

Zuva KTA Connector v1

Pocket Guide

December 2023
Version 1.0

TUNGSTEN
AUTOMATION

Table of Contents

What is the Zuva KTA Connector?	3
Technical Assets included in the Package.....	3
Why Kofax Marketplace?	3
How do I download the Zuva KTA Connector from Marketplace?	4
How does the Zuva KTA Connector work?	4
Architecture of the connector	5
KTA import	5
Getting started	5
Connector Methods.	6
Zuva Methods.....	6
Post-Zuva methods	7
Frequently Asked Questions?	9
Storage	9
Detection	10
Who do I need to contact to get the Zuva API private keys or user credentials?.....	11

What is the Zuva KTA Connector?

Zuva/DocAI is an API-driven platform that enables you to implement contracts AI into your applications, without requiring development from the ground-up. Zuva brings their core business expertise, which is the extraction/indexing of legal documents and has a team of experienced lawyers & para-legals which aid in training the AI extraction models.

Zuva AI fields were built and trained by experts over the past 10+ years. Over 1,300+ out of the box AI fields that are built and trained by subject matter experts. Fields identify and extract common legal clauses, provisions and data points from unstructured documents and contracts, including ones written in non-standard language.

The Zuva KTA Connector from Tungsten Automation Labs is a KTA Package that provides the assets you need to connect to Zuva's API using best practices. This connector is designed to be easily installed and up and running to test out a few prepackaged Use Cases for Contracts:

Contracts Risk Scoring – Using KTA to process Contract documents, the Zuva connector for Total Agility includes a pre-built Business Process to review Contracts and flag contract documents that have been identified as having a risk score of high based on preset business rules.

Technical Assets included in the Package

The Kofax Labs Zuva Connector consists of a .Net Framework v4.8 DLL (Kofax.ZuvaConnectorV1.dll) which will aid in obtaining and presenting the information extracted by Zuva through the Connector. The connector also includes several functions in order to return data as well as Zuva's normalizations.

Why Kofax Marketplace?

Kofax Marketplace is a digital showcase of assets connecting customers with pre-built, integrated components and solutions to extend their digital workforce. The Zuva KTA Connector will be easily available on Marketplace, built on top of Kofax's Intelligent Automation platform. It accelerates customer automation journeys and reduces the need for custom development work. The Kofax Marketplace also allows partners to expose their value-added solution to new prospects and customers.

How do I download the Zuva KTA Connector from Marketplace?

The solution is available at <https://marketplace.kofax.com/> and downloadable from... It comes with a detailed readme file which describes how to set up the solution in KTA.

How does the Zuva KTA Connector work?

The Zuva KTA Connector handles all REST calls into the Zuva API, including:

Upload document file to Zuva.

The Zuva KTA Connector will convert the file to the PDF format and upload it to the Zuva/DocAI platform.

Get document classification information from Zuva.

The Zuva KTA Connector will contact the Zuva/DocAI platform and request the document uploaded classification information.

Send extraction request to Zuva.

Once the file upload has been confirmed, the Connector will make another call to the REST API by creating an extraction request for a list of pre-defined fields. For a list of fields and which documents can they be used on, please visit [Zuva Dashboard](#).

Retrieve extraction results from Zuva.

After the extraction request has been submitted to Zuva, the Connector will wait until the extraction request is completed and will then retrieve all extraction results for all field IDs sent on the request.

Output the extraction results to the calling process.

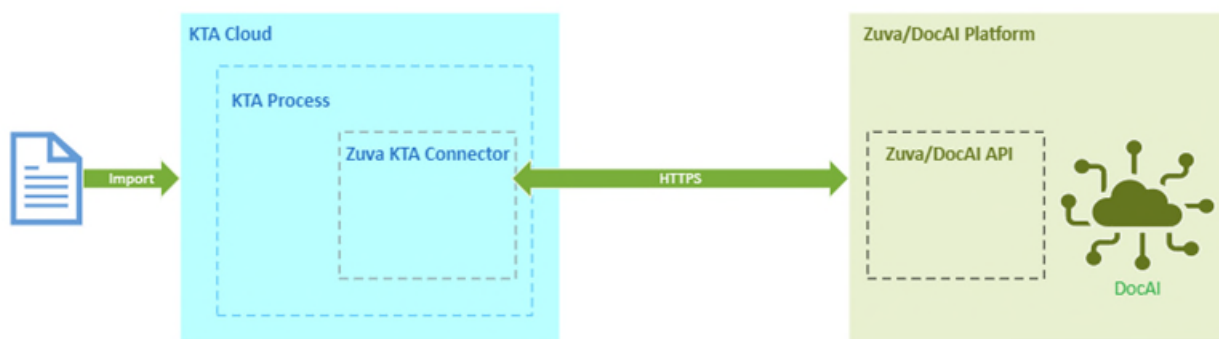
When the extraction results are retrieved, the JSON is turned into a data model variable which is returned to the calling process as an output variable; this allows all the extraction information (OCR, text location, extracted text) to be used.

Architecture of the connector

KTA connects with the Zuva/DocAI services through their API ([DocAI API Reference – Zuva DocAI API Reference](#)) within the KTA workflow to automate the customer experience without any hassle of setting up or configuring any additional steps, subprocesses or the JSON data models.

The below diagram shows

- The documents received via multi-channel are ingested by KTA
- KTA calls the Zuva/DocAI API to send the file & JSON payload over a secured channel
- The result is received in a form of JSON
- The Zuva Connector will make sense of the JSON and will return the extracted text.



KTA import

The documents can be ingested by KTA through different means (email, shared folder watch, scan client). The Zuva KTA Connector is designed to handle one document at a time, so if you have multiple files or a folder setup, it is best to call the Connector within a document iteration loop.

Getting started

Before you get started with the Zuva connector, please visit their website ([Embeddable contracts AI that's dead simple to use. | Zuva](#)) and sign-up for an account; once your account is created, you can get a token that you will need in order to send files & extraction requests. For more information on this, please visit Zuva's Quick Start guide ([Quick Start | Zuva](#)).

After you have the Zuva token, please follow the steps below:

1. Follow the steps to install the Zuva Connector Package to a KTA instance.

2. Import the Kofax.ZuvaConnectorV1.dll into your KTA instance.
3. To call the connector from a workflow activity, create a .Net activity and configure the method as desired.
4. The first function you would need to call is to upload the file to the Zuva DocAI service; this will return a file ID.
5. Using the resulting file ID you can then request the document to be classified, request a field extraction or answer.

Connector Methods.

Tungsten Automation Labs' Zuva Connector exposes the methods to assist you in getting the information you need from Zuva. The following is a list of methods:

Zuva Methods

The following methods exposed in the connector interact with the Zuva DocAI API:

1. ZuvaUploadFileBase64String: this method uploads a base64-encoded string variable as a file, and returns the Zuva file ID.
 - a. base64Doc: base64-encoded string file.
 - b. zuvaToken: The token provided by Zuva.
2. ZuvaUploadFileKTASDK: this method would get the document.instance ID, use the TotalAgility SDK to get it, then upload it to Zuva. Zuva will then return the file ID:
 - a. docID: TotalAgility Document.InstanceID.
 - b. ktaSDKUrl: URL path to the TotalAgility SDK to be used.
 - c. sessionID: The TotalAgility session ID to be used.
 - d. zuvaToken: The token provided by Zuva.
3. ZuvaGetClassificationResult: this method requests the uploaded document's classification by Zuva. It returns the classification name.
 - a. zuvaFileID: the file ID obtained from the upload method.
 - b. zuvaToken: The token provided by Zuva.
4. ZuvaGetExtractionResult: this method creates an extraction request to Zuva, using the file ID obtained from the upload method. It waits for the extraction to complete from Zuva, then it would return the resulting JSON string.

- a. zuvaFileID: the file ID obtained from the upload method.
 - b. zuvaFieldList: List of comma-delimited field ID GUIDs from Zuva.
 - c. zuvaToken: The token provided by Zuva.
- 5. ZuvaGetAnswerResult: this method uses the Zuva Answers feature; it creates an Answers request to Zuva using the file ID from the upload method. It waits for the extraction to complete from Zuva and return the resulting JSON string.
 - a. zuvaFileID: the file ID obtained from the upload method.
 - b. zuvaAnswerID: Single answer field ID GUIDs from Zuva.
 - c. zuvaToken: The token provided by Zuva.
- 6. ZuvaExtractAndConcat: this is a 2-in-1 method. It first does an extraction request to Zuva and then looks at the resulting JSON string and concatenates all extracted text with the string being passed. The result is a "--" delimited string containing all extracted text.
 - a. zuvaFileID: the file ID obtained from the upload method.
 - b. zuvaFieldList: List of comma-delimited field ID GUIDs from Zuva.
 - c. zuvaToken: The token provided by Zuva.
 - d. stringToConcat: String to be concatenated.

Post-Zuva methods

The following methods exposed in the connector are to be used after the

- 1. GetAnswerResultsForField: this method inputs the resulting JSON from calling the ZuvaGetAnswersResult method and return a simplified JSON string containing the selected answer and supporting paragraphs.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 2. ConcatExtractionResultsForField: Concatenates ("||") all text extracted from the results for the specified Field ID.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 3. GetFirstFromExtractionResultForField: Gets the first instance of extraction results for the specified Field ID.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.

- b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 4. GetLastFromExtractionResultForField: Gets the last instance of extraction results for the specified Field ID.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 5. GetFirstNumberFromExtractionResultForField: Gets the first instance of at least 4 digits in the extraction results for the specified Field ID.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 6. GetLastNumberFromExtractionResultForField: Gets the last instance of at least 4 digits in the extraction results for the specified Field ID.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 7. GetSumOfAllNumbersFromExtractionResultForField: Gets the sum of all instances of at least 4 digits in the extraction results for the specified Field ID.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 8. GetFirstDateFromExtractionResultForField: Gets the first date instance in the extraction results for the specified Field ID in “MM/dd/yyyy” format.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 9. GetLastDateFromExtractionResultForField: Gets the last date instance in the extraction results for the specified Field ID in “MM/dd/yyyy” format.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 10. GetEarliestDateFromExtractionResultForField: Gets the earliest date instance in the extraction results for the specified Field ID in “MM/dd/yyyy” format.
 - a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
- 11. GetLatestDateFromExtractionResultForField: Gets the latest date instance in the extraction results for the specified Field ID in “MM/dd/yyyy” format.

- a. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - b. extractionResut: Zuva raw extraction results returned from a Zuva method.
12. GetNormalizationFieldFromExtractionResultForField: Gets the Zuva-normalized information for the given extraction result.
- a. jsonString: Zuva raw extraction results returned from the Zuva extraction field method.
 - b. zuvaFieldID: Method will only process extraction texts where it equals this parameter.
 - c. normalizationField: String containing the normalization value to be returned; valid values are “currencies”, “dates” & “durations”

Frequently Asked Questions?

Below is the list of possible questions gathered to answer the basic questions about storage and detection

Storage

Is data stored in the cloud? And how long is it stored?

Yes. To ensure maximum system performance and support flexibility, the Zuva/DocAI service is offered primarily via the cloud. Zuva DocAI is not intended as a long-term storage repository. Thus, uploaded documents expire and are removed from the system after 7 days by default. The expiration duration is also configurable, from 1 hour to 14 days, with the option of extension (as long as the file needs to be kept for further processing).

Also, you can choose to delete it before the expiration date, using the Files API. Once a document has been deleted from DocAI, it can not be restored.

How do you secure the transfer of the data? and what happens after the document is uploaded to the server?

Data transfer is handled via secure TLS connection. Each user is authenticated using an API key in the https header. Data is encrypted in transit and at rest. Transport Layer Security (TLS, protocol version 1.2) is utilized for encryption during transport. Zuva & KTA support high-grade 256-bit ciphers as well as ciphers providing forward secrecy. Once data reaches Zuva, it is processed and stored in a client-specific object store on a 256-bit encrypted disk.

How many documents can the system process simultaneously?

Although the connector is designed to handle one document at a time, this can be used in multiple processes/jobs to run in parallel and independently from one another. The only limit you have is the number of documents that can be processed daily.

Detection

What type of files do you support?

Although KTA supports all kinds of document files, Zuva requires PDFs for their process; the Connector includes a step so that all files are converted to PDF before they get uploaded to Zuva's API.

How long does it take to process a document?

Depending on the size of the document and number of fields to extract, it can take anywhere from 15 seconds to a couple of minutes.

What regions are supported?

Presently, Zuva only allows contracts from the European Union (EU), Japan & USA.

Are there any limits when it comes to the size of the document?

AWS S3 limits file sizes of up to 5GB. The only technical limitation for the document processing is the associated 15-minute timeout. Any processing that takes longer than this will not succeed.

Who do I need to contact to get the Zuva API private keys or user credentials?

Please visit the Zuva Website (<https://dashboard.zuva.ai/>) and sign-up; once you have your credentials registered, please obtain your Zuva API token to be used by the Connector. If you have any issues, please contact Zuva support by visiting <https://dashboard.zuva.ai/support>

Once the package is downloaded and imported in KTA, the private key can be replaced in the server variables section as described below:

Key	Value
ZUVA_API_TOKEN	Optional – Zuva API Token obtained after you have signed up for a Zuva account; this would be used globally in KTA for all Zuva calls.
ZuvaAuthHeader – process variable	Variable within the process to allow for multiple API Tokens for each call.
Do not change below, unless consulted or advised	
KTA_SDK_URL	Internal to KTA – URL for the KTA RESTful API SDK.
ZuvaExtractionResult	Data model which represents Zuva's extraction result JSON structure
ZuvaFieldExtractionRequest	Data model which represents Zuva's extraction request JSON structure