

# Training Guide

## Mozenda Migration



### A Feature Comparison between Dexi and Mozenda

#### Table of Contents

- 1. Introduction** ..... 2
- 2. Key Differences Between Platforms** ..... 2
- 3. Projects View in Dexi vs. Folder Structure in Mozenda** ..... 2
  - 3.1 Dashboard in Dexi vs. All Agents View in Mozenda** ..... 3
- 4. Building Robots in Dexi vs. Building Agents in Mozenda** ..... 3
  - 4.1 Main Differences** ..... 3
  - 4.2 Building Extractors vs. Building Agents** ..... 4
    - 4.2.1 Extractor Editor vs. Agent Builder** ..... 4
    - 4.2.2 Steps vs. Actions** ..... 5
    - 4.2.3 Network Filters vs. Navigation Requests** ..... 5
    - 4.2.4 Captcha Solver vs. Resolve Captcha** ..... 6
    - 4.2.5 Proxies in Dexi vs. Premium Harvesting** ..... 6
  - 4.3 Building Pipes vs. Building Sequences** ..... 7
    - 4.3.1 Main Differences** ..... 7
    - 4.3.2 Execute Robots with Pipes vs. Running an Agent with Sequences** ..... 8
    - 4.3.3 Data Collecting vs. Data/Jobs Settings** ..... 8
    - 4.3.4 Error Handling in Pipes vs. Error Handling in Sequences** ..... 8
- 5. Running a Configuration vs. Running an Agent Using a Job** ..... 10
  - 5.1 Main Differences** ..... 10
  - 5.2 Schedules vs. Schedule Agents/Sequences** ..... 10
  - 5.3 Triggers vs. Sequences** ..... 10
- 6. Datatypes & Datasets vs. Collections** ..... 11
- 7. Public APIs vs. RestAPIs** ..... 12
- 8. Accounts, Subscriptions, and Payments in Dexi vs. Mozenda** ..... 12



# 1. Introduction

Both Mozenda and Dexi are powerful web data extraction and automation platforms. While feature terms and interfaces may differ, the fundamental functionalities are closely aligned. Mozenda's Agents and Sequences function similarly to Dexi's Extractor and Pipes Robots, respectively. Additionally, Mozenda's support for various data formats and extensive APIs make it a strong alternative for users of Dexi.

If you are currently in the first steps of migrating from Dexi to Mozenda, please consider the below similarities and suggestions that can facilitate an easier migration and leverage the strengths of Mozenda.

Below you will find functionalities of each platform to be closely aligned and serve the same purpose. This guide offers a mapping of functionalities and key components to assist in understanding how to achieve the same scope when migrating platforms.

## 2. Key Differences Between Platforms

**Dexi is a web-based platform**, while Mozenda requires downloading a Windows-based Agent Builder. Dexi can be accessed and used from any existing browser without installation, while **Mozenda will require an installation** and initial configuration.

**Dexi uses Robots** with modular steps, enabling technical users to build complex workflows by connecting individual actions like data extraction, navigation, and transformation giving technical users flexibility. **Mozenda uses Agents with point-and-click capture logic**, which is simpler for non-technical users to use as it visually identifies data patterns and components on a webpage itself, without requiring manual setup of each step.

Dexi uses **an interface for designing and creating robot flows**. Mozenda uses the **Agent Builder (Windows application) for Agent creation** and a Web Console for monitoring and maintaining agents.

Link to Dexi Glossary: <https://help.mozenda.com/dexi/docs/glossary-2>

Link to Mozenda Glossary: <https://help.mozenda.com/docs/definitions>

## 3. Projects View in Dexi vs. Folder Structure in Mozenda

ItemID	Name	Config	Status	AgentBuilderVersion	Created	CreatedBy	LastRunTime	LastRunResult	Pro
1307	AtlasProf-02		Ready	7.14.106	2022-05-20 23:59:44	Hoxholli, Aldo			US\$
1306	AtlasProf-01		Ready	7.14.103	2022-05-20 23:50:58	Hoxholli, Aldo			US\$
1297	AGRL-3t-02		Ready	7.7.164	2022-04-15 10:46:35	Hoxholli, Aldo			US\$
1296	AGRL-3t-01		Ready	7.7.164	2022-04-15 10:32:23	Hoxholli, Aldo			US\$
1266	Shorterm-02		Ready	7.7.163	2022-04-02 02:33:08	Hoxholli, Aldo			US\$

### Folders



**Projects View** is the folder structure in Dexi where users organize assets like robots, datasets, datatypes etc. (similar to a folders structure in a personal computer).

**Folder structure** in Mozenda’s Web Console is where users organize all agents and collections according to purpose or domains. Users can switch between Agents and Collections views in each folder.

Link to Folders in Mozenda:

<https://help.mozenda.com/docs/add-agents-and-collections-to-a-folder>

### 3.1 Dashboard in Dexi vs. All Agents View in Mozenda

ItemID	Name	Config	Status	AgentBuilderVersion	Created	CreatedBy	Last	...
1573	Report-Test-D		Ready	9.18.101	2025-05-23 11:53:10	Derguti, Roxhers	2025-05-23 11:53:10	
1574	Report-Test-2-D		Ready	9.18.101	2025-05-23 11:53:10	Derguti, Roxhers	2025-05-23 11:53:10	
1575	Report-Test-3-W		Ready	9.18.101	2025-05-23 11:53:10	Derguti, Roxhers	2025-05-23 11:53:10	
1576	Report-Test-4-M		Ready	9.18.101	2025-05-23 11:53:10	Derguti, Roxhers	2025-05-23 12:04:54	Done
1572	TestingRock		Ready	9.18.100	2025-05-17 01:21:03	Hoxholli, Aldo	2025-05-17 01:26:10	Done
1571	CheckTickets		Error	9.16.102	2025-05-07 15:14:04	Hoxholli, Aldo	2025-05-15 12:03:17	Done
1570	Armend-Test		Ready	9.16.102	2025-04-29 14:45:37	Hoxholli, Aldo	2025-05-07 15:37:43	Canceled
1569	Testing		Ready	9.16.100	2025-04-08 15:40:12	Hoxholli, Aldo	2025-04-08 15:47:07	Done
1568	Ukraine Secretariat of the Cabinet - Ald		Ready	9.16.100	2025-04-08 10:33:45	Hoxholli, Aldo	2025-05-07 20:03:17	Canceled
1567	Ukraine Secretariat of the Cabinet - MI		Ready	9.16.100	2025-04-08 10:31:27	Hoxholli, Aldo	2025-04-08 10:59:11	Done
1566	TEST-ALDO-R		Ready	9.14.100	2025-04-04 13:47:25	Derguti, Roxhers	2025-04-04 13:47:56	Done
1565	SafeWay-01-Departaments-Native		Ready	9.14.100	2025-03-18 15:08:24	Hoxholli, Aldo	2025-03-18 15:09:11	Canceled
1564	XML-Parser		Ready	9.14.100	2025-03-17 15:48:32	Hoxholli, Aldo		F
1563	A-B-TestAgentGroupFields		Ready	9.11.117	2025-03-04 00:23:12	Hoxholli, Aldo	2025-03-04 00:23:35	Done
1559	walmartLogicTest-justToSave		Ready	9.10.106	2025-02-27 23:05:24	Hoxholli, Aldo		F
1558	CheckingSafeway		Ready	9.16.100	2025-02-24 16:27:46	Hoxholli, Aldo	2025-04-15 13:51:51	Done
1547	SafeWay-Native-Paraination-Test		Ready	9.14.100	2025-02-19 15:41:14	Hoxholli, Aldo	2025-03-17 12:15:08	Canceled

All Agents View

**Dashboard** in Dexi is the main page where users can manage and monitor recent executions and divide them into running, recently completed and scheduled.

**All Agents View** in Mozenda is where users have a full view of all the agents with different available filtering likes status, created, created by, proxies etc. Users can also create and edit custom views.

## 4. Building Robots in Dexi vs. Building Agents in Mozenda

### 4.1 Main Differences

There are 4 types of robots in Dexi – Extractors, Pipes, Crawlers and Autobots. Extractors and Pipes are the most advanced and commonly used.

**Extractor** – extractor robots are our most advanced robots. They allow users to choose every action the robot needs to perform including filling out forms, clicking buttons and extracting screenshots.



**Pipes** – a pipe robot is a super robot. Pipe robots can control other robots and can pull in external information from API's, databases and similar. Pipe bots do not extract data from websites themselves but combine other robots, API's and data sets to make a single flow for data extraction and processing.

**Crawler** – crawlers are less intelligent robots than scrapers – they simply visit each link found from the starting URL.

**Autobot** – autobots accept a URL as input and then map that URL to a list of extractors for a range of sites.

**Mozenda uses Agents**, which work similarly to the different types of Dexi robots. Agents can navigate web pages, automate browser tasks, and harvest data. Agent scripts are built to run with a set of instructions and do almost anything a human can do, but much faster.

## 4.2 Building Extractors vs. Building Agents

**Extractors** extract data from web pages, and either publish to a 3rd party application; or store in a Dataset. Extractors are created and edited through the Extractor Editor.

**Agents** collect data from websites and store the data in Collections. Use the Agent Builder to create a new Agent or to modify an existing Agent.

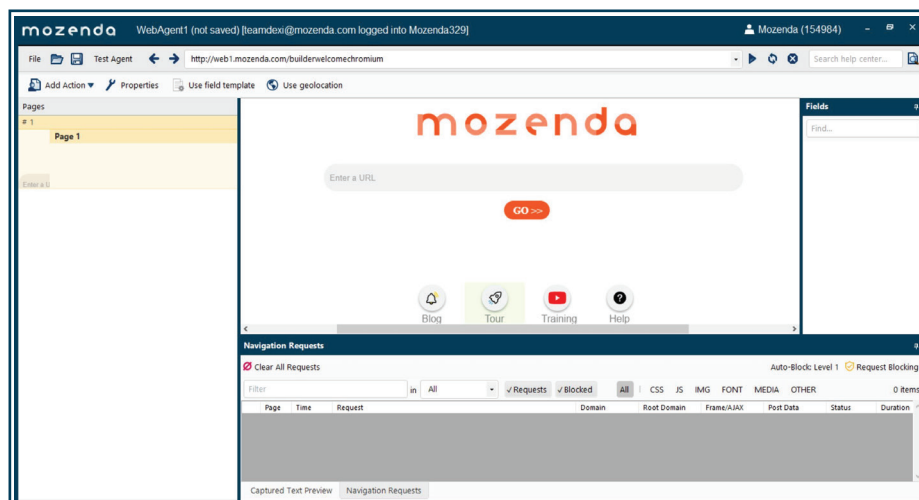
Link to Building an Agent: <https://help.mozenda.com/docs/build-an-agent>

### 4.2.1 Extractor Editor vs. Agent Builder

**The Extractor Editor** is used to build, update, and test scraping robots. The main parts of the Editor are the preview area, the suggestions box, the timeline and the menu.

**Agent Builder** is a downloadable application used to build, design, and test Agents. After the Agent is created, it is saved to the Web Console and runs in the Mozenda servers using jobs (instead of workers).

Link to the Agent Builder: <https://help.mozenda.com/docs/build-a-simple-agent>



Agent Builder



## 4.2.2 Steps vs. Actions

**Steps** are actions a scraper robot performs upon visiting a Web page that simulate real-world human interaction with the page. They provide users with a view of what is happening and when it is happening. Steps can also be grouped

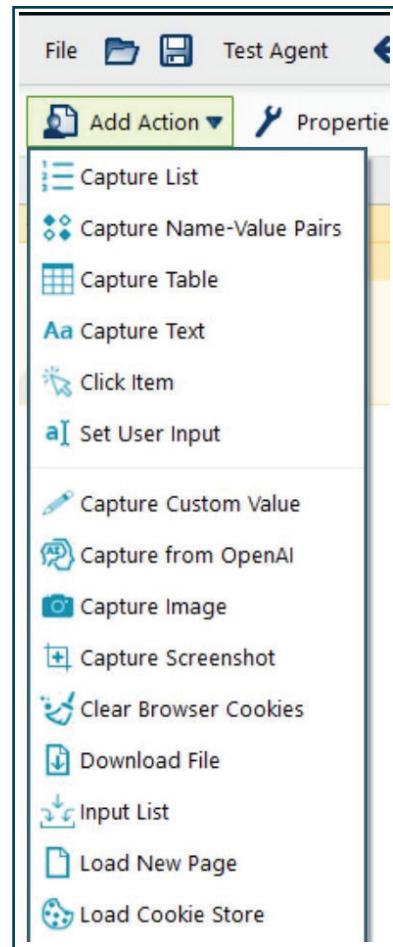
**Actions** are tasks an agent can perform to simulate human interactions on a site. They provide users with the ability to scrape different elements on the page. When extracting data, Mozenda automatically creates a XPath, it is often needed to implement more accurate XPath or as websites page structure often change. XPath's help ensure the agent is guided to collecting the correct data.

Link to Actions:

<https://help.mozenda.com/docs/build-an-agent>

## 4.2.3 Network Filters vs. Navigation Requests

**Network filters** can be applied in Dexi under the Network tab in the robot editor. They disable network filters that might block requests. Network filters are used for social media and advertising networks, tracking and analytics providers, chat and support providers, fonts, and third party cookies management services.

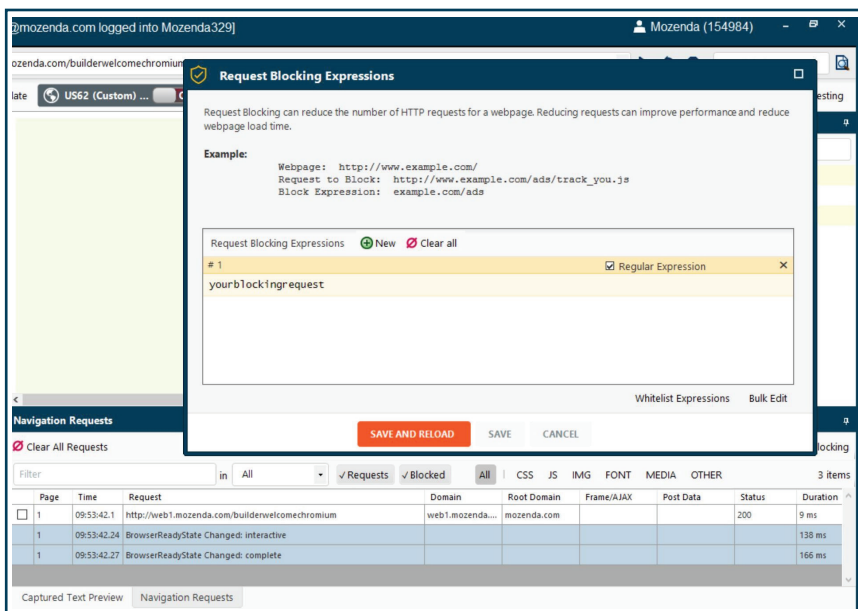


Agent Actions

**Request Blockers** are instead used in Mozenda. The Request Blocker is under the Navigation Request tab, which allows users to speed up agent processing and efficiency by blocking domain requests unrelated to page content. Mozenda supports 4 distinct levels of request blocking.

Link to the Request Blocker:

<https://help.mozenda.com/docs/how-to-customize-your-auto-request-blocking>



Request Blocking



## 4.2.4 Captcha Solver vs. Resolve Captcha

**Captcha Solver** feature in Dexi is done through the Element panel and requires two element paths: the Captcha image and input field.

**Resolve Captcha** in Mozenda is configured in the Web Console. Users can choose the Captcha Solver provider and then add it to specific agents through the Agent Builder, where users can add a Resolve Captcha action with different parameters. Mozenda also provides a system build captcha solver for Press & Hold captchas.

Link to Resolve Captcha: <https://help.mozenda.com/docs/resolve-captcha>

**Resolve Captcha**

Use a **Resolve Captcha** action to solve different types of user challenges encountered on websites. Your access credentials need to be configured for each of the different providers you wish to use. Click [here](#) to configure them for your account. **Not all providers can solve all types of captchas.** Please reference the solvers' websites for up-to-date information concerning which types of captchas are currently supported for each provider.

Name

Type

Solver

Solver Timeout  seconds

Max Tries

**Additional information:**

The Resolve Captcha action is an advanced feature. The system will try and automatically detect which type of captcha is on the page. But, in many cases, the type will need to be manually selected by the user. In addition, most resolve captcha actions will require the user to improve the xPaths that the system will use to resolve the captcha. If no xPaths are entered then the system will attempt to automatically find the correct DOM elements.

Resolve Captcha

## 4.2.5 Proxies in Dexi vs. Premium Harvesting

**Proxies** are set in Dexi on account level and can be used in the Robot or configuration to allow a more organic simulation of traffic and to avoid being blocked.

**Proxies in Mozenda** are also created on account level. Using Premium Harvesting settings, users can configure an agent to pull from a pool of region-specific IP addresses to get data quickly with less detection.

Link to Premium Harvesting: <https://help.mozenda.com/docs/manage-proxies>



Harvesting Settings
✕

Method
Behavior
Error Handling
Advanced

**Run this agent using**

Premium harvesting

**Run this agent in**

US62

**Note**

Modifying where this agent runs may cause some of the agent's actions to not find items on the page if the website changes its content to reflect the new location.

SAVE

CANCEL

Premium Harvesting

### 4.3 Building Pipes vs. Building Sequences

#### 4.3.1 Main Differences

**Pipes** in Dexi are super robots that integrate web data extraction and web data processing into a single seamless workflow.

**Sequences** are workflows for automating and scheduling common web scraping tasks, including clearing collections and views, running multiple chaining agents, and publishing data.

ItemID	Name	Status	Config	Progress	NextRunTime	LastRunTime	LastRunResult	Created	CreatedBy
1149	TestingIps	Running		Running TestingProximizeUS2 (1552)	2025-05-27 09:00:00	2025-05-26 09:04:15	Done	2025-02-04 16:06:23	Hoxholl, Aldo
1056	MerjjeReport	Error		Running MerjjeReport (1173)	2025-05-26 17:30:00	2025-05-25 17:31:21	Done	2022-01-05 21:36:39	Hoxholl, Aldo
1055	MerjjeRenew	Error		Running MerjjeRenew (1174)	2025-05-26 10:00:00	2025-05-23 16:01:57	Done	2022-01-05 21:34:23	Hoxholl, Aldo
1147	publishingTest	Ready			2025-05-19 15:15:16	2024-12-03 15:10:35	Done	2024-09-02 13:57:55	Hoxholl, Aldo
1152	TestingRock	Ready			2025-05-17 01:26:01	2025-05-17 01:25:46	Done	2024-07-25 12:20:20	Hoxholl, Aldo
1151	SafeWay-Native	Ready			2025-03-24 01:57:44	2025-02-14 19:14:17	Canceled	2024-06-14 13:21:21	Hoxholl, Aldo
1150	TestAni	Ready				2025-02-13 00:39:42		2024-05-14 16:21:57	Hoxholl, Aldo
1148	Sofia test expor...	Ready				2025-01-16 14:01:30		2024-04-05 11:05:16	Hoxholl, Aldo
1145	TestAldoIP	Ready				2024-09-26 11:23:38			Dexi, Team
1144	GiantFood	Ready				2024-09-24 22:51:45			Hoxholl, Aldo
1143	Publishing	Ready				2024-09-02 13:57:55			Hoxholl, Aldo
1142	testBug	Ready				2024-07-25 12:20:20			Hoxholl, Aldo
1141	iherbTest	Ready				2024-06-14 13:21:21			Hoxholl, Aldo
1140	AAA	Ready				2024-05-14 16:21:57			Hoxholl, Aldo
1139	LexMachina	Ready				2024-04-05 11:05:16			Hoxholl, Aldo

Sequences



### 4.3.2 Execute Robots with Pipes vs. Running an Agent with Sequences

**Execute Robots with Pipes** supports the execution of multiple extractors and/or crawlers, connecting outputs to inputs. Pipes don't need to wait for the 1st robot to end before the 2nd robot begins.

**Running Agents with Sequences** allows agents to run one right after the other. If an agent has a list, the list can be split and ran concurrently. Additionally, multiple agents may be run in parallel.

Link to Create Sequence: <https://help.mozenda.com/docs/en/sequences-1>

#	Step	Config	Status	Progress
1	Run TestingRock		Ready	
2	Publish 100MileZip	2	Ready	
3	Export 100MileZip (A) into 100MileZip		Ready	
4	Clear 100MileZip		Ready	
5	Run sequence Publishing		Ready	

Sequences Steps

### 4.3.3 Data Collecting vs. Data/Job Settings

**Data collecting with Pipes** can be used as an input for other bots while simultaneously gathering information. This method configures robots to perform specific tasks based on conditions set.

**Using Data/Job Settings** with Sequences in Mozenda, the input for the next Agent (process) is already defined within the Agent Builder. Therefore, when creating a Sequence, users simply specify the first process to run, followed by the second process. The sequence will then execute the steps one after the other in the defined order.

### 4.3.4 Error Handling in Pipes vs. Error Handling in Sequences

**Error handling in Pipes** e.g. When one robot errors, it doesn't disrupt any other bots from continuing their individual process. Pagination runs smoothly but delays the QA process as it does not notify users when an error occurs.

**Error handling in Sequences** stops the Agent as soon as an error is encountered. Users can increase the number of retries when an error occurs in the process. Users can also select specific errors for the job to try again.



**Lists** **Error Handling** **Parameters**

Do not automatically resume jobs that stop with an error

Automatically resume jobs that error  times

Only for the following error codes:

2 selected error codes

<input type="checkbox"/>	Name	Description
<input checked="" type="checkbox"/>	CfxNameNotResolved	The website was unreachable because of a name lookup error
<input checked="" type="checkbox"/>	CfxUnhandledError	The website was unreachable because of an unknown error.
<input type="checkbox"/>	ClickElementFailed	A click action could not be performed on the page.
<input type="checkbox"/>	CriticalElementNotFound	An item could not be found on the page.
<input type="checkbox"/>	DownloadJavaScriptFailed	An external JavaScript file failed to load.

**SAVE** **CANCEL**

Error Handling (a)

**Run Agent**

Agent ID: 1534 **CHANGE**

Name: MerrjepReport - Sofia test

Agents that do not use request blocking will automatically use **Request Blocking Level 1 - Standard**

**Lists** **Error Handling** **Parameters**

Do not automatically resume jobs that stop with an error

Automatically resume jobs that error  times

Only for the following error codes:

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	CfxNameNotResolved	The website was unreachable because of a name lookup error
<input type="checkbox"/>	CfxUnhandledError	The website was unreachable because of an unknown error.
<input type="checkbox"/>	ClickElementFailed	A click action could not be performed on the page.
<input type="checkbox"/>	CriticalElementNotFound	An item could not be found on the page.
<input type="checkbox"/>	DownloadJavaScriptFailed	An external JavaScript file failed to load.

**SAVE** **CANCEL**

Error Handling (b)

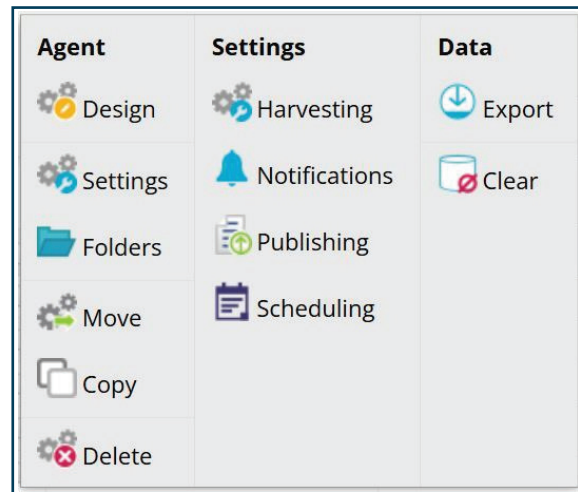


## 5. Running a Configuration vs. Running an Agent Using a Job

### 5.1 Main Differences

**Configurations in Dexi** are created to "Run" a robot and define data inputs, scheduling, error handling, proxies etc. Users can add multiple customizations to the configuration of a Robot.

**Agent Settings in Mozenda** are settings that a user applies before running an agent using a "Job". Users can set multiple customizations like harvesting, notifications, publishing and scheduling.



Mozenda Agent Settings

### 5.2 Schedules vs. Schedule Agents/Sequences

**Scheduling of robots in Dexi** is supported in the configuration settings.

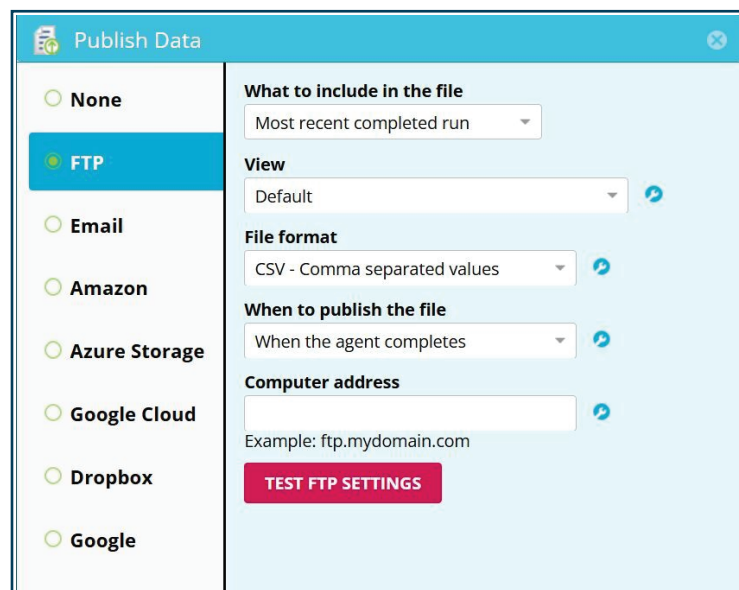
**Scheduling Agents and Sequences in Mozenda** is handled via the Schedule Agent feature, and can be configured to run one time or at specific intervals.

Link to Schedule Agents: <https://help.mozenda.com/docs/en/run-an-agent-on-a-schedule>

### 5.3 Triggers vs. Sequences

**Triggers in Dexi** are used on robots for automating workflows. Triggers can add rows to a dataset or publish to a third-party app, etc..

**Triggers in Mozenda** – while Mozenda does not have a specific 'Trigger' feature, most functionality can be configured with Agent Settings and Sequences Steps.



Publish Data



## Agent Settings

- A- Publishing -> When the agent finishes, it will automatically publish the file where is set.
- B- Notification -> When Agent is started/stuck/finished users can select to receive an email.

**Sequences** – examples of sequence steps include:

- Run Agent - The agent will begin to scrape data according to its configuration.
- If applicable, you can divide an agent's Data List inputs among multiple concurrent jobs and add additional error handling.
- Clear Collection - Clear the contents of a collection.
- Delete View Items - Specify certain data from a scrape and delete it.
- Publish Collection Data - Send your data to a specific location and format
- Update Field Value - Update the values of a field.

Link to Sequences: <https://help.mozena.com/docs/en/sequences-1>

## 6. Datatypes & Datasets vs. Collections

**Data types in Dexi** give users a way of defining the desired data structure.

**Datasets** are used in Dexi to store data permanently and support deduplication, triggers, and advanced key definitions.

**Collections in Mozena** are tables used to store data. Each agent has a collection where it saves the data. Type of Collections include:

**Standard Collections** - created manually or by a file import.

**Agent Collections** - created at the time users create a new agent.

**Combined Collections** - combine data from multiple agent collections.

Link to Collections: <https://help.mozena.com/docs/en/what-are-collections>

Link to Combined Collections: <https://help.mozena.com/docs/en/combined-collection>

ItemID	Name	Created	Description	Modified
1031	100MileZip	2019-08-28 10:23:15	Sample description of the Collection.	
1006	100MileZipList	2019-08-08 10:38:13	Sample description of the Collection.	
1321	1010PWC	2022-02-18 14:53:46	Sample description of the Collection.	
1777	10MileZip-2	2024-09-12 14:49:32	Sample description of the Collection.	
1088	10MileZipList-D...	2020-01-20 09:47:43	Sample description of the Collection.	
1327	2466PWC	2022-03-01 16:35:23	Sample description of the Collection.	
1030	250MileZipList	2019-08-28 09:25:07	Sample description of the Collection.	
1096	25Mile-Default	2020-01-22 10:58:15	Sample description of the Collection.	2020-01-22 10:58:18
1090	5MileZip-Default	2020-01-20 10:32:34	Sample description of the Collection.	
1694	7Eleven-Token	2024-06-10 10:15:13	Sample description of the Collection.	
1800	A-125	2024-10-16 11:55:06		
1706	AA	2024-06-23 23:57:46	Sample description of the Collection.	2024-06-23 23:57:46
1776	aaaaaaaaaaaa	2024-09-11 17:12:25	Sample description of the Collection.	
1387	aaeTest-01a	2022-04-08 09:17:00	Sample description of the Collection.	

Collections



## 7. Public APIs vs. Rest APIs

**Dexi** offers various public **API access** to manage robots and data.

**Mozenda** offers a robust **REST API** to Enterprise and Professional customers. The Rest API in Mozenda allows users to manage Agents, Sequences, and data collections without having to manually access the Web Console.

Users may need account system privileges to access the WebServiceKey to utilize REST API.

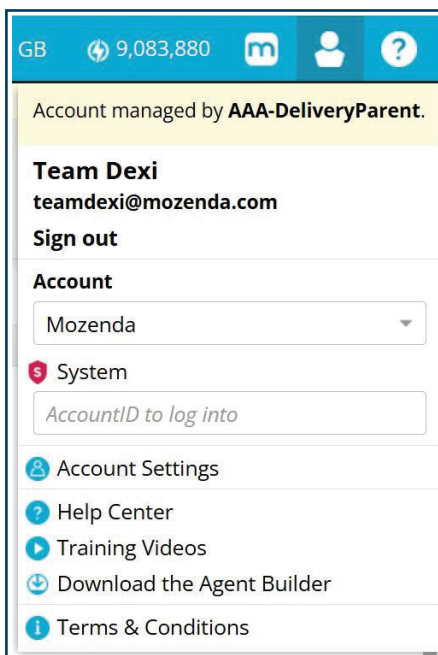
Link to Mozenda REST API: <https://help.mozenda.com/docs/rest-api-introduction>

## 8. Accounts, Subscription, and Payments in Dexi vs. Mozenda

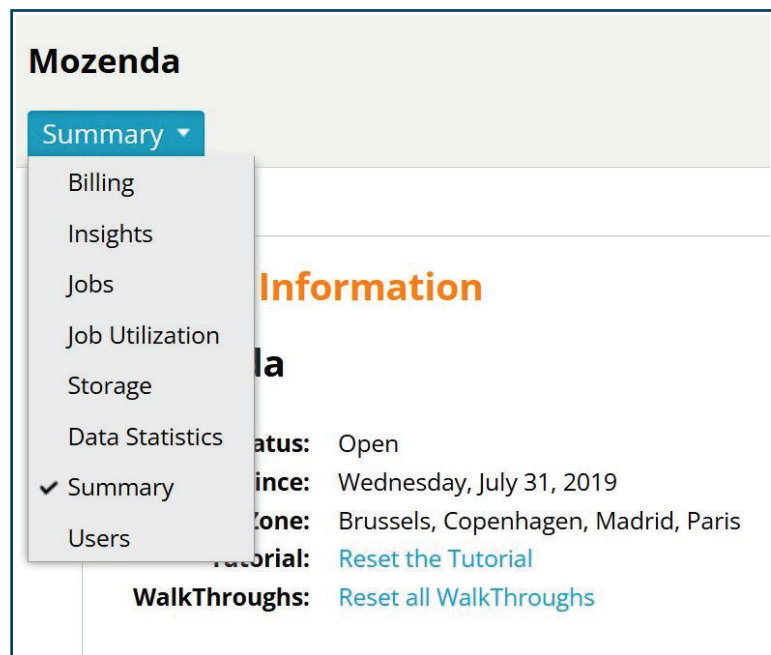
**Account details in Dexi**, like subscription and billing, can be accessed through the Dexi platform directly under Settings. Dexi uses workers to determine the capacity of your account.

**Account details in Mozenda**, like subscription and billing, can be accessed through Account Settings. Here users can edit user permissions and add additional users and departments. Mozenda also uses subscription-based credits for concurrent jobs, instead of workers.

Link to Processing Credits: <https://help.mozenda.com/docs/how-do-processing-credits-work>



Account Settings



Account Summary