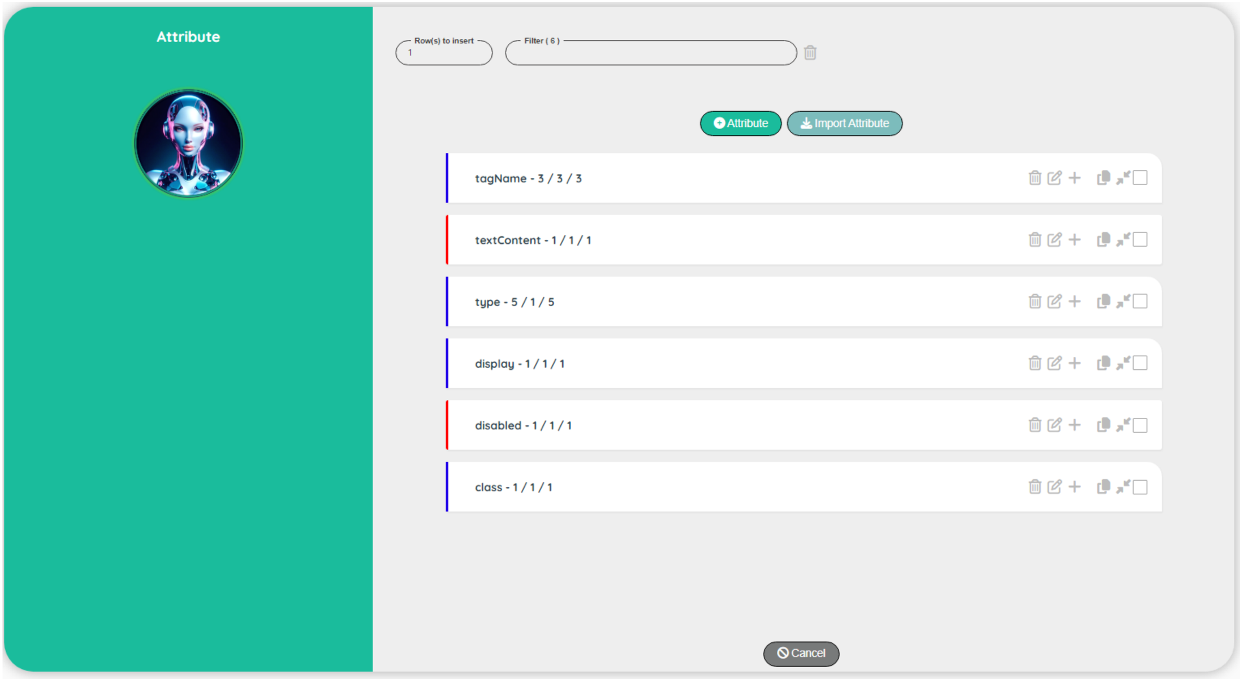
Question is how do you know the end tag to be used?
You need first to investigate with the inspect within your application.
If the training returns no candidate, you can inspect the application and check that the tag that you see in the application is also present in the selector.

Attributes

User Interface

The attributes are used by the statistics to check the relevant attributes to be used to build a generic pattern. It is not necessary to use a lot of attributes.

If you start a new project, you can import a basic template that you can adapt to your need



tagName - 3 / 3 / 3
textContent - 1 / 1 / 1
type - 5 / 1 / 5
display - 1 / 1 / 1
disabled - 1 / 1 / 1
class - 1 / 1 / 1

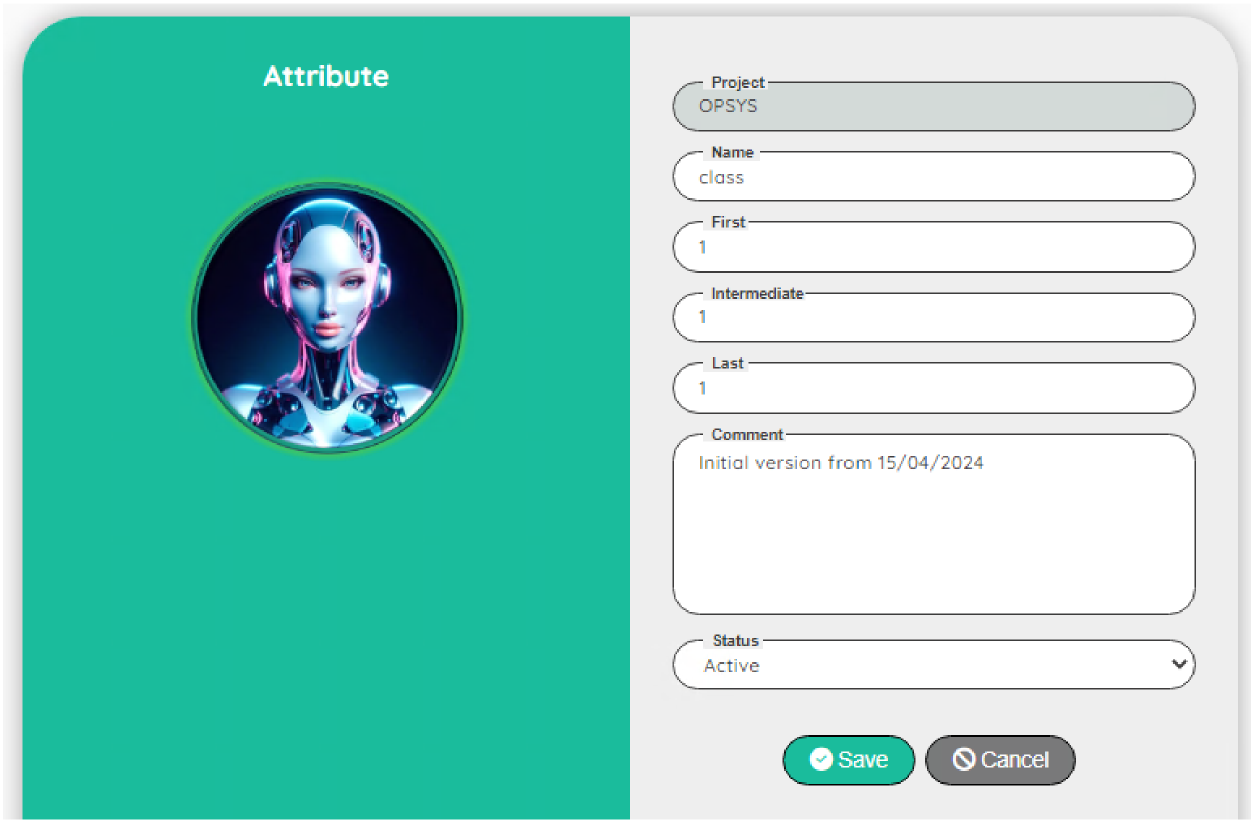
Note 1: if you disabled an attribute, it will be displayed in the training session (with the status not active) and it will not take into account during the statistics to generate the pattern.
This option is useful if you want to display the attribute for your information

Note 2: textContent is never used during the statistics to generate the pattern.
The status Active or Not active has no effect

Note 3: display, visible and disabled are only used for the last part of the xpath

User Interface- Edit

The attribute must be a valid html attribute



Project: OPSYS
Name: class
First: 1
Intermediate: 1
Last: 1
Comment: Initial version from 15/04/2024
Status: Active

Topic	Comment
Project	For info: Name of the project
Name	Valid html attribute (case sensitive)
First	Number of possible occurrence in the first tag
Intermediate	Number of possible occurrence in the intermediate tags
Last	Number of possible occurrence in the last tag
Comment	Comment on the selector
Status	Active / Not Active

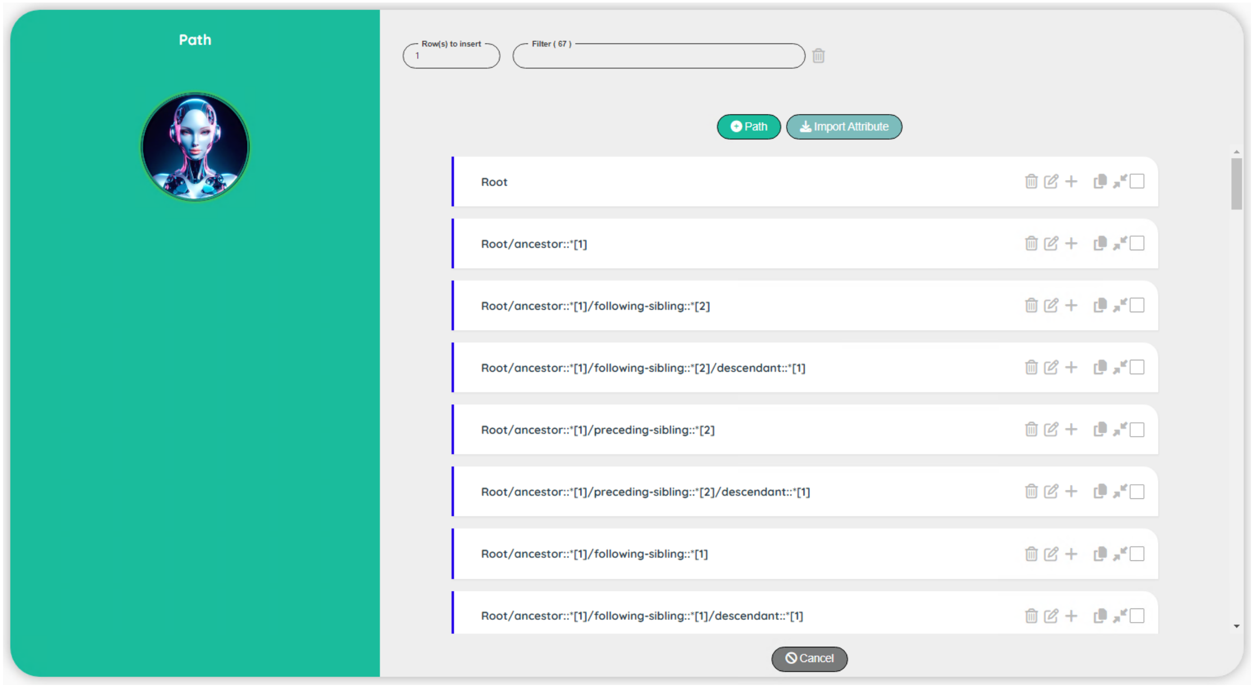
Note: First, Intermediate and Last are used to define if we need to build a new pattern when we see a difference or if we can include the new value as a possible list of value for an existing pattern.
Example : First level for the tag is 1: in this case <h5>Portfolio<h5> and <div>Portfolio<div> will generate 2 different patterns. If the First level is now 3 we will generate a pattern with a list of possible tag like: h5|div. This will impact the performances of the detectGUI function.

Path

User Interface

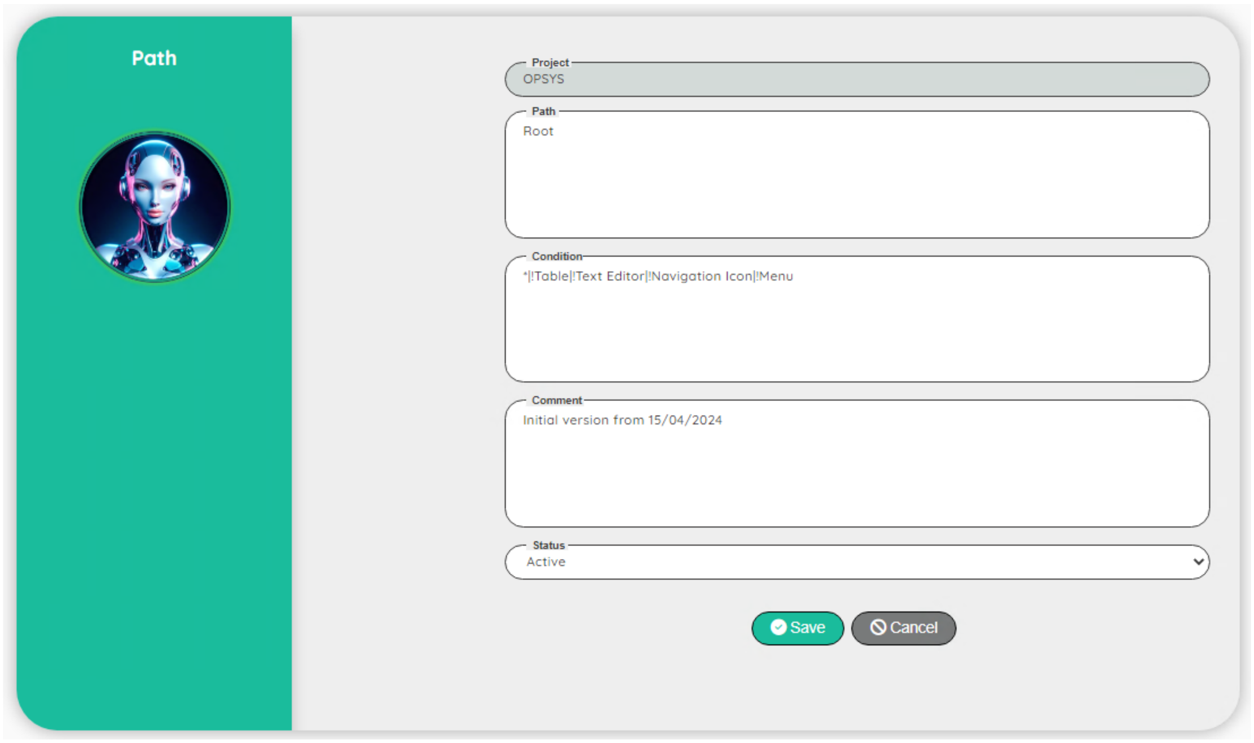
The Path are used during the training to provide a guideline to the Robot.
With a given set of paths, we can control the performance of the training (a training session is about 10 to 20 seconds)

If you start a new project, you can import a basic template that you can adapt to your need



```
Root
Root/ancestor::*[1]
Root/ancestor::*[1]/following-sibling::*[2]
Root/ancestor::*[1]/following-sibling::*[2]/descendant::*[1]
Root/ancestor::*[1]/preceding-sibling::*[2]
Root/ancestor::*[1]/preceding-sibling::*[2]/descendant::*[1]
Root/ancestor::*[1]/following-sibling::*[1]
Root/ancestor::*[1]/following-sibling::*[1]/descendant::*[1]
```

User Interface- Edit



Project: OPSYS

Path: Root

Condition: *|!Table|!Text Editor|!Navigation Icon|!Menu

Comment: Initial version from 15/04/2024

Status: Active

Topic	Comment
Project	For info: Name of the project
Path	The path starting with the word Root
Condition	The possible selector(s) that can be used this path
Comment	Comment on the selector
Status	Active / Not Active

Path

A path start always with a Root.
Root will be replaced during the training by an expression to look for your criteria.
You can have more complex path like: Root/ancestor::*[1]/following-sibling::*[2]/descendant::*[1]

You can even build a very specific path for a specific selector:
Example a dedicated path for the selector: Menu
Root/ancestor::span[contains(@class, 'eui-menu-item__link-label')]/span[contains(@class, 'eui-menu-item__label')]

The syntax is exactly the same as for a normal xpath except that the first tag must be Root.
Note: of course you cannot try the path in the inspect tool but you can easily replace the Root by //*[contains(text(), ‘Portfolio’)] for instance.

Condition

Condition is used to restrict the path to selector(s)
You can have a single selector name: like Menu to restrict the path to only the menu selector.
You can have multiple selectors by excluding some ones with: *|!Table|!Radio Checkbox|!Menu
(all the selectors except the Table, Radio Checkbox and the Menu)
You can have all selectors with: *

Note 1: in the new version (06/2024), Condition is just an information, and it will not use anymore by the detectGUI function.

Note 2: If you want to create a new Path, it's highly recommended to prefix all the sub paths by a position (except. for Root)
For instance: Root/ancestor::span[contains(@class, 'eui-label')] is not a secure path.
A better path is: Root/ancestor::span**[1]**[contains(@class, 'eui-label')]

The reason is linked with the training process.
During the training, the Robot will try to detect the attributes of each part of the Path.
For the Root, all the occurrences found will be tested, but for the sub path, only the first one will be considered!
Be careful, because your path can work on your specific example (for instance, if the first sub path is the best choice) and fails later on other example, because the second sub path must be used!

To summarize:

- If the Robot cannot detect an element based on the provided Paths, you can create your own path.
- If you want to create your own path, it's highly recommended that you prefix the tags of each sub path by the position (if necessary, create multiple paths with different positions)