

# **xTwins User Guide**

## **Matilde Agent Documentation**

**Copyright © Ritain.io** (Binary Heroes, LDA)

All information in this document is confidential and property of Ritain.io (hereinafter “supplier”) and meant for use only by the client and only in connection with and subject to the tender. Client shall not disclose or otherwise make available to any person who is not an employee of the client with a definite need to know, or to any other entity, any of the information contained herein, without the written consent of the supplier. Any entity or person with access to this information shall be subject to this confidentiality statement. No part of this document may be reproduced or transmitted in any form or by any means for any purpose without the express



written permission of supplier. In the event this document originates a contract between the parts, the terms and conditions of the contract regarding confidentiality shall apply.

## Table of contents

1.	Introduction.....	3
1.1.	Purpose of the Documentation .....	3
2.	Overview of the Agent .....	3
2.1.	Purpose.....	3
2.2.	Key Functionalities .....	3
3.	How Matilde Works.....	4
3.1.	Input Explanation .....	4
3.2.	Interaction Example .....	4
4.	Frequently Asked Questions (FAQs) .....	6
5.	Support and Contact Information.....	7



## 1. Introduction

### 1.1. Purpose of the Documentation

This document provides a detailed usage guide for Matilde agent. It includes an introduction to the agent, a description of its purpose/objective, an enumeration of its features, and a listing of its different versions. It is designed to assist you in integrating and utilizing Matilde to enhance your daily productivity.

## 2. Overview of the Agent

### 2.1. Purpose

Matilde is a specialized agent designed to assist clients in extracting XPath's from mobile applications using Appium. With expertise in navigating intricate mobile UI structures, Matilde ensures accuracy and efficiency in mobile data scraping tasks. Matilde also interprets provided XML code and offers personalized XPath suggestions, ensuring a smooth workflow for developers and testers of mobile projects.

### 2.2. Key Functionalities

Matilde contains a powerful set of capabilities to streamline and enhance the extraction of XPath's from mobile applications. These capabilities make Matilde a go-to agent when working on mobile projects. The following functionalities highlight the key features of Matilde:



- **XML Parsing and Analysis**
  - Matilde efficiently parses provided XML data identifying key attributes and elements for XPath generation.
- **Mobile Application Expertise:**
  - Matilde is optimized in XPath extraction for mobile applications based on XML code snippets provided by clients.
- **Elements Prioritization:**
  - Clients have the flexibility to describe specific elements in the page that should receive additional attention by Matilde during the XPath extraction process.
- **Output Consistency:**
  - Ensures that identical XML code produces similar XPaths, enhancing reliability on the generated output.
- **Multiple Output Formats:**
  - Matilde provides flexibility in the output structure, meaning, it supports XPaths extraction to different formats (e.g., json, yaml, ...).
- **Visual Image Support:**
  - Images of the web page can be incorporated in the request to improve Matilde's response

## 3. How Matilde Works

### 3.1. Input Explanation

#### Input Request – Required Variables:

- **xmlCode:** string
  - Provide the XML code snippet that Matilde should extract XPaths from.
- **outputFormat:** string
  - Specify the desired output format of the generated XPaths.

#### Input Request – Optional Variables:

- **elementsDescription:** list<string>
  - List of descriptions. Each description outlines an element or set of elements on the mobile page that Matilde should focus on during the XPath extraction process.
- **images:** list<file>
  - List of images (e.g., interface screenshots) that contain valuable information to improve Matilde's response.

### 3.2. Interaction Example

#### Input Example:



- **xmlCode:**

```
<android.widget.RelativeLayout index="1" package="com.nasctech.streamline"
class="android.widget.RelativeLayout" text="" checkable="false" checked="false"
clickable="false" enabled="true" focusable="false" focused="false" long-
clickable="false" password="false" scrollable="false" selected="false"
bounds="[132,634][948,755]" displayed="true">
    <android.widget.ImageView index="0" package="com.nasctech.streamline"
class="android.widget.ImageView" text="" resource-
id="com.nasctech.streamline:id/iconLock" checkable="false" checked="false"
clickable="false" enabled="true" focusable="false" focused="false" long-
clickable="false" password="false" scrollable="false" selected="false"
bounds="[176,661][234,727]" displayed="true" />
    <android.widget.EditText index="1" package="com.nasctech.streamline"
class="android.widget.EditText" text="Password" resource-
id="com.nasctech.streamline:id/password" checkable="false" checked="false"
clickable="true" enabled="true" focusable="true" focused="false" long-
clickable="true" password="true" scrollable="false" selected="false"
bounds="[278,661][832,728]" displayed="true" />
    <android.widget.ImageView index="2" package="com.nasctech.streamline"
class="android.widget.ImageView" text="" resource-
id="com.nasctech.streamline:id/iconEye" checkable="false" checked="false"
clickable="true" enabled="true" focusable="true" focused="false" long-
clickable="false" password="false" scrollable="false" selected="false"
bounds="[832,672][948,716]" displayed="true" />
</android.widget.RelativeLayout>
```

- **elementsDescription:** []
- **outputFormat:** yaml
- **images:** null

## Output Example:

```
locators:
  lock_icon:
    xpath: .//android.widget.ImageView\[@resource-
id='com.nasctech.streamline:id/iconLock'\]
    wait: timeouts.defaults.small
    description: > Icon representing a locked item indicating that the
user must provide a password.
  password_field:
    xpath: .//android.widget.EditText\[@resource-
id='com.nasctech.streamline:id/password'\]
    wait: timeouts.defaults.small
    description: > Input field for the user to enter their password. The
field is password protected.
  eye_icon:
    xpath: .//android.widget.ImageView\[@resource-
id='com.nasctech.streamline:id/iconEye'\]
    wait: timeouts.defaults.small
    description: > Icon that toggles the visibility of the password
entered in the password field.
```



## 4. Frequently Asked Questions (FAQs)

We will continuously update this FAQ section with the most commonly asked questions and insights from our clients to better support your needs.



## 5. Support and Contact Information

For further assistance, please contact our support team through our service desk system:

- <https://support.jira.ritain.io/servicedesk/customer/user/login?destination=portals>