IBM Cloud Pak for Business Automation Demos and Labs 2024

Lab Guide – Automation Document Processing

V 1.0.0 (for CP4BA 24.0.0)

Clandis Baker
SWAT Business Automation Portfolio Specialist – Capture Products
bakercl@us.ibm.com

Krish Lakshminarayanan
Global Technical Program Leader for Capture / Intelligent Document Processing Global Sales (WW)
krishkrish@ibm.com

Ryan Sparks
Advisory Business Automation Tech Sales Leader – RPA/ADP
rmsparks@us.ibm.com

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is

Table of Contents

1 Overview

1.1 Icons

The following symbols appear in this document at places where additional guidance is available.

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step but is helpful or good to know.
	Trouble-shooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

1.2 Abstract

Set up a capture solution in minutes. Introduce technical sellers to IBM Automation Document Processing. In this session, students will configure their own capture project. They will learn how to use machine learning classification for their sample documents, define fields for extraction, create validation rules, and use deep learning (subject to environment configuration) to automate data extraction.

1.3 Introduction

Welcome to the Automation Document Processing lab. This lab will introduce you to Document Processing and provide you with an understanding how you can configure it for your customer opportunities.

Automation Document Processing provides a tailored solution that reads your documents (in English, French, Spanish, German, Dutch, Portuguese), extracts data, and refines and stores the data for use.

With the right business knowledge, you can design deep learning models without being a data scientist. The Document Processing Designer includes pre-trained deep learning models that you can use as a base for your own model. The pre-trained document types include bills of lading, invoices, and utility bills.

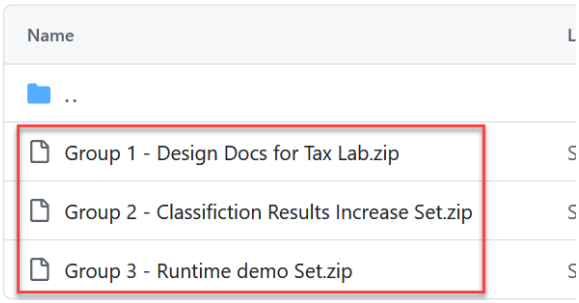
You can extract text, check boxes, forms, tables, barcodes, signature detection and even free text. With no or low code options, you can create an application that

processes documents, extracts data, flags issues, and stores your documents and data. And the data enrichment capabilities ensure that the extracted data is standardized and ready for use in downstream integrations.

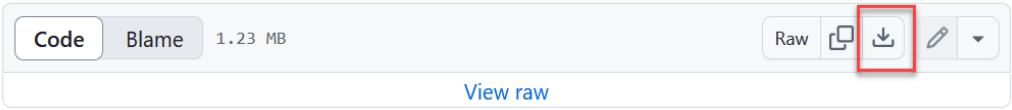
This lab will not cover all the available functionality available due to time constraints. It is intended as an entry point.

2 Getting started

1. If you are performing this lab as a part of an IBM event, access the document that lists the available systems and URLs along with login instructions. For this lab, you will need to access **IBM Business Automation Studio**.
2. **Download** the **sample documents** in the zip file. We will be using these sample documents: https://github.com/IBM/cp4ba-labs/tree/main/24.0.0/Document%20Processing/Lab%20Data



- **Click** on "Group1 – Design Docs for Tax Lab.zip"
- **Click** on the **Download raw file** icon



Repeat above steps **"Group 2 – Classification Results Increase Set.zip"** and **"Group 3 – Runtime Set.zip"**

- **Unzip** the files and keep them in their designated folder

  You will notice the images are in various unique folders that will be referenced specifically in the different labs later. Please keep them in their proper folders.

3 Lab overview

The lab will focus on the design time tasks for Automation Document Processing (ADP). Despite the push for the digitization of content for many years, there are still a lot of paper documents that require workers to read and interpret the information – whether it is structured data, such as tax forms, or semi-structured data, such as invoices, utility bills, and so on. This lab describes how to set up an automate document processing pipeline using ADP.

3.1 How does ADP work?

Document Processing Designer

You use the Designer interface to create a set of document types and related fields that comprise your Document Processing project. Document Processing Designer combines an intuitive interface with a set of AI and deep learning tools that identify and learn the document types that matter to your organization. For each document type, you designate which pieces of information to extract as data for that document to be used by downstream applications. You can also apply tools to clean up and standardize the data as it is extracted.

Deployment tools

After you build the Document Processing project in the Designer, you deploy the project to make it available for building your document processing application. The deployment process is also used to configure the repository to receive the processed documents from your end-user application by making the capabilities and artifacts available for integration into an application and into the destination repository.

Application templates and toolkits

You use the no- or low-code application building capabilities of Application Designer, customized templates and toolkits, and the AI model of your Document Processing project to create a document processing end-user application. This application recognizes your documents, extracts your relevant data, and presents issues to fix before sending the documents to storage and using the data in other systems.

Document processing application and document management

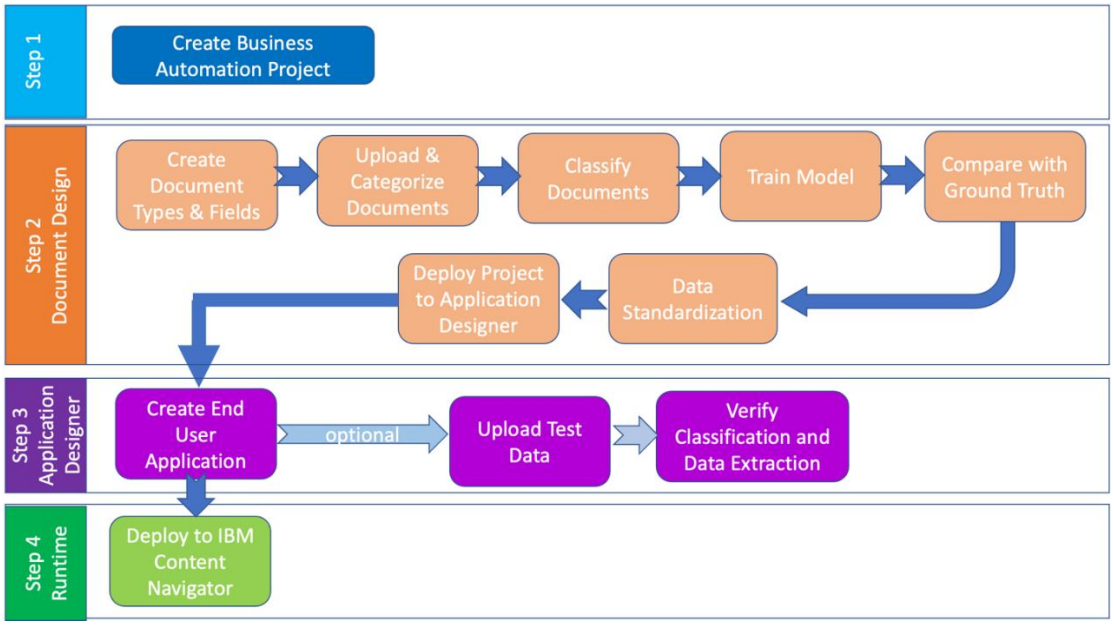
The application that you build uses AI and deep learning to automatically detect, extract, and standardize the data in all your documents. Any anomalies are flagged according to your customized model and the priority that you set so that your document processing user can correct issues before the documents are finalized.

When you deploy your document processing application, you connect it to a content repository that manages the document types and the extracted data for each document. The solution is fully integrated with IBM FileNet® Content Manager, simplifying document and data storage by applying your existing filing architecture and business rules to each processed document. The content and metadata are automatically saved in FileNet within the appropriate document class.

End result

Your document types are stored in the content repository, with appropriate retention and access controls. An associated JSON file reflects all the extracted data for the document. Properties are set on the document with the data definition-controlled values. Your extracted data is cleaned, standardized, and ready for use in other applications.

The following diagram shows the tasks required to configure and deploy a new ADP project.



Step1 – Create an ADP Business Automation Project

Each document processing project requires a separate repository in your Git organization. Coordinate with your Git administrator to create the repository for your project.

Step 2 – Document Design

This step shows the high-level tasks that will be needed to complete to train the system to recognize document types, successfully extract fields and tables, configure the fields in FileNet and finally deploying your ADP project to the application designer so you can configure the end-user interfaces.

Step 3 – Application Designer

The application designer is where you would configure end-user interfaces such as the classification and verification screens. The lab will not go in a lot of details on how to configure the interfaces. It will instead show you how to create an application, and test processing a batch of documents through the system. To get more information on creating/using the Business Automation Application (BAA) look at the Lab for Business Automation Application.

Step 4 – Runtime

End-users would be using the runtime IBM Content Navigator interface to process documents or batches, classify document and verify extracted field data in the verification screen.

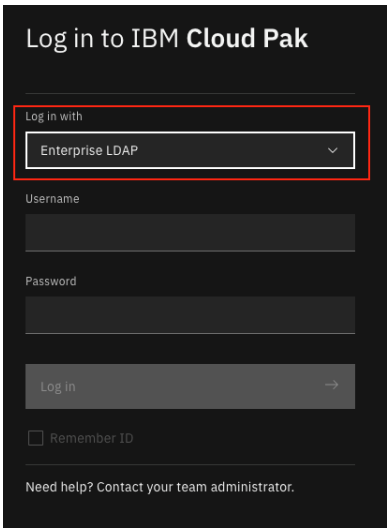
4 Create a Document Processing Project

Step 1

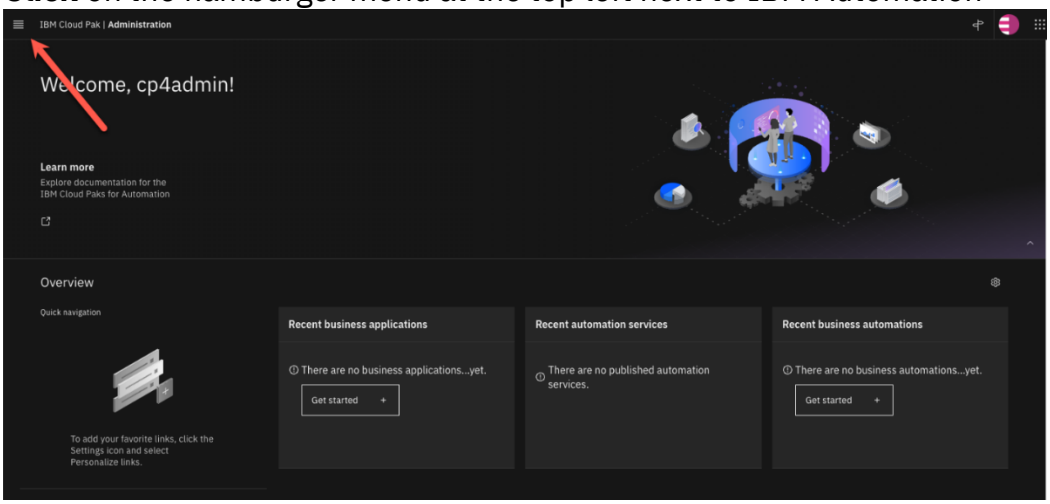
Create Business Automation Project

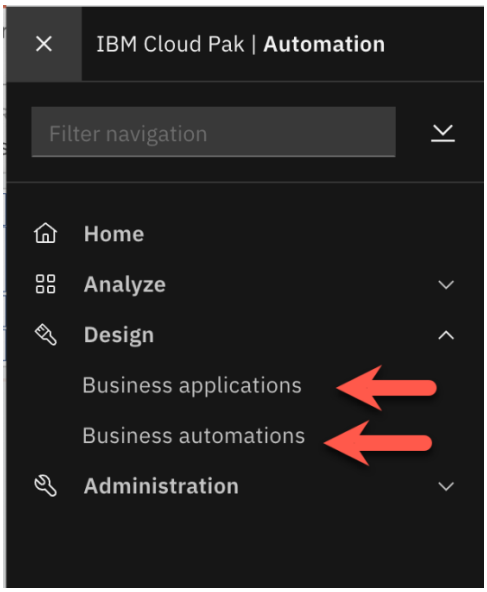
Cloud Pak for Business Automation Studio is the single authoring and development environment for the IBM Cloud Pak for Business Automation platform that accelerates digital transformation. Business Automation Studio provides an entry point to various designers to help you reach your goals.

_1. In your browser, **login** to IBM Business Automation Studio using the **Enterprise LDAP** option



_2. **Click** on the hamburger menu at the top left next to IBM Automation





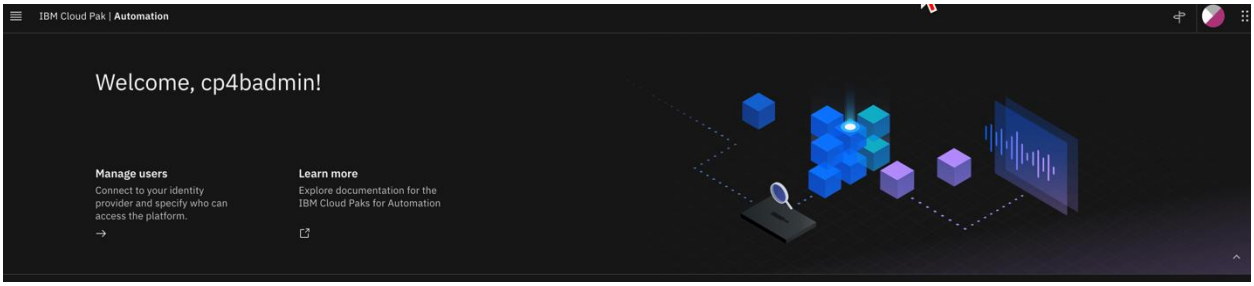
Business automations provides access to the designer of the Document Processing configuration of the document classes, and ***Business applications*** provides access to the designer for the user interfaces.

Within the *Business automations* you can create or reuse automations. An automation is a collection of artifacts that fulfills a business purpose. You can publish some automation artifacts as automation services that you can be called and reused in a consistent way. Also in Business Automation, you use the ***Document Designer*** interface within Automations to create a set of document types and related fields that comprise your Document Processing project.

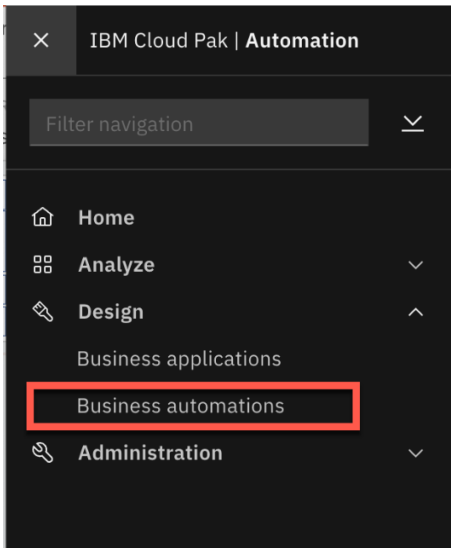
The Document Processing Designer combines an intuitive interface with a set of AI and deep learning tools that identify and learn the document types that matter to an organization. For each document type, you designate which pieces of information to extract as data for that document to be used by downstream applications. You can also apply tools to clean up and standardize the data as it is extracted.

Within *Business applications* you can quickly create user interfaces that integrate tasks, data, and automations. You can start with a template to ensure consistency. You can also use toolkits to share artifacts from existing applications.

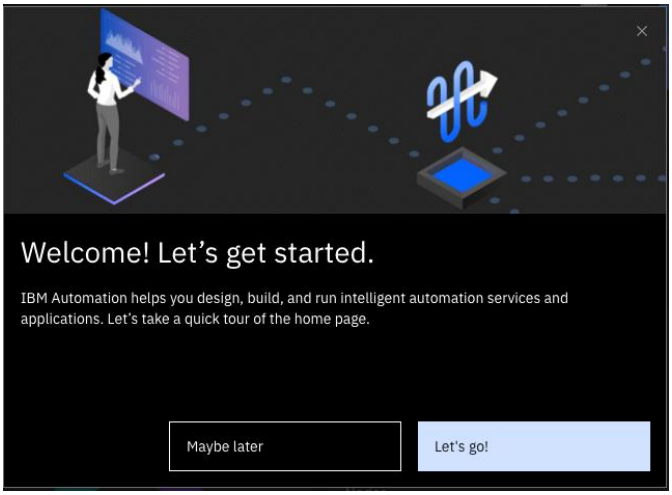
We will start with the Business Automations. Once logged in to the IBM Automation Server, you should see the Welcome screen.



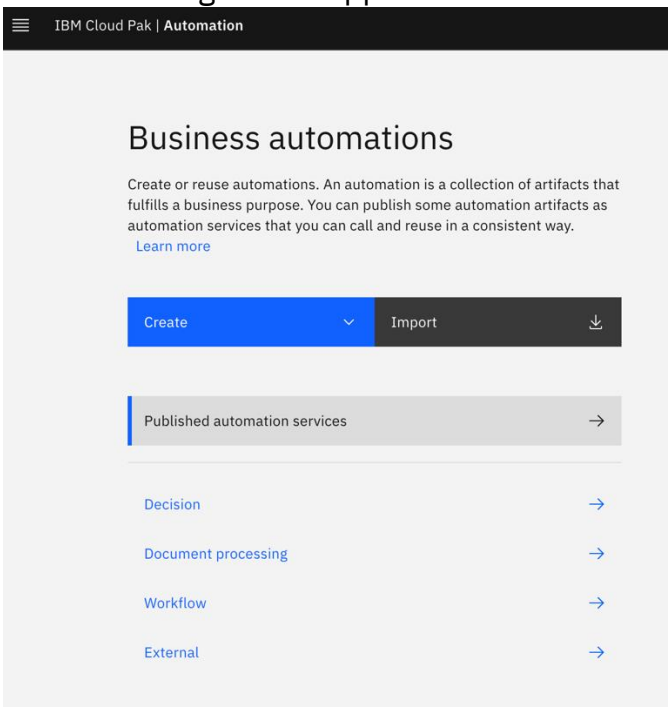
_3. **Click** on **down arrow** next to **Design** then **select Business automations**



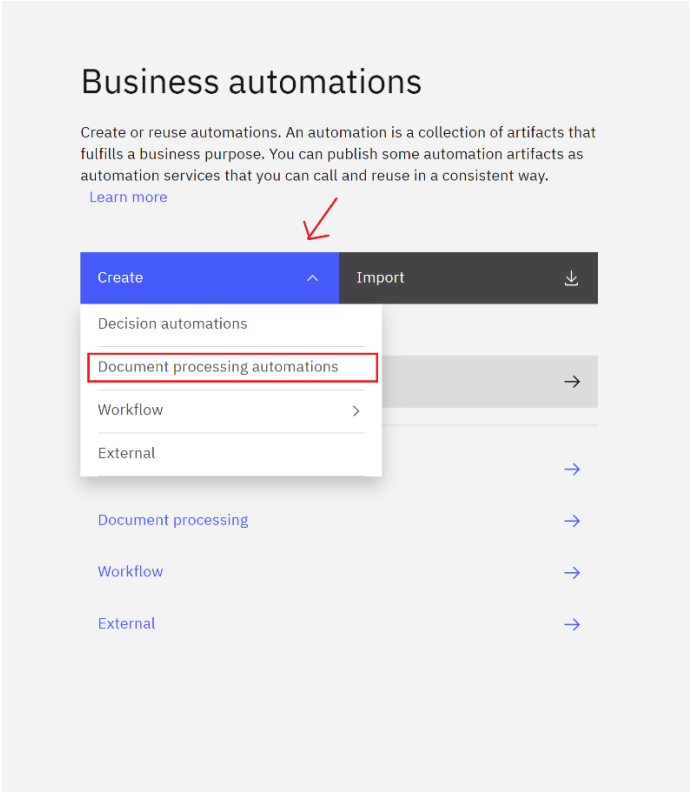
You may be presented with an overview screen. **Select Maybe Later**.



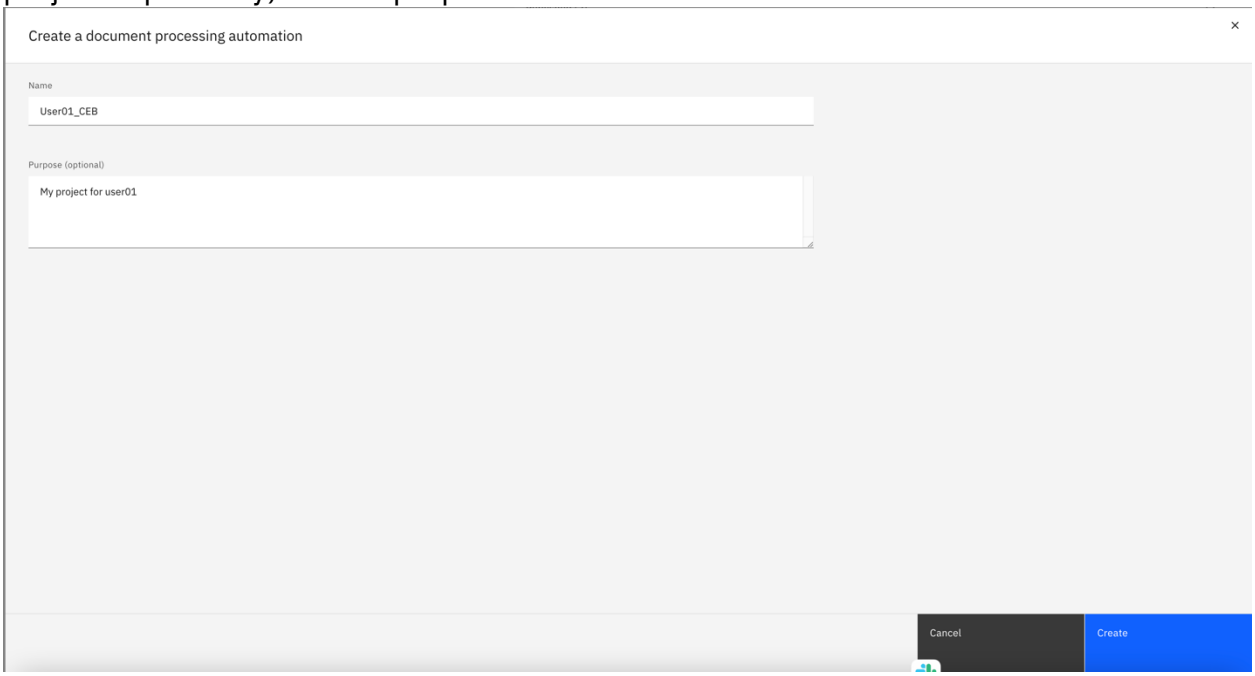
Then following screen appears



_4. Click on the **Create** twisty and select **Document processing automations**

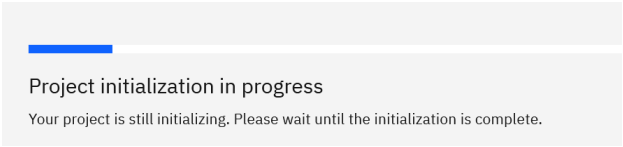


_5. In the Create a document processing automation window **enter a name** for the project. Optionally, enter a purpose.

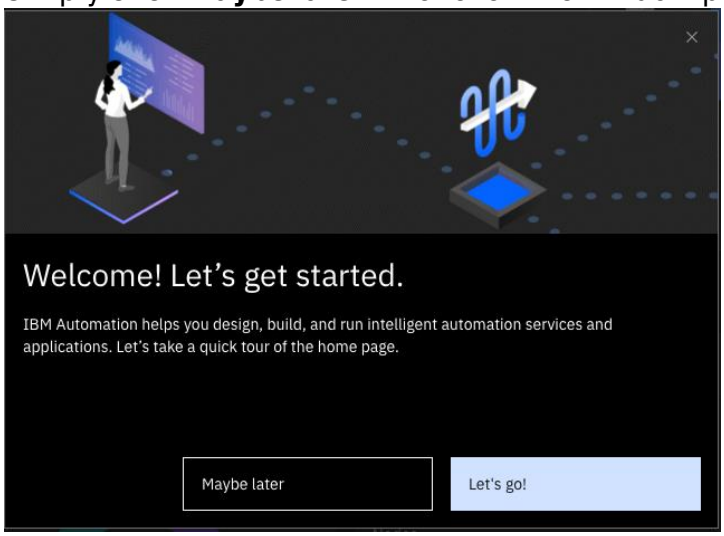


_6. **Click** on **Create** in the lower right-hand corner

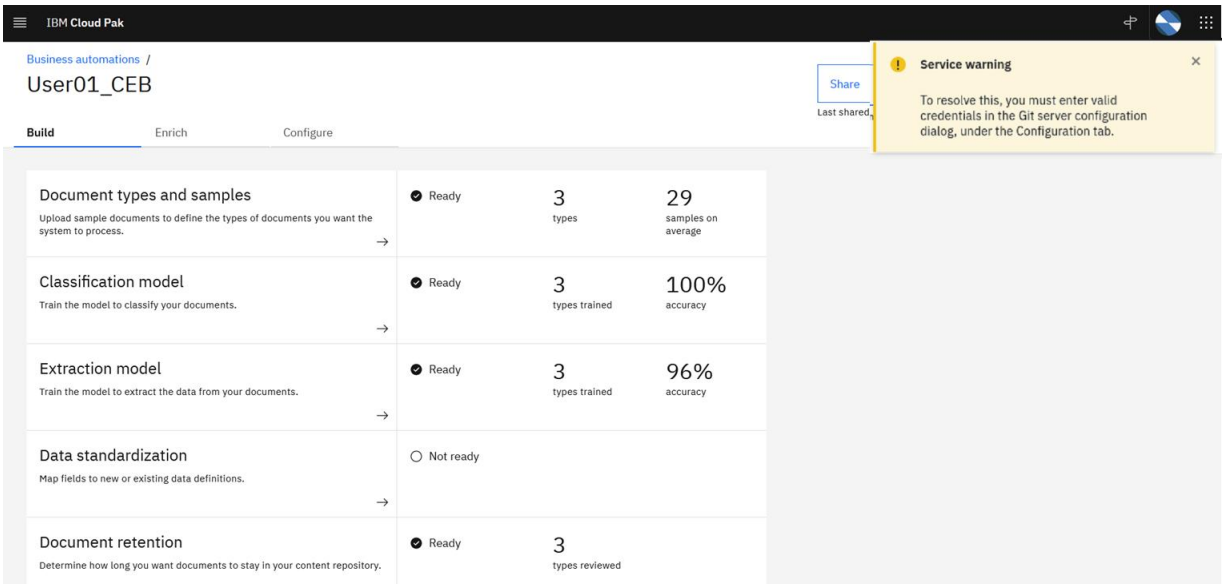
Creating and initializing the project will take some time and you will see a respective message.



You may see the *Welcome Let's get started* dialog throughout the lab. Simply **click Maybe later** whenever this window pops up.



4.1 Reviewing the interface

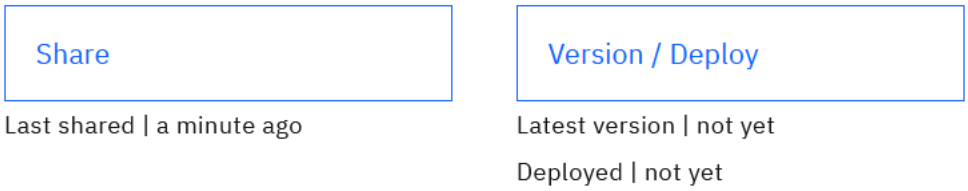


Upon opening the project, there are three major sections:

1. **Build** tab
2. **Enrich** tab
3. **Configure** tab

Depending on your environment you may initially see a yellow *Service warning* in the top right. This will only appear in case in your environment ADP is not yet connected to a Git repository. In case needed, you will take care of it in section 4.1.3, therefore close this warning for now.

Observe the ***Share*** and ***Version/ Deploy*** buttons in the top right corner.



The **Share** button is used to save your configuration to your GitHub repository.

The **Version / Deploy** button is used to create a snapshot, or version of your configuration. Like the **Share** button, the **Version** button will save your configuration, but will also create a version of it while retaining your previous version.

Once you have created a version of your configuration, you can also use this button to **Deploy** your version to the Business Applications area of ADP. You need to do this before you can go into the Business Application tile and configure your user interfaces.

4.1.1 **Build** Tab

This is what you will be spending most of your time on. The Build tab shows the guided configuration for building a Document Processing project. It shows the five steps required.

Document types and samples: Here you will define the document types that can be recognized by this automation and upload sample documents for training. By default, any project will be pre-populated with three pre-trained document types (Bill of Lading, Invoice, and Utility Bill).

Classification model: Here you will teach the system how to recognize the different document types.

Extraction model: Here you will teach the systm how to extract information for each document type based on the classification.

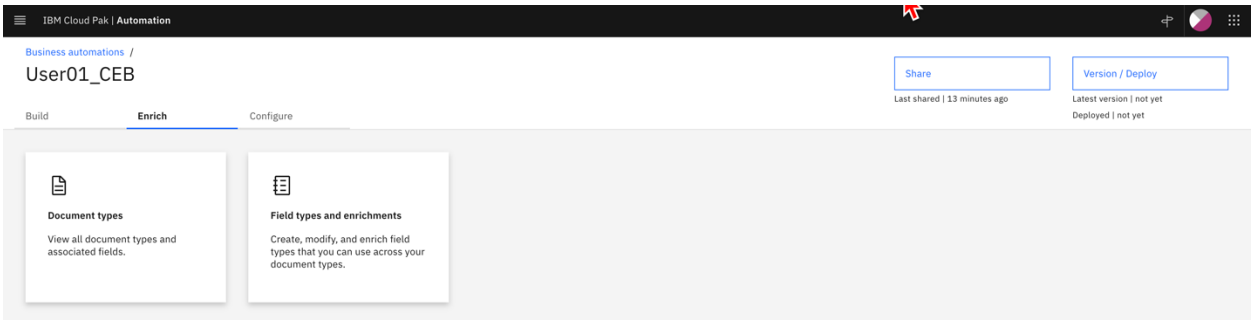
Data Standardization: This allows further refinement of the extracted information. For example, we want to standardize all dates to be formatted as YYYY/MM/DD. Having a standardized data format will help with any subsequent automation process.

Document retention: This allows us to define how long we want our documents to be kept in the system. Documents that have exceeded the retention period will be automatically expunged. This could be important for regulatory compliance or for managing the overall storage size.

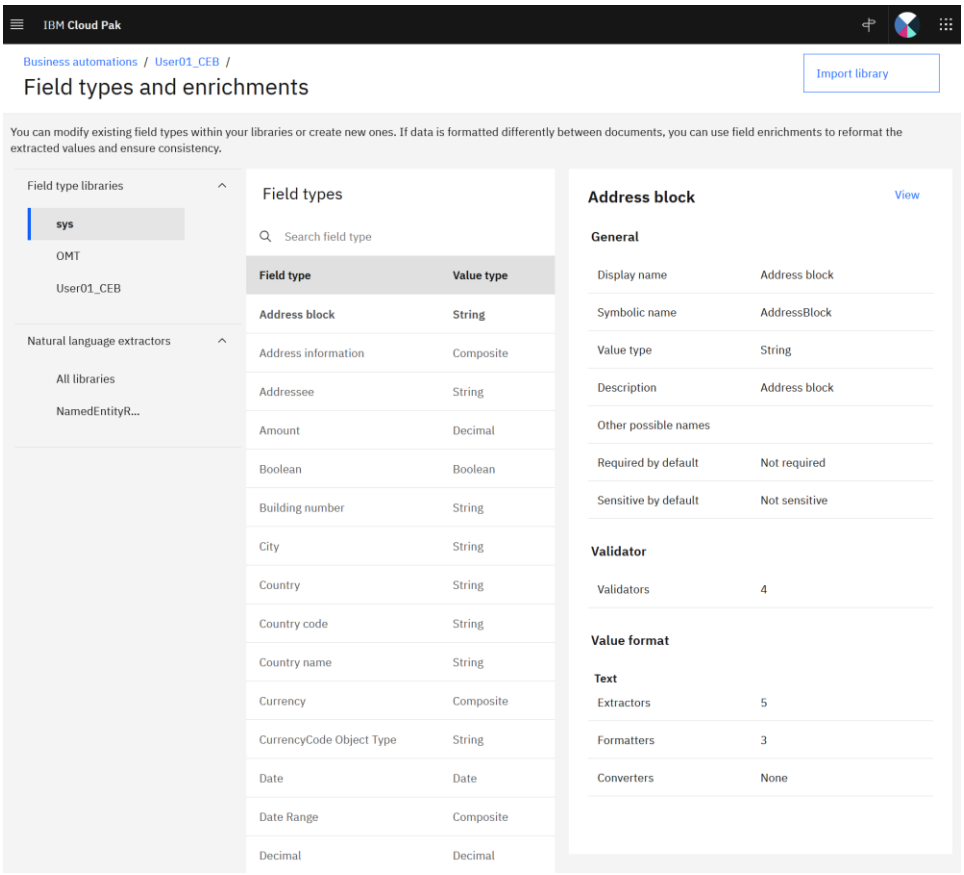
4.1.2 **Enrich** Tab

_1. **Click** on the **Enrich** tab

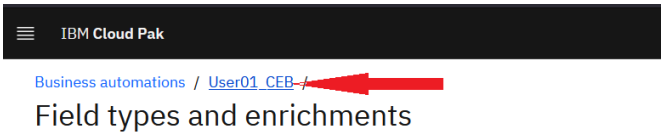
Enrich provides a quick way to define your document types and the fields you wish to extract. In this section, we can define additional enrich rules. An example of an enrich rule is to specify the expected format for an invoice number (all numerical) or a driver's license. The more we can tell document processing about how different data will be formatted, the higher the chance it will recognize the information.



_2. **Click** on **Field types and enrichments** to begin. In this tile, you will see some of the pre-configured fields in the *SYSTEM LIBRARY (sys)*. Customers can use these fields in their document type field definitions as needed.



_3. **Click** on <***your project name***> in the bread crumb trail at the top to go back to the Enrich tab.



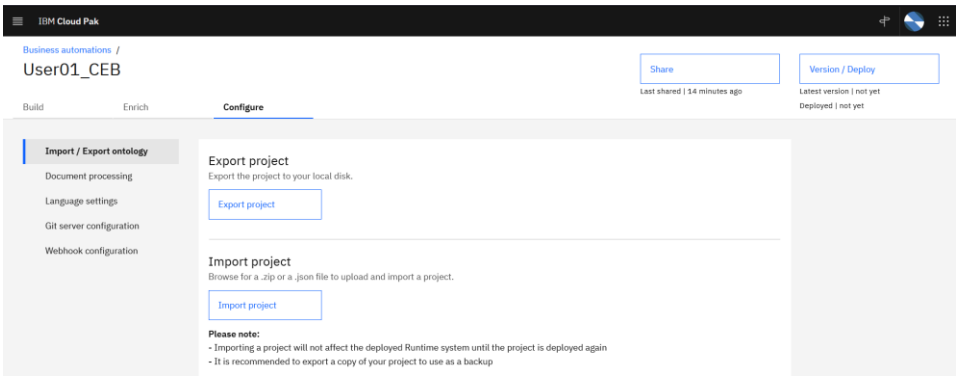
4.1.3 **Configure** Tab

_4. **Click** on **Configure** tab

This is where we can configure other operational aspects of the project.

On the default tab **Import / Export project**, the **Export project** creates a zip file that contains the document types, field types and enrichments, which you can use to start training with new sample files. You can also decide to include the training model and the sample training files in your export if you want to move your entire project to a new instance of Document Processing for example. You can import a project by clicking **Import project** selecting the zip file to import. When you import

a zip file you have two options: overwrite the existing project or merge the existing project. If you merge the existing project, document types, field types, enrichments, and sample training files are imported unless there is a conflict. Models are not imported.

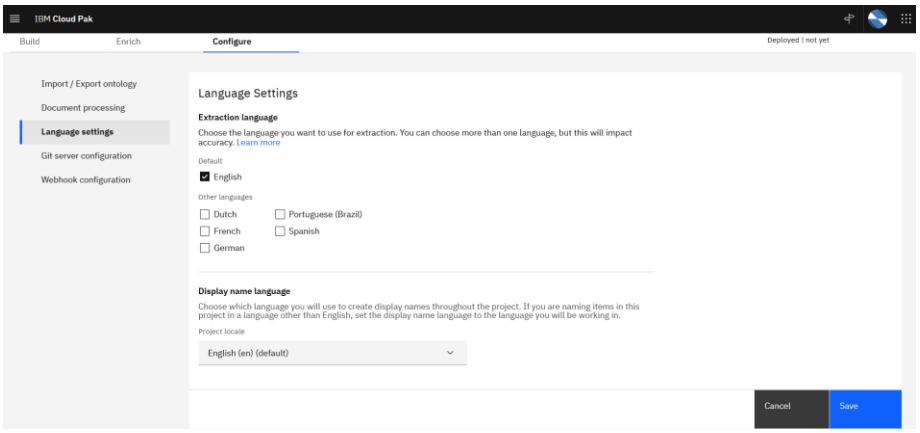


On the **Language settings** tab under **Extraction language**, you select which languages are used in the documents that you plan to process. You can choose English, Dutch, French, German, Brazilian Portuguese, or Spanish.

Make sure to choose only the language or languages that are likely to be used in your document sets. Choosing more than one language can affect the accuracy of your document processing model.

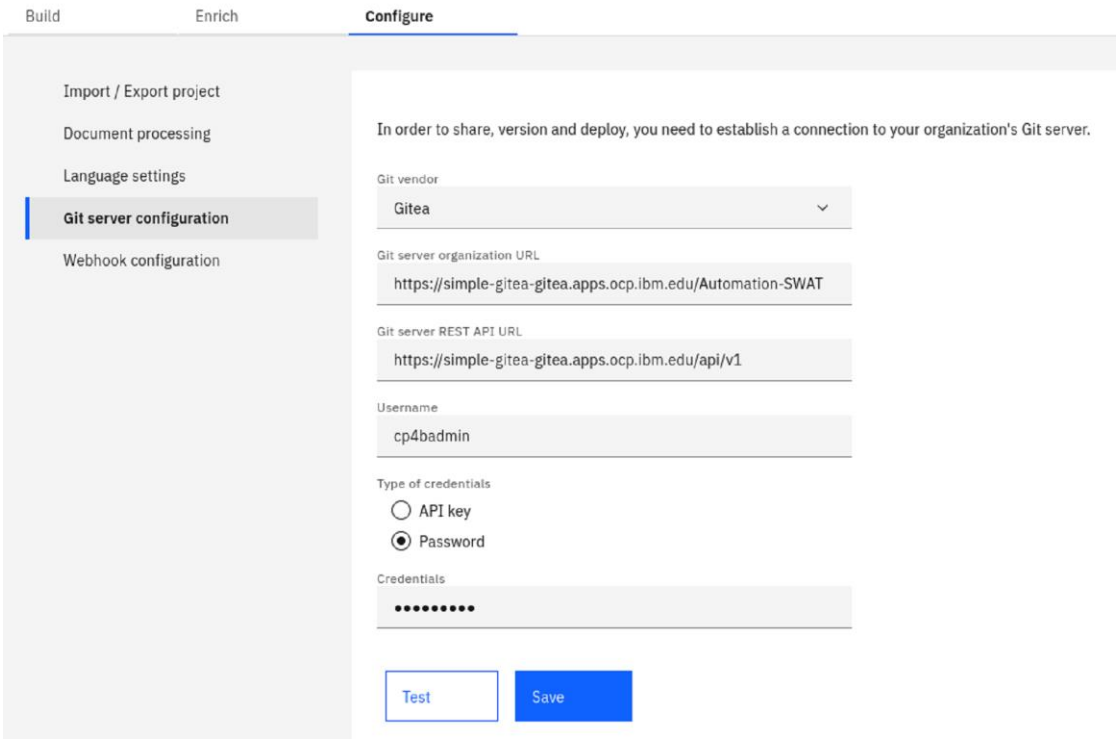
In **Display name language**, select the language that you use to enter display names for fields and document types. These are the names that are displayed in the Designer and in the applications.

The display name language is also used in the Content Engine as the localized string locale setting for document classes and properties. Document Processing project deployment supports only one language per project. If your organization has multiple projects with different language settings, these projects cannot be deployed to the same Content Engine server if they share common properties. For example, when you define data definitions during data standardization, you cannot map a field to an existing data definition that was created in a different language.



On the **Git server configuration** tab, you can create a connection to the Git server for the first project that you create in Document Processing Designer. This setting applies to all subsequent projects that you create!

The administrator of the environment can also preconfigure the Git server at deployment time. Then these fields are prepopulated.



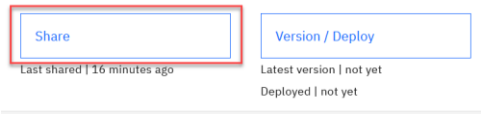
In case not prepopulated:

_5. **Fill** in the respective details for your Git server

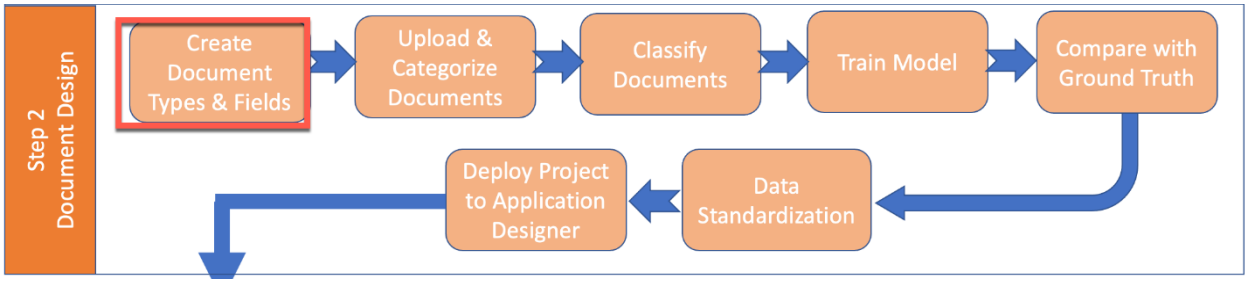
_6. First **click** on the **Test** button, which should result in a **Test connection successful** message being shown in green

_7. Once successful, click on the **Save** button, this should also succeed

_8. When the Git server connection is set up, in the top right corner **click** on the **Share** button. This is required to be able to create a version later.



5 Configure a Wage and Tax document type

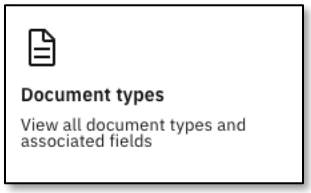


Before we use the guided configuration, you will configure some additional document types and fields used to extract data prior to uploading sample documents.
To do this lab, we will use the *Enrich* tab to add fields to a newly created Wage and Tax document type.

5.1 Create Wage and Tax document type

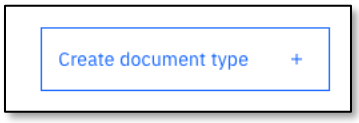
_1. **Click** on the **Enrich** tab

_2. **Click** on **Document types**



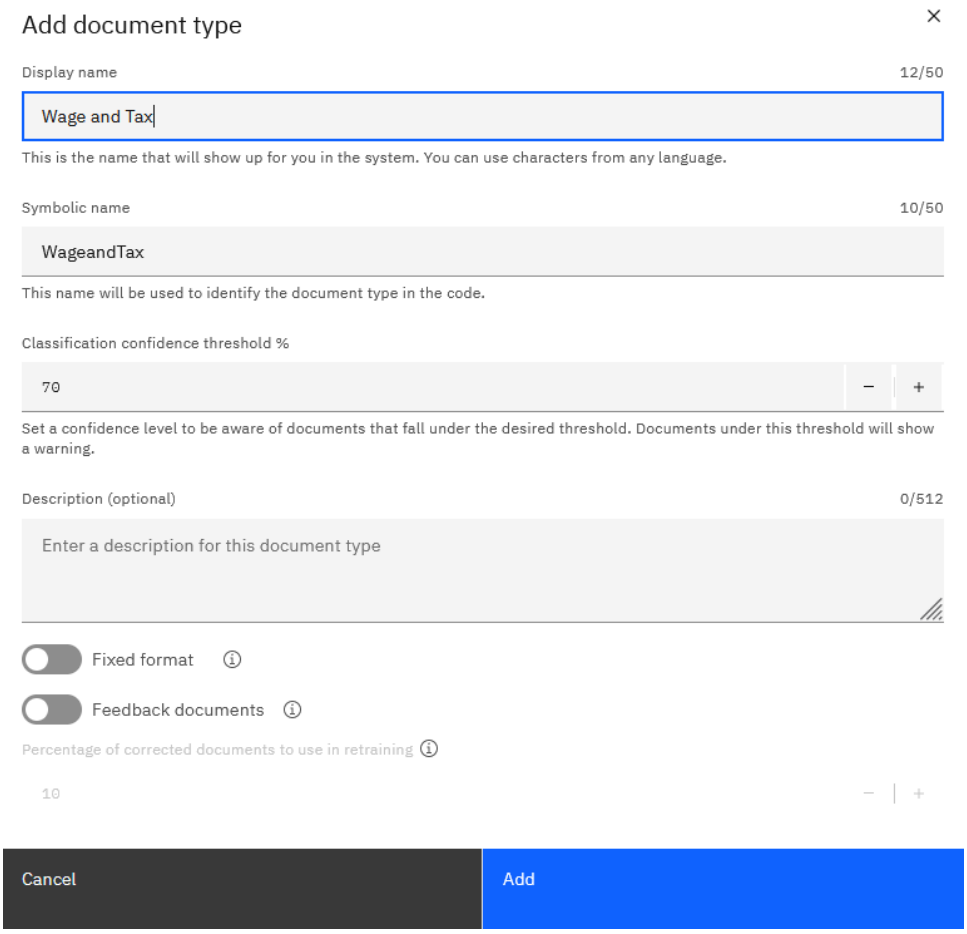
You will now create a document type for Wage and Tax documents and fields to extract data from them.

_3. **Click** on the **Create document type +** button in the top right corner



_4. The *Add document type* window pops up. **Enter "Wage and Tax"** for the display name

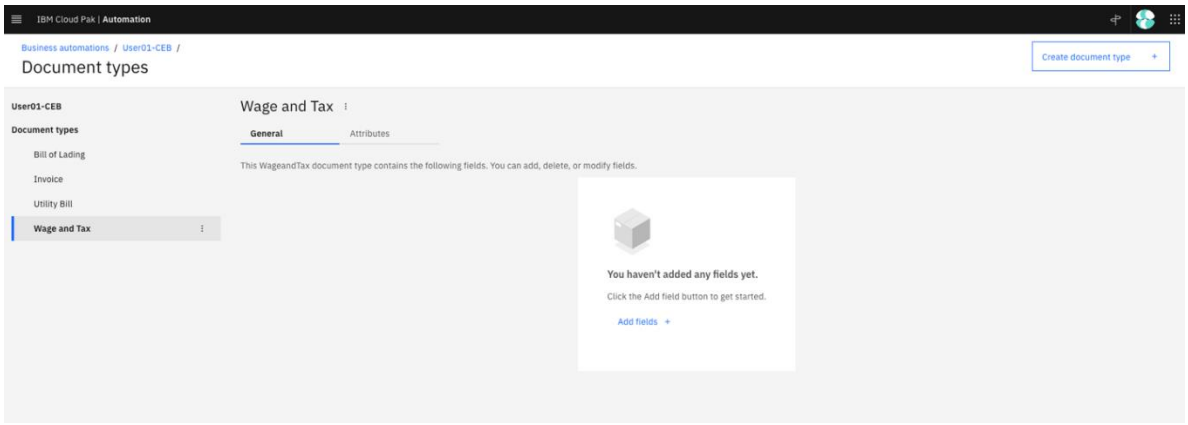
There is no need to enter a symbolic name, ADP will use the display name as a base and remove the spaces. There's no need to add description in this lab unless you want to.



Note: *Notice the option for "Fixed-format document type". If your form is static in nature or has a fixed structure that does not change, select this option so you will not have to provide as many samples. In our use case Wage and Tax documents have a variety of formats and are not static.*

_5. **Click** the **Add** button

You should now see your new document type (class) in the list of classes on the left.

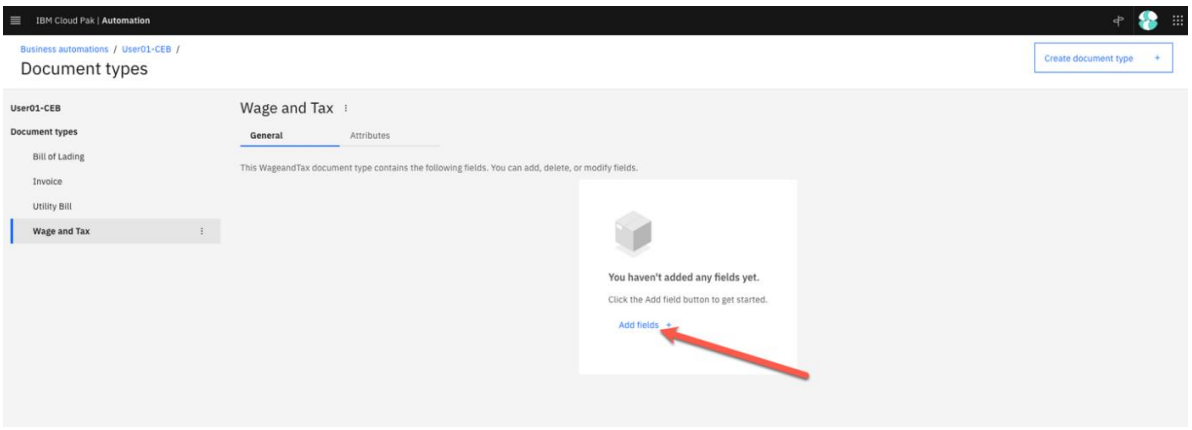


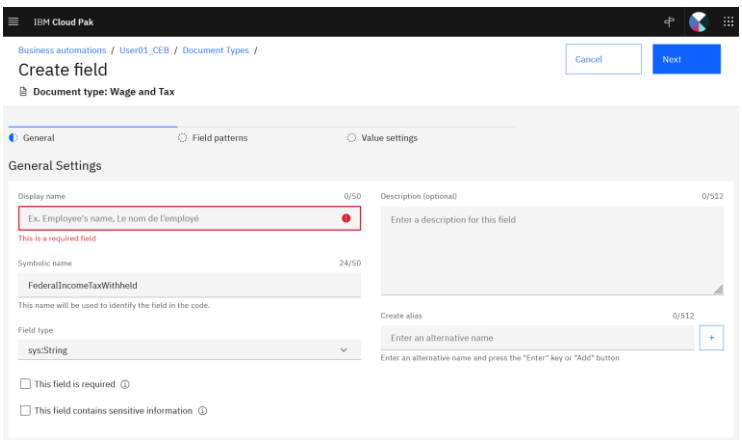
_6. **Select** your **Wage and Tax** doc type. On the right, you should see an empty table of fields.

5.2 Create Field

We can now add some fields to the class. From examination of the forms, we can see there are different fields names, or they are not consistent across the forms. We'll need to add these different "aliases" during this process.

_1. **Click Add fields +** to get to the wizard to define a new field



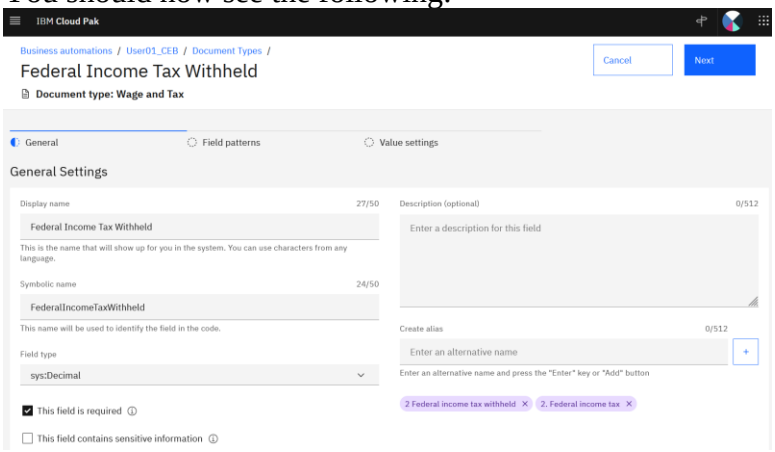


_2. **Enter** the following values under the **General Settings** header:

- Display name: **Federal Income Tax Withheld**
- Field type: **Sys:Decimal**
- This field is required: **Yes**
- In Aliases enter other possible names. Case and punctuation are very important when creating aliases. Enter the alias listed below. These are representations of what it looks like on the different forms. **Press** the **"+"** after entering each one or **press Enter** key:
    - **2 Federal income tax withheld**
    - **2. Federal income tax**

Note: *In the second case, the number two has a period after it!*

You should now see the following:



_3. **Click** the **Next** button

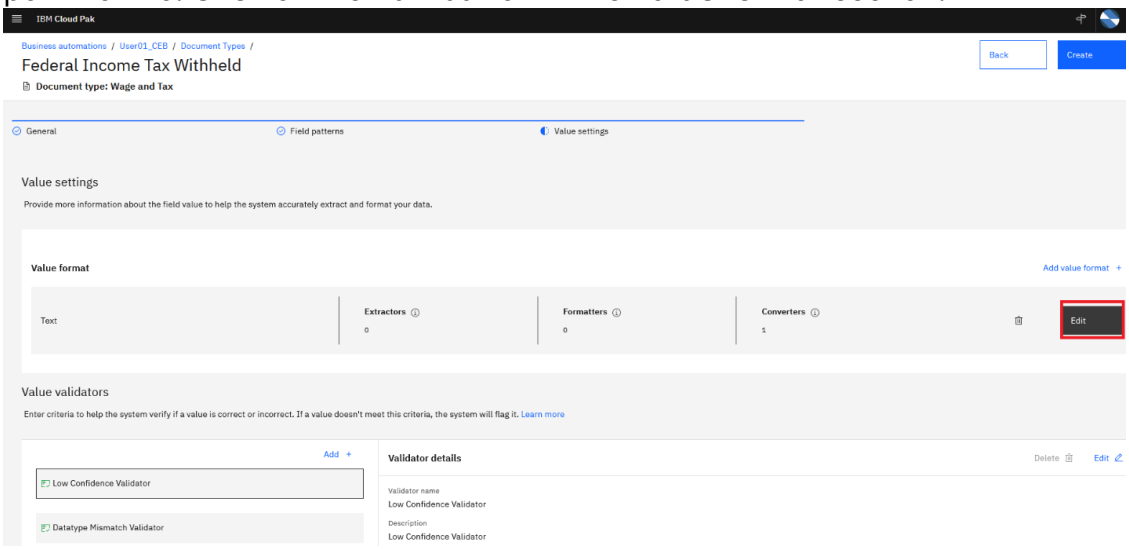
Field patterns are regular expressions that can be associated with a field to help identify and extract fields and their values. A regular expression is a sequence of characters that define a search pattern. The use of regular expression patterns and extractors is optional. Regular expression patterns can provide extra information to potentially improve the accuracy in extracting the correct fields. Python syntax is used for defining the regular expressions. You will not be adding any field patterns in this lab.

_4. **Click Next** again. You should now be on the **Value settings** page. This is where you can set up validators, formatters, and converters.

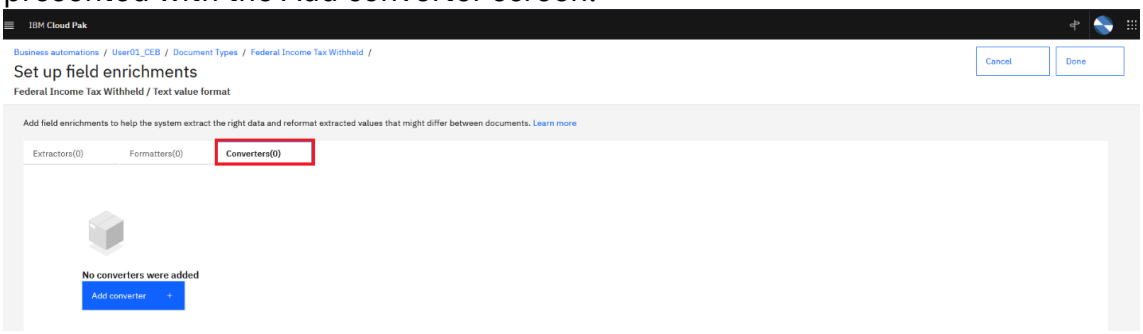
Value Settings for a specific field; if the potential values follow a rule that can be expressed in a regular expression, you can specify an extractor. This pattern can match all the variations of your values. For example, the expected value for a Start Date field might be in a date format. You can create a regular expression pattern for `US Date` and then associate the extractor of `US Date` to your field.

Also, sometimes you want to extract a value that does not have a corresponding key in the document, but you know the pattern of the value. You can define the extractor and denote that the value might be anywhere in the document without attaching to the field name. This designation allows for the presence of a field name to be optional. For example, you want to extract the employee ID number, which can be described with a regular expression pattern. However, some documents show the employee number with a field name Employee ID, while other documents show the employee number without a corresponding field. You can specify the Extractor and be able to extract the employee ID number in both types of documents.

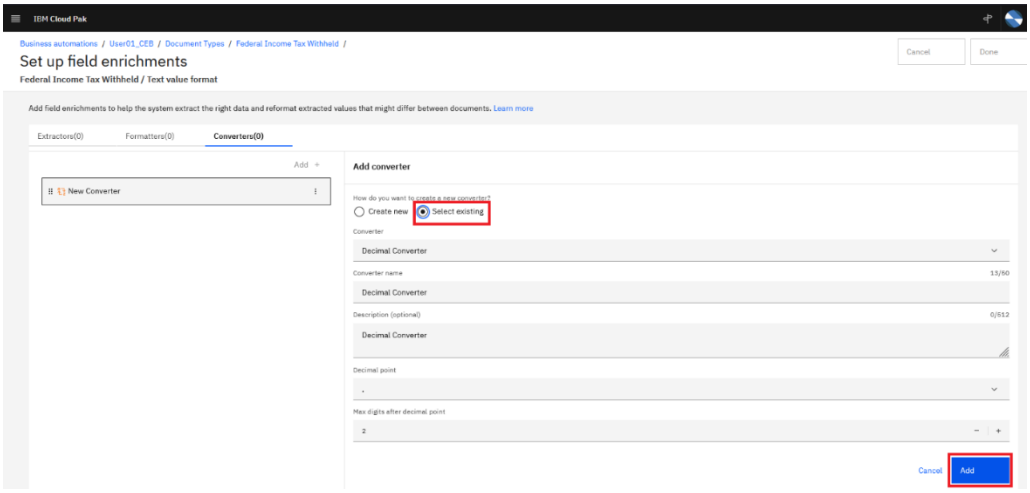
_5. The decimal data type can contain only integers to the left and right of a decimal point. But some of our data may contain commas between the integers and we only need two integers after the decimal point. Let's add a converter that will remove all extra punctuation and limit the number of integers after the decimal point to two. **Click** on the **Edit** button in the **Value format** section.



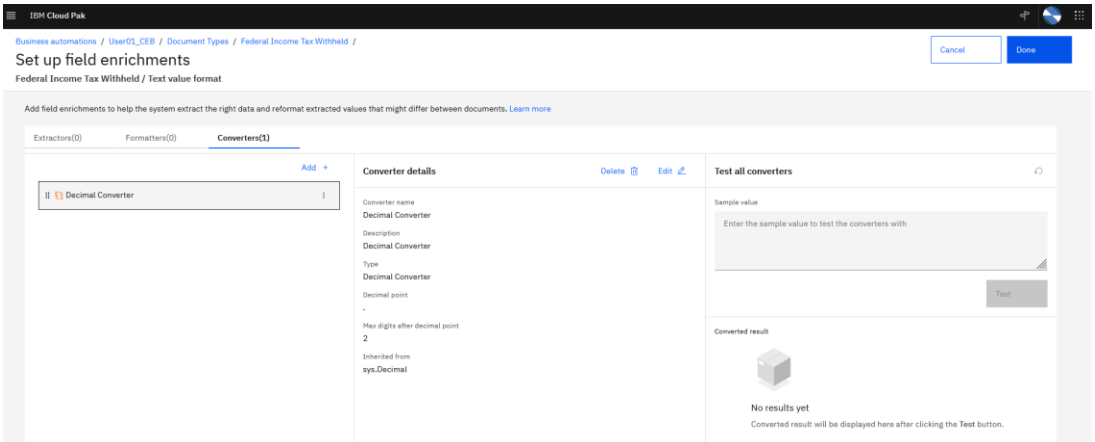
_6. **Click** on **Converters** tab then click on the blue **Add converter +** button. You will be presented with the Add converter screen.



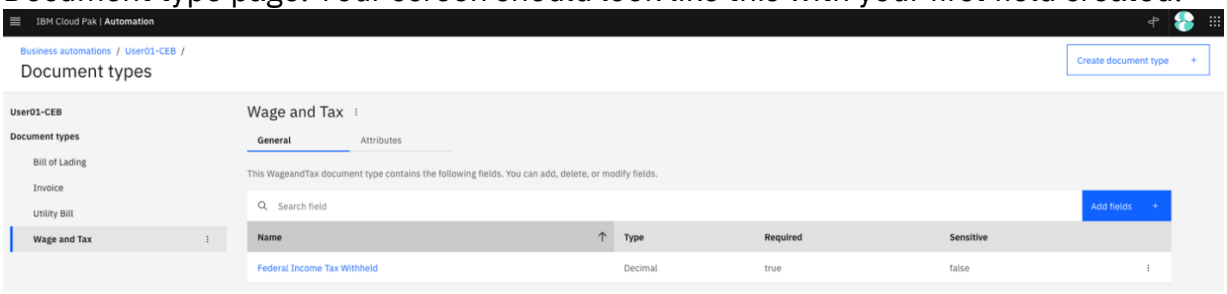
_7. **Click** on **Select existing.** This populates the converter name, description, Decimal point, and Max digits after decimal point for you. If you wanted to change the decimal point from a period to a comma you could do it here as they do in other countries outside the United States. **Click** the blue **Add** button.



_8. You will then be presented with the Converter details information screen. On this screen you can also test your converters to make sure they are behaving like you intended. **Click** on **Done** at the top right. Refer [Enrichments-Converters] for more details.



_9. **Click Create** in the top right. Once it is created you will be taken back to the Document type page. Your screen should look like this with your first field created.



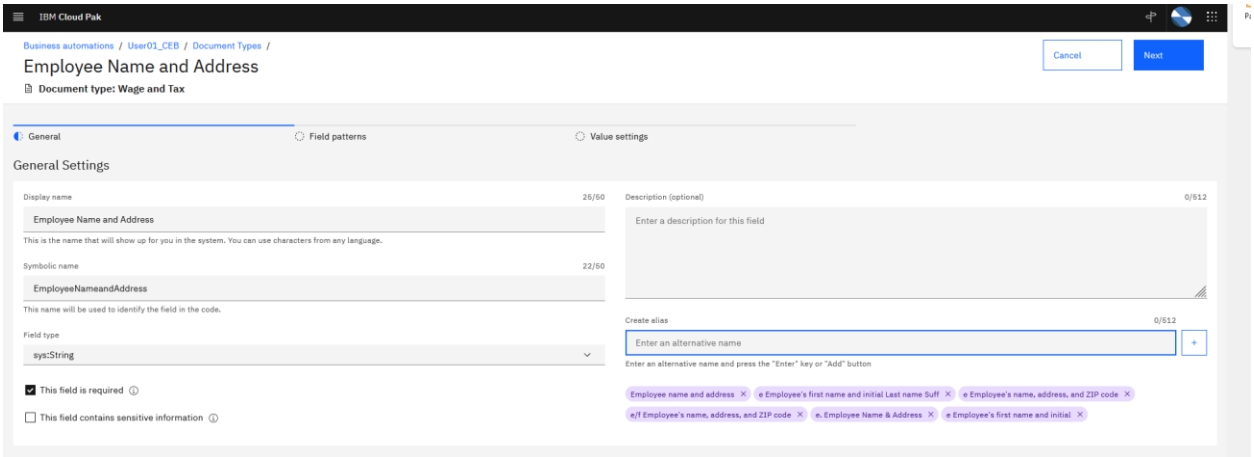
5.3 Create the Employee Name Address field

_1. **Click Add fields +**

Enter the following values under the **General Settings** header:

- Display name: **Employee Name and Address**
- Field type: **sys:String**
- This field is required: **yes**
- Enter the following other possible names (aliases):
  - ***Employee name and address***
  - ***e Employee's first name and initial Last name Suff***
  - **e Employee's name, address, and ZIP code**
  - **e/f Employee's name, address, and ZIP code**
  - **e. Employee Name & Address**
  - **e Employee's first name and initial**

By default, the system will use the field name as an alias. So, you do not have to add it. For example, below, Employee Name and Address (field name), would be automatically used as an alias even if you do not add it to the list.



_2. **Click Next**. No field patterns will be created.

_3. **Click Next**. No value settings will be created.

_4. **Click Create** to finish creating the Employee Name and Address

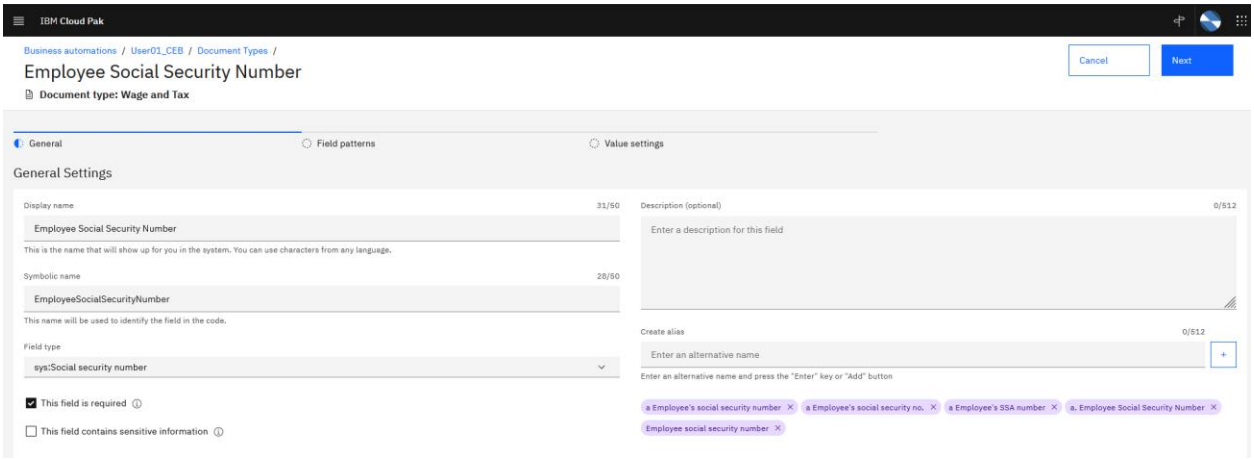
5.4 Create Employee Social Security Number Field

_1. **Click on Add fields +**

Enter the following values under the **General Settings** header:

- Display name: **Employee Social Security Number**
- Field type: **sys:Social Security Number**
- This field is required: **Yes**
- Other possible names (aliases). Remember, press RETURN or hit the '+' button on your keyboard between each entry:
  - **a Employee's social security number**
  - **a Employee's social security no.**
  - **a Employee's SSA number**
  - **a. Employee Social Security Number**
  - **Employee social security number**

Your screen should now look like below:



_2. **Click Next**

_3. **Click Next** again on the Field patterns screen

_4. **Click Create** on the Value settings

_5. **Create** the following **additional fields**

The following table contains the values to use when adding the additional fields.

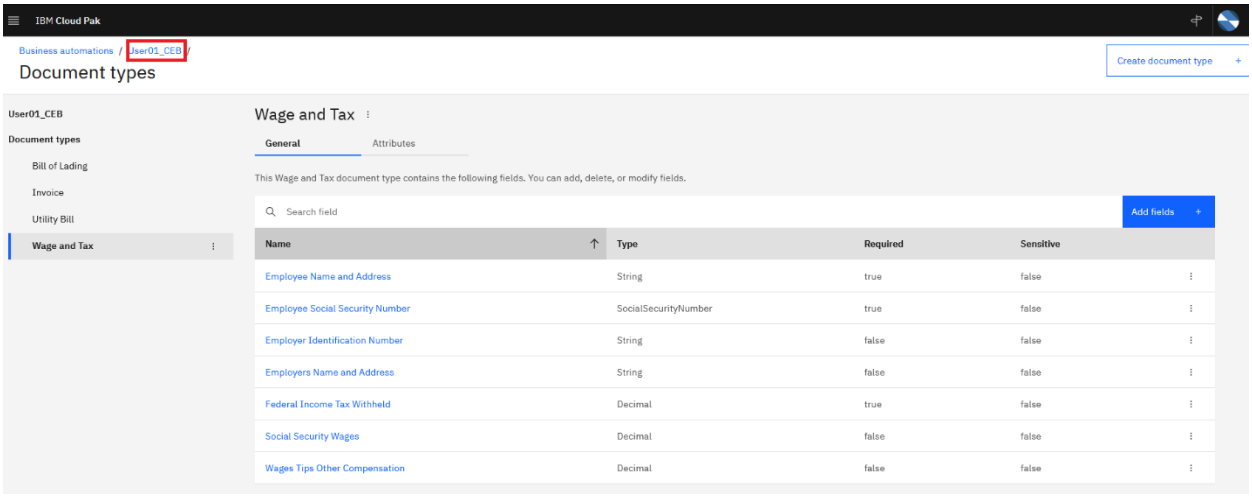
Follow the steps from the previous section to add the following fields. **Don't forget to add your converter for datatypes of Sys:Decimal.**

Display Name	Description	Type	Mandatory	Aliases
Employer Identification Number		sys:String	N	• b Employer identification number (EIN) • b Employer's FED ID number • b. Employer ID number • Employer identification number
Employers Name and Address		sys:String	N	• c Employer's name, address, and ZIP code • c Employer's Name & Address • Employers name and address
Social Security Wages		sys:Decimal	N	• Social security wages • 3 Social security wages
Wages Tips Other Compensation		sys:Decimal	N	• 1 Wages, tips, other compensation • Wages, tips, other comp. • 1 Wages, tips, other comp. • 1. Wages tips, other comp • Wages tips other compensation

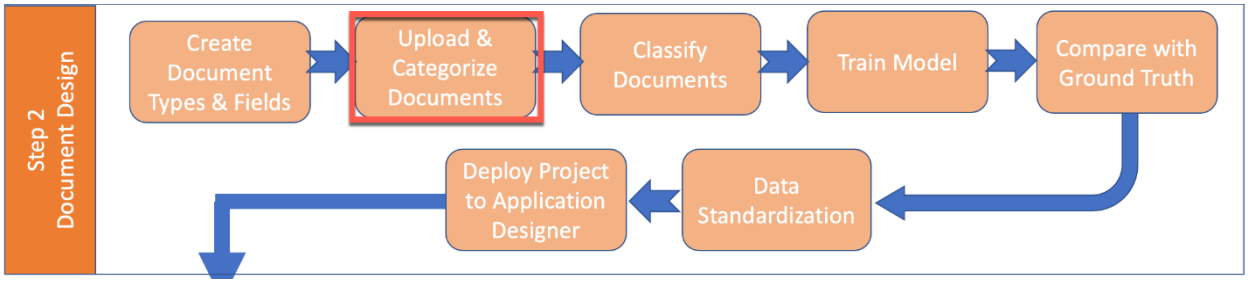
Reference for various field types:

> ***Note:*** *The basic default field types included in ADP are found here in the documentation*
>
> *https://www.ibm.com/docs/en/cloud-paks/cp-biz-automation/24.0.0?topic=enrichments-field-types-document-processing*

_6. **Click** on the ***<name of your project>*** in the breadcrumb link in the top left of your screen. This will take you back to the **Enrich** tab, then **click** on the **Build** tab.



6 Document types and samples overview



At this point in the process, we have created a new document type and configured the field names we want to extract off the document. For the system to know what to extract from your documents, it needs to be able to classify the documents. In this part of the lab, we will teach the system to recognize the various document types on your system.

In the first part of the classification section, you will explore the system's ability to automatically group similar documents together. This can be used to discover document types in a file share for example. You can also upload documents and have the system tell you what it finds. You would then use this information to create document types so you can classify the documents and data extract fields.

The project template comes pre-loaded with three document types: Bill of Lading, Invoice, and Utility Bill. In the last section we added a new document type *Wages and Tax*. In the *Build* tab of your project, you should now be seeing 4 document types. The three pre-loaded documents already have documents in them. You will be adding documents to the Wage and Tax document type. Your actual screen may vary from the screenshot below.

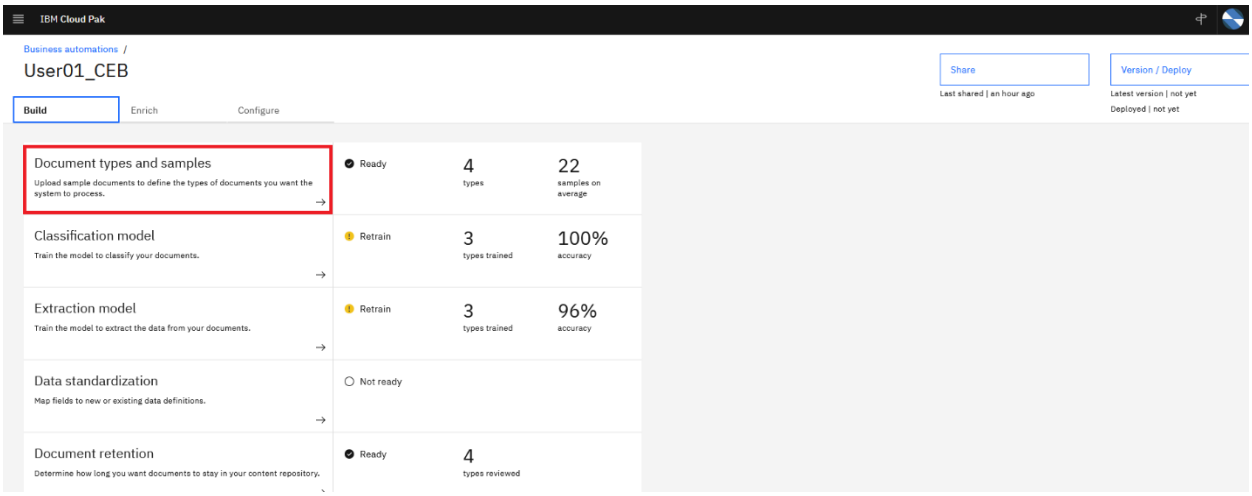
You will be asked to review the document categories the system finds and create the appropriate document types as needed.

6.1 Categorize documents

For categorizing, we will have the system help us group similar documents together.

To get started

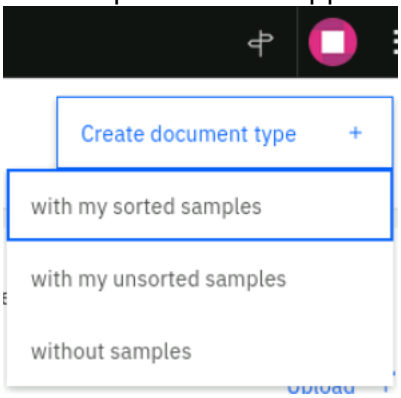
_1. **Click anywhere** in the **Document types and samples** box



The *categorize* feature analyzes each document and tries to find similarities between them. Based on these similarities, the system will divide the samples into categories for you to review. You can add documents or entire categories into either an existing document class or create new classes as needed. Let's see what that looks like.

_2. **Click** on **Create document type** in the top right of the screen

The drop down that appears:

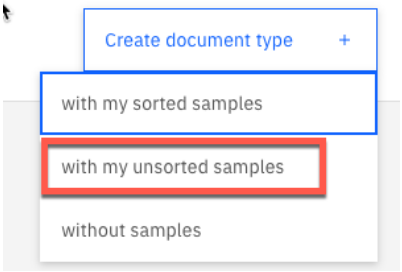


If you have the same document types already separated into folders, you can choose the first option, *with my sorted samples*. The system would simply ingest the documents from each folder into a different group.

For this exercise, we will select the second option, *with my unsorted samples* and let the system sort the documents for us. Use this option when you don't know how many different document types there are.

Select the second option titled **with my unsorted samples**

_3.

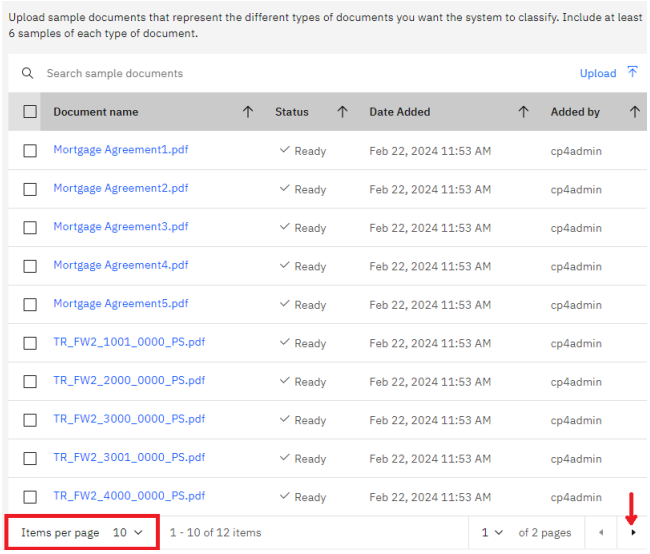


You should have already downloaded the files from Section 2 to your laptop. You can select upload and grab all the files from where they were downloaded to on your laptop. Make sure you have already unzipped them.

Click Upload to upload the document samples

_4. From the downloaded sample documents open the folder name ***Group 1 – Design Docs for Tax Lab*** *and select all documents.*

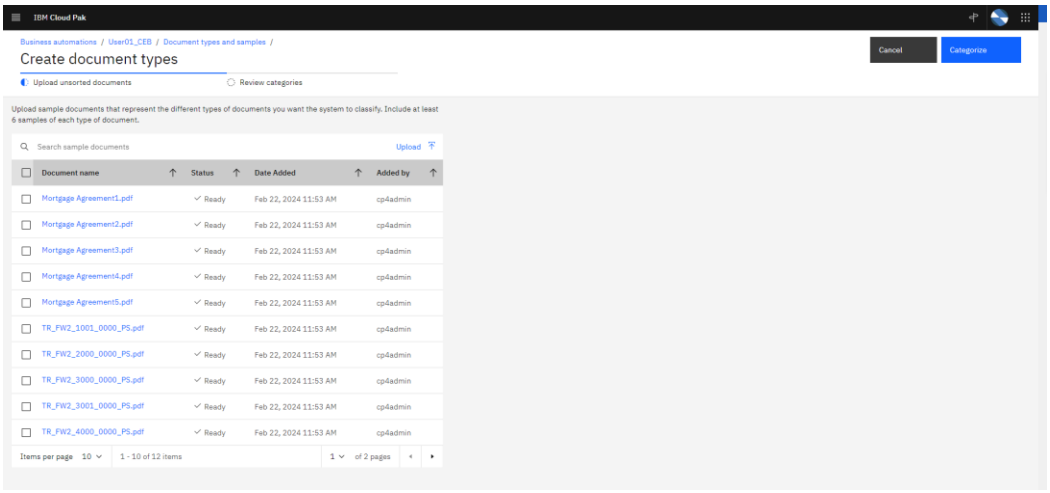
At the bottom of the window, you can select the number of items to display in the window or click on the arrows to move to the next page.



> ***Note:*** *This will take several minutes, good time for some coffee or a stretch. Make sure to check ALL documents have been uploaded there are two pages or 12 items to verify.*

Click on the blue **Categorize** button on the top right corner

_5.



> ***Note:*** *The results may vary based on the documents uploaded, what the system already has learned, the version of ADP and more. Please look at this lab exercise from a high level. The categories you will be presented are the system's best guess on how they should be separated.*

You will need to:

- Review the categories to see if the documents were separated correctly
- Move documents into either a NEW document type or into an EXISTING document type
- There should be 3 types in the samples you were provided
  - Wage and Tax
  - Utility bills
  - Mortgage Agreements
- You will need to assign either an entire category (i.e., all sample documents) or individual documents in each category to the Wage and Tax and Utility bills document types which already exist on your system
- You will need to create a new document type for Mortgage Agreements

_6.

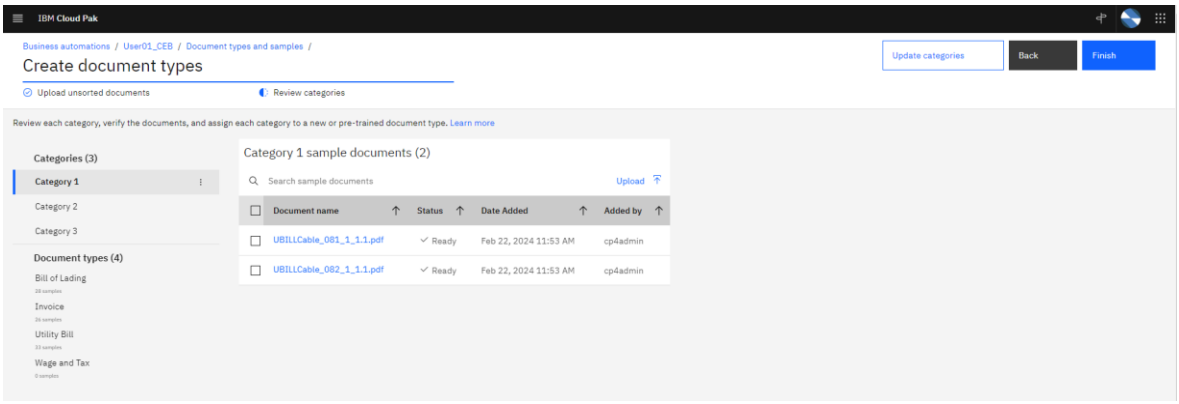
After a few seconds, the system will mark the documents with a status of ready as seen in the above image.

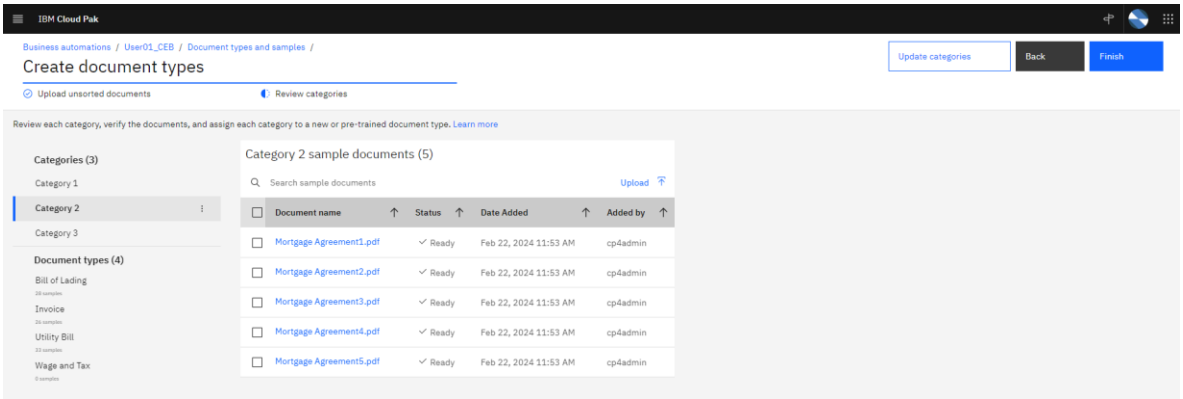
Click on **each of the categories** to see what was grouped together as shown below.

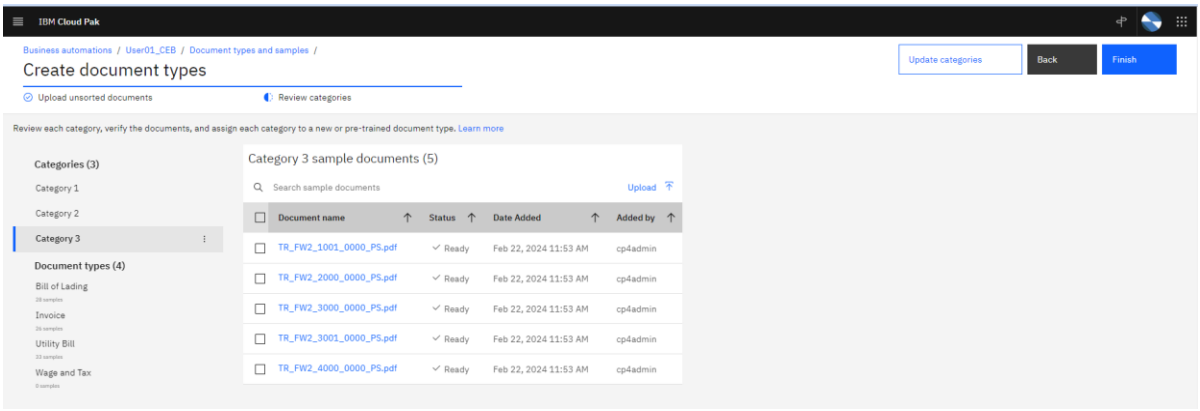
The order of the categories shown in the screenshots below may differ from the order in your environment.

You can click on any document to see a preview of it. This will help ensure the documents are correctly grouped.

> ***Note:*** *The names of the files are not used in any way in this process. The files were merely named this way to make it easier for you to quickly ascertain whether the documents were grouped correctly.*



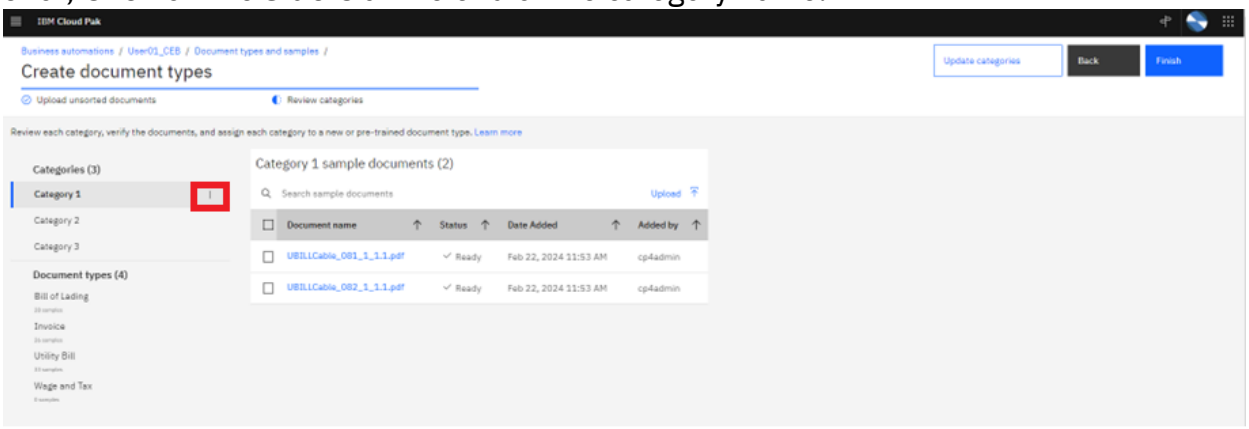




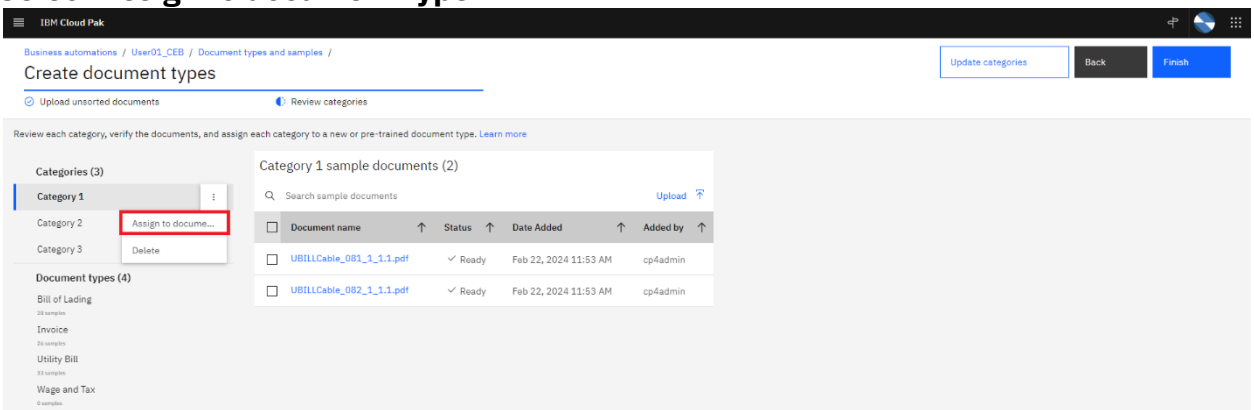
At the time of writing this documentation ADP was able to categorize the sample set into each category. This is not always the case, sometimes document types will be combined into one category, so it's very important to look at each category and verify documents.

For each of the three categories perform the following steps:

_7. If all documents within a category are correct as illustrated in the following screen shot, **Click** on the **3 dots** at the end of the category name.



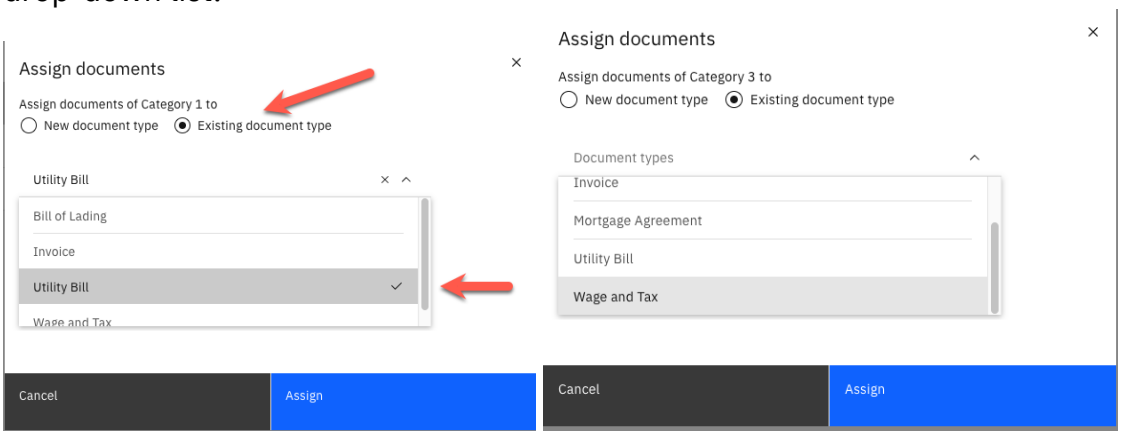
_8. **Select Assign to document type**



If the documents are either of type **Utility Bill** or **Wage and Tax**:

Select Existing Document type then the appropriate **document type** from the drop-down list.

_9.

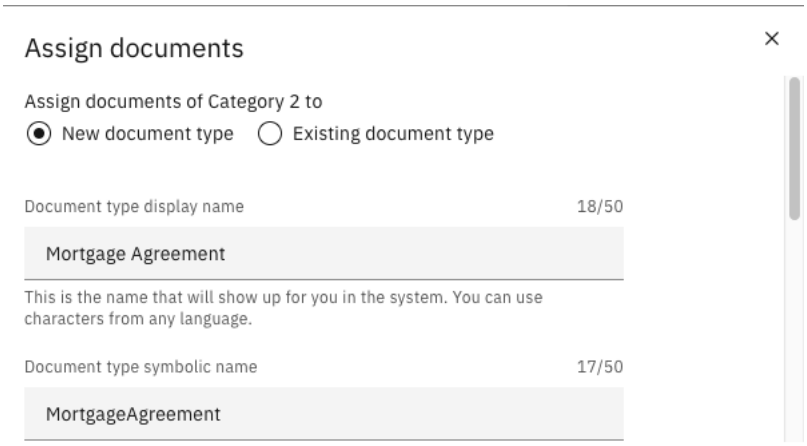


Click Assign to close the dialog box

If the documents are of type **Mortgage Agreement**:

Select a **New Document Type.** Since we have not defined a mortgage agreement document type yet.

Enter *Mortgage Agreement* in the field

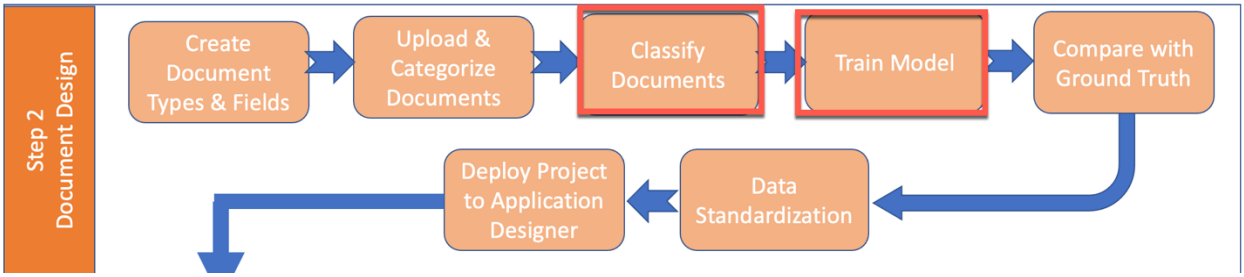


_10.

Click Assign to have the system automatically rename and move the category into the Document Types section.

Click the **Finish** button in the top right corner

7 Train classification



Now that we have documents uploaded in the system, we are ready to train the classification. Note that although you don't need a ton of document samples to train (minimum of 5), you are going to get better accuracy if the system has a deeper understanding of the documents, so more could be better.

In this lab, we curated some document samples for you. In normal circumstances, you would need to do this yourself. Make sure the documents you upload to train classification are good documents:

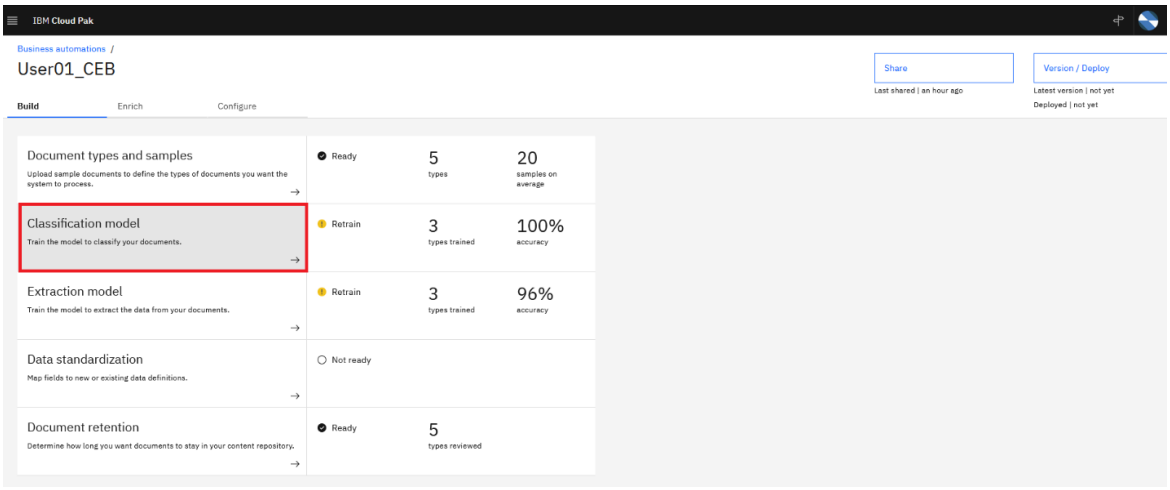
- Clean documents
- High resolution
- Representative of the document type(s)
- Accurately grouped and uploaded to Document Processing

This is NOT the time to try and trick the system. Uploading a document that doesn't get recognized well would not help the system recognize the types of words, phrases, and concepts it needs to learn to classify documents correctly.

The most common error is introducing a sample document into the incorrect document type, usually by uploading them to the wrong document type. If that happens, you are introducing conflict into the classification. For example, an invoice added to Tax Forms may confuse the system and result in it thinking invoices are tax forms and vice versa. Once that happens, you need to clean your documents and retrain the system.

_1. **Click** on ***<your project name>*** in the bread crumb trail to return to the start page

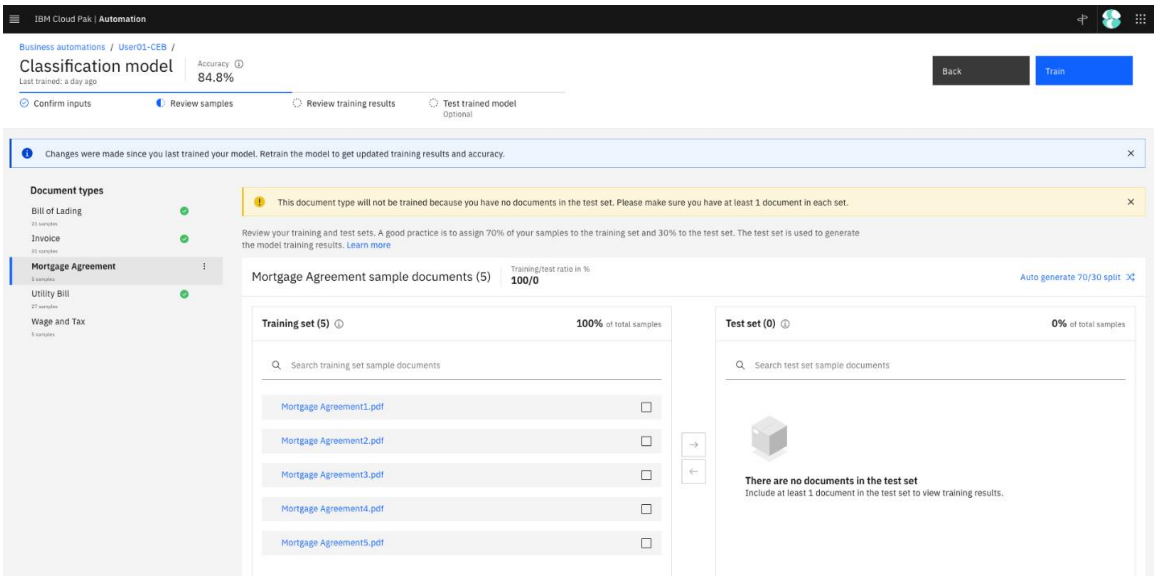
_2. **Click** anywhere in the **Classification model** line



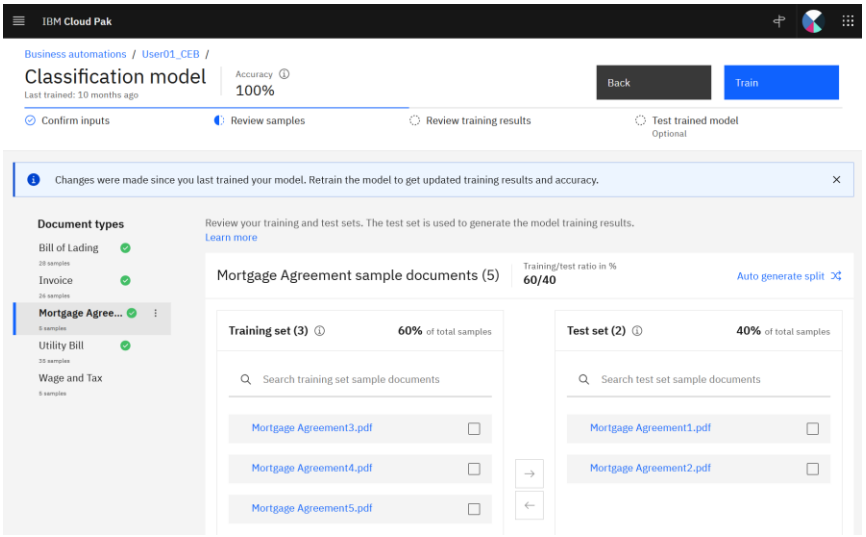
Once we open the classification model, you will be presented with details on how to perform the retraining. There are four basic steps – Confirm inputs, Review Samples, Review Training Results, and Test Trained model.

On the *Confirm inputs* screen you can confirm all the documents that will be used in this training exercise. We can also use the opportunity to remove documents that are no longer relevant or upload additional documents.

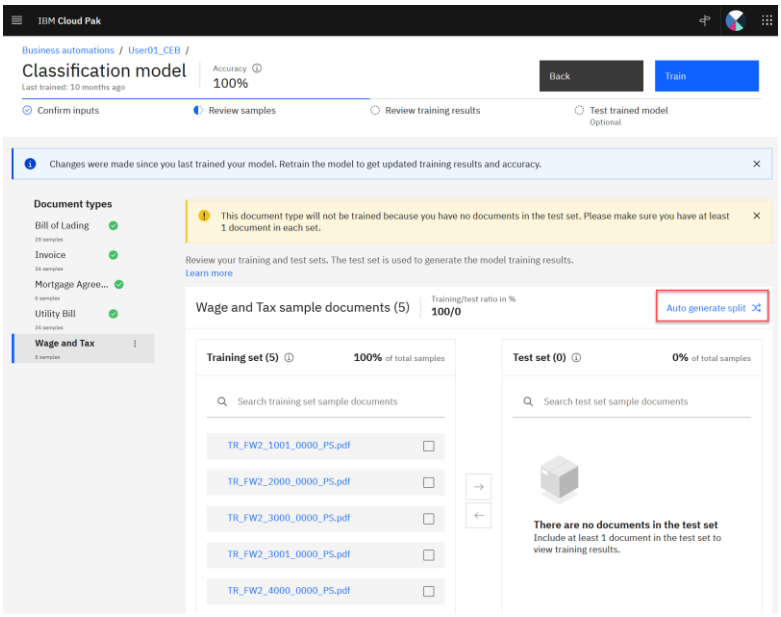
_3. **Click Next** this will move from the ***Confirm inputs*** to the ***Review Samples*** step. Notice three document types have green icons next to them. These green icons show these documents have test samples already assigned. The new document types (Mortgage Agreement and Wage and Tax) do not have any test samples assigned yet therefore there are no green icons.



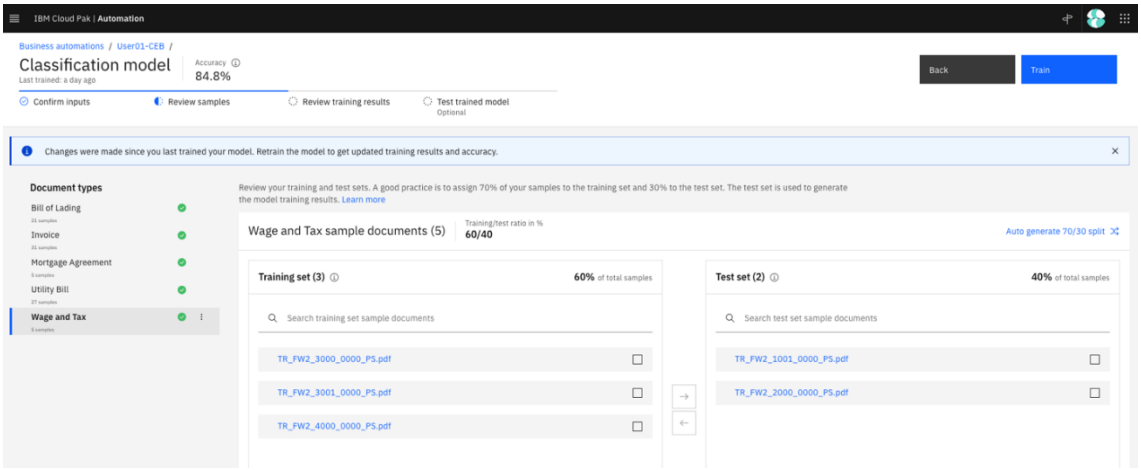
_4. **Click** on **Mortgage Agreement** and **move** the **first two documents** to the **Test set** by **checking** and **click** on the **arrow** in between columns.



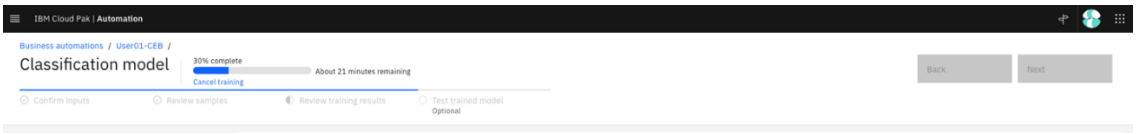
_5. **Click** on **Wage and Tax** under Document types. This time let the ADP system **Auto generate** the 60/40 split to the test set by **clicking** on the **Auto generate split** link.



The suggested split is 60/40 – that is, 60% of the available sample documents should be used for training, and we will validate the training results with 40% of the sample documents. This split is only a suggestion, and we can adjust it, but 60/40 is a good starting point.



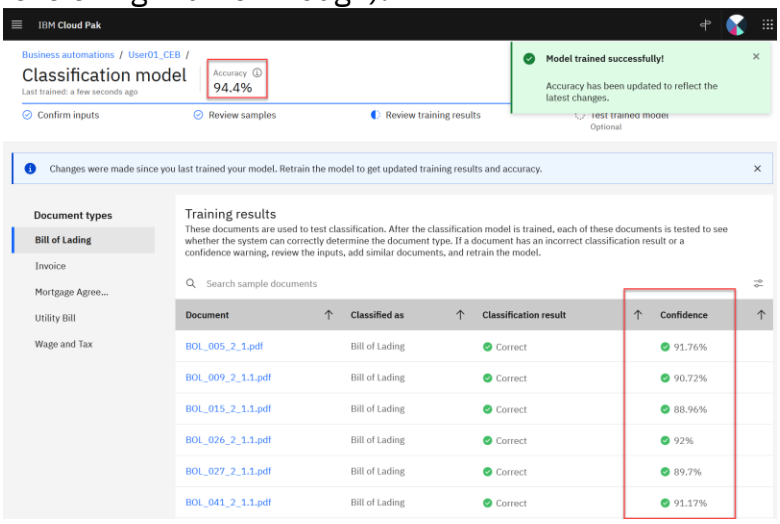
_6. **Click on Train** and then **Confirm** to launch the training. This may take a several minutes. You will see a progress bar showing how the training progresses.



Once complete, you will be able to see the training results.

What's happening: All the samples are run through multiple machine learning algorithms. These machine learning algorithms learn from the ground truth, the association between the sample documents (the OCR text) and the document types. The yielding models are then evaluated with the documents in the test set. The model-predicted document types on these documents are compared with the human-provided answers to compute the accuracy. The top three accurate models are presented to the user, with the most accurate one being selected by default.

You should see something like the following (the actual accuracy and confidence levels might differ though):



_7. **Close** the green **notification**. **Click** on **each** of the **document types**. Notice the confidence levels. You can notice either or both Mortgage Agreement or Wage and Tax have a confidence of low. Low confidence means we probably need to add more documents to our document class to get better confidence values.

You can easily see where the system may be struggling with Wage and Tax and Mortgage Agreement. You should look for document types that don't match the actual file or have a low confidence. Remember the more documents you give to train, the better the results.

_8. **Click** on **Next**. This is the **Test trained model** page. Here you can try and test other documents to see if they classified correctly. This step is optional but would be useful to try out the AI model to determine whether additional samples are necessary.

_9. **Click Done**

7.1 How do I improve my results?

7.1.1 Option 1 – Add more samples

To improve results, you would normally want to add more samples of the document ensuring they are clean and representative document to improve the system's understanding of the document.

_1. **Click** anywhere on **Document Types and Samples**

_2. **Click** on **Wage and Tax** type

_3. **Click** on **Upload**

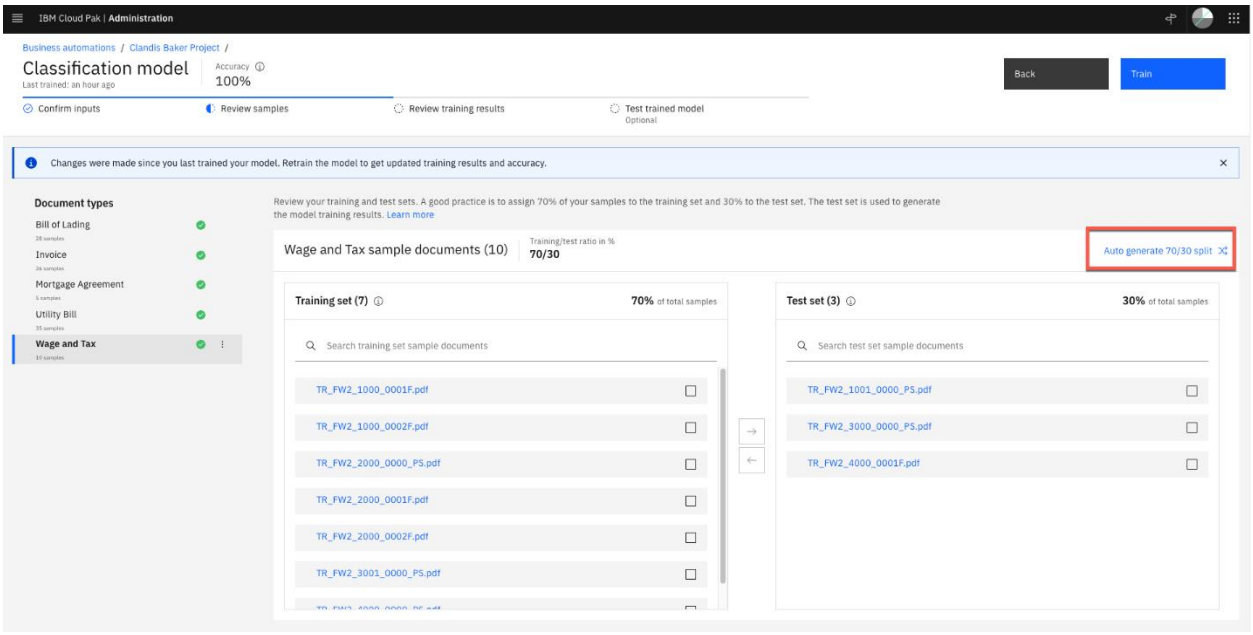
_4. From the zip files you downloaded and unzipped earlier upload all the files from the directory ***Group 2 - Classification Results Increase Set***. Wait until the status for all documents is Ready.

_5. Go back to the **Build** tab and let's retrain the **Classification module** again

_6. **Click** anywhere on **Classification model**

_7. **Click** on **Wage and Tax**

_8. **Click Next** button followed by **click** on the **Auto generate split** link.



_9. **Click on Train** followed by **Confirm** and wait until the training is complete

_10. Now look at the confidence score for **Wage and Tax**. They should have improved considerably compared to before you added new documents.

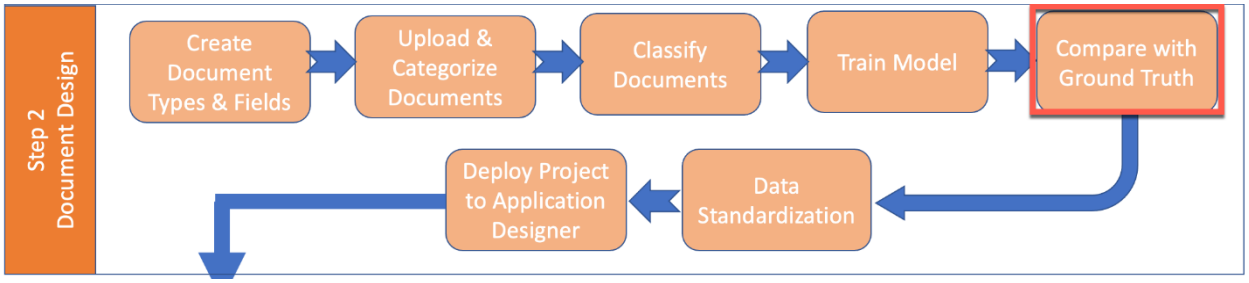
_11. **Click Next** and then **click Done**

7.1.2 Option 2 – Review all uploaded samples

As pointed out before, the quality of the sample documents determines the quality of the results. Therefore, in general:

- Remove those that are not a clear representation
- Remove those that are poor quality documents
- Carefully confirm that none of the samples contain multiple document types in the file. This is a common occurrence. A document is listed as a Purchase Order, but in the back pages, also contains other document types in that same file. This confuses the system.

8 Data extraction



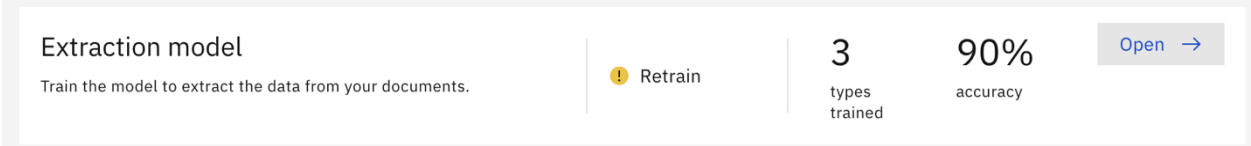
At this point, we have defined a document type, told the system which fields we want off the document and trained the system on how to recognize (classify) the document. In the Data Extraction portion of the lab, we will upload new Wage and Tax documents to Document Processing and see how our earlier configuration of the document type and related fields are working. This is comparing a new document extracted elements with the ground truth.

Once we open the Extraction model, we will be presented with details on how to perform the retraining. There are five basic steps – Review samples, Add fields, Teach the model, Review the trained model, and Test the model.

_1. From the guided configuration screen, **Click** anywhere in the **Extraction model** box

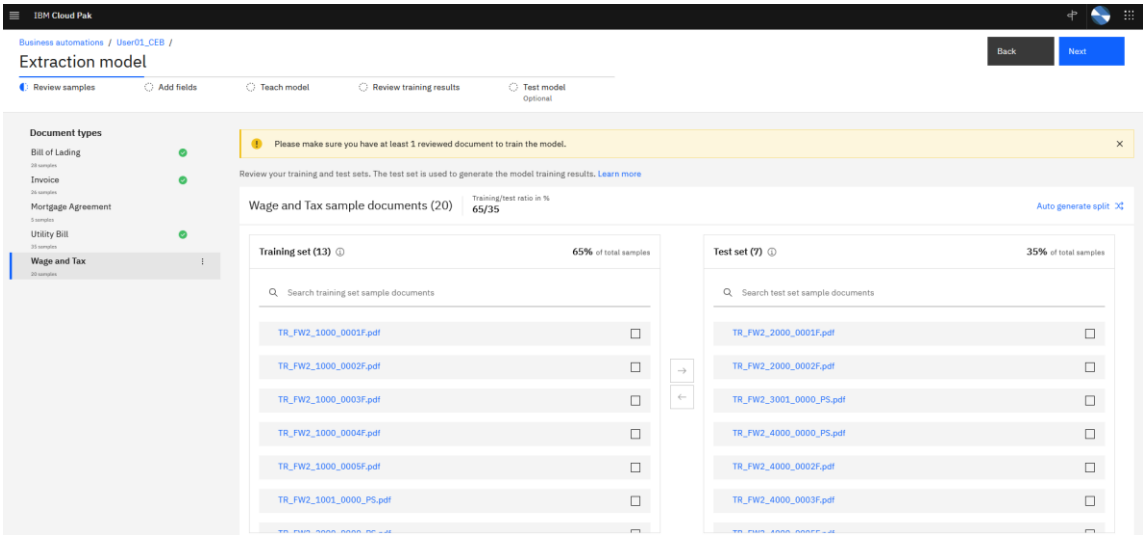
> ***Note:*** *The status will be reset to Retrain if ADP detects something may have changed. This is just a reminder that if you indeed changed something, you may benefit from retraining the model.*

_2.



Next **Click** on the **Wage and Tax** document type under the Document Types section

Like in the classification step, ADP needs to have the documents divided into a training and test sets. In general, *deep learning*-based AI requires a larger number of sample documents to achieve a reasonable result. But since our environment does not have GPU, deep learning is not turned on.
You should have something that looks like what you see in the following screen shot.



Again, lets train with an Auto generate spilt. **Click Auto generate split**.

_3. **Click** on the **Next** button at the top

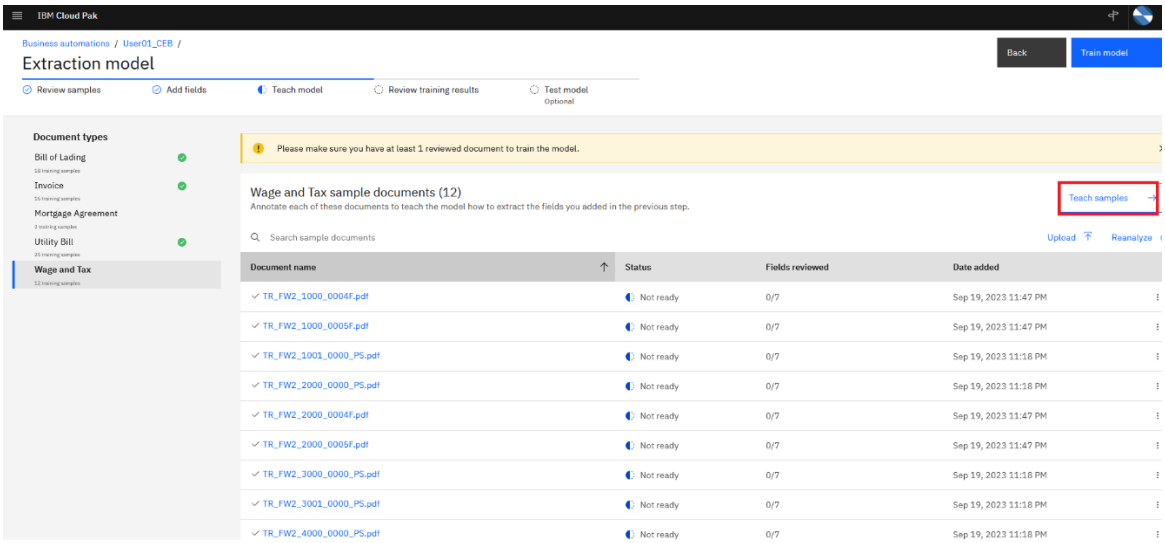
_4.

You will now be on the *Add fields* step. If there were more fields to add we could do it here. But since we have already added all the fields needed, proceed to the next step.

_5. **Click** the **Next** button. You are now at the *Teach model* step.

_6. Teach the model is where you will spend most of your time. We can see that our documents are “not ready”, so we’ll need to teach the model with new documents.

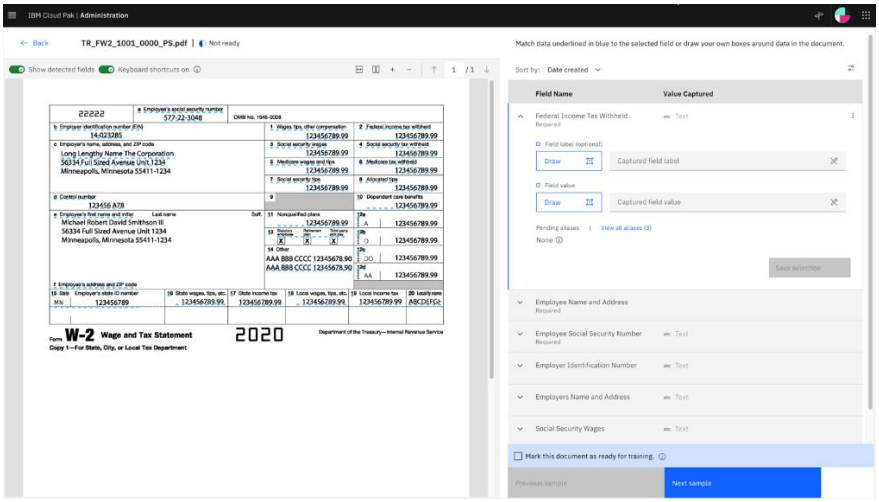
Click on **Teach Samples →**



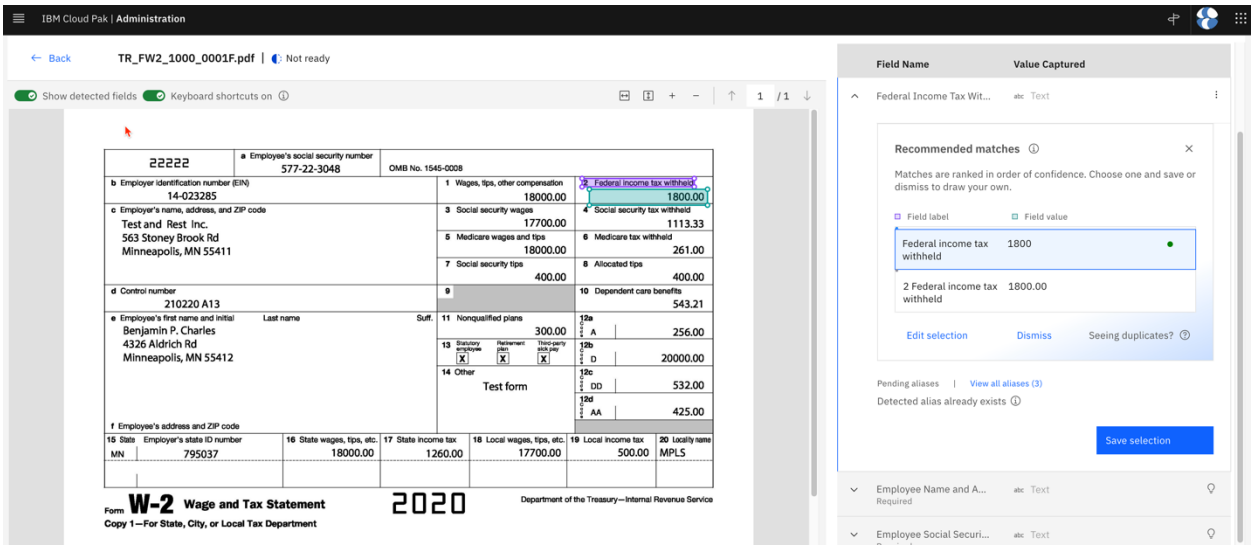
Note: *Your individual results may vary based on the exact documents you upload, how you configure your fields etc. Therefore, general guidance is given here versus exact step by step instructions.*

We will now review the fields that were extracted, correct any that may be wrong and add others.

7. You should now see the field data extracted by the system. Nothing has been trained yet. All it is doing is using the field name and aliases we entered when we created the document class to locate data. Now, you need to correct and improve the model.



Note: You may see different results than shown on the image above. Depending on how the algorithms interpreted the results you could see either type of extraction.

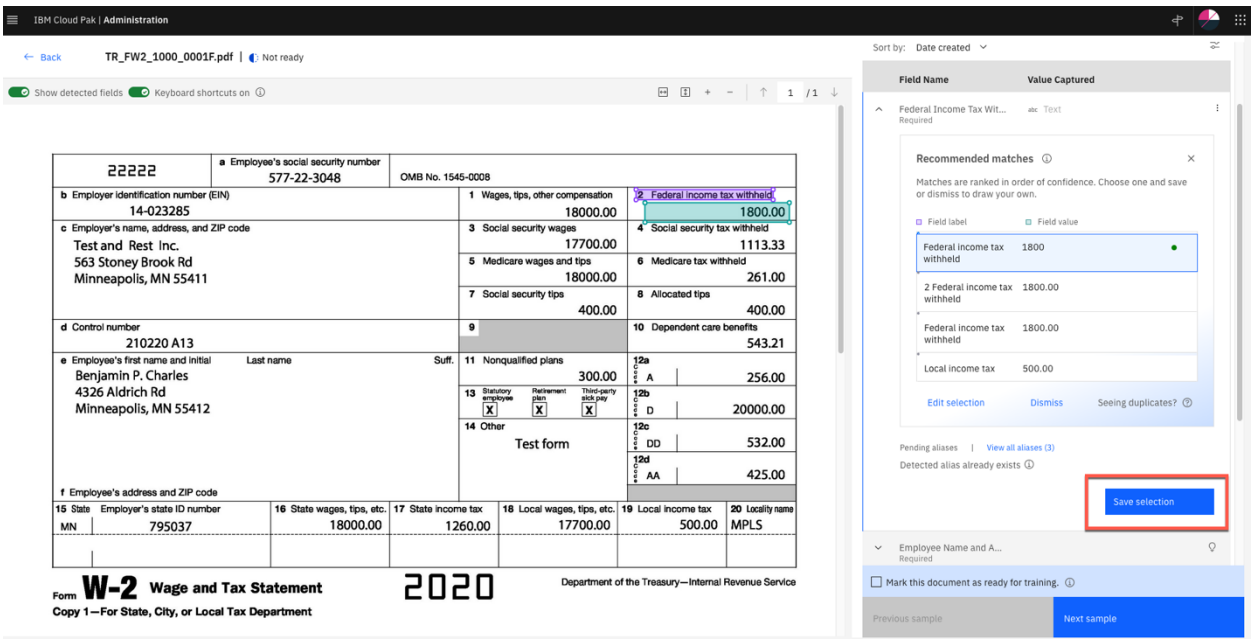


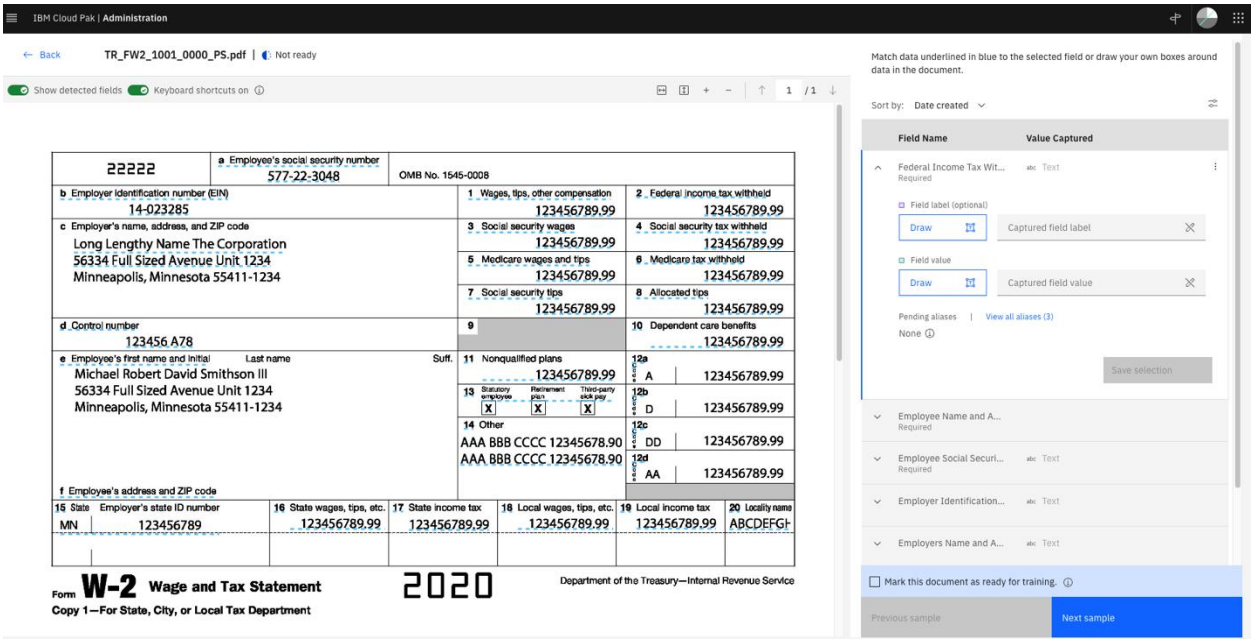
Let's spend some time showing how to go about correcting these issues to help the system learn how to extract the values accurately.

8.1 Correcting extracted values

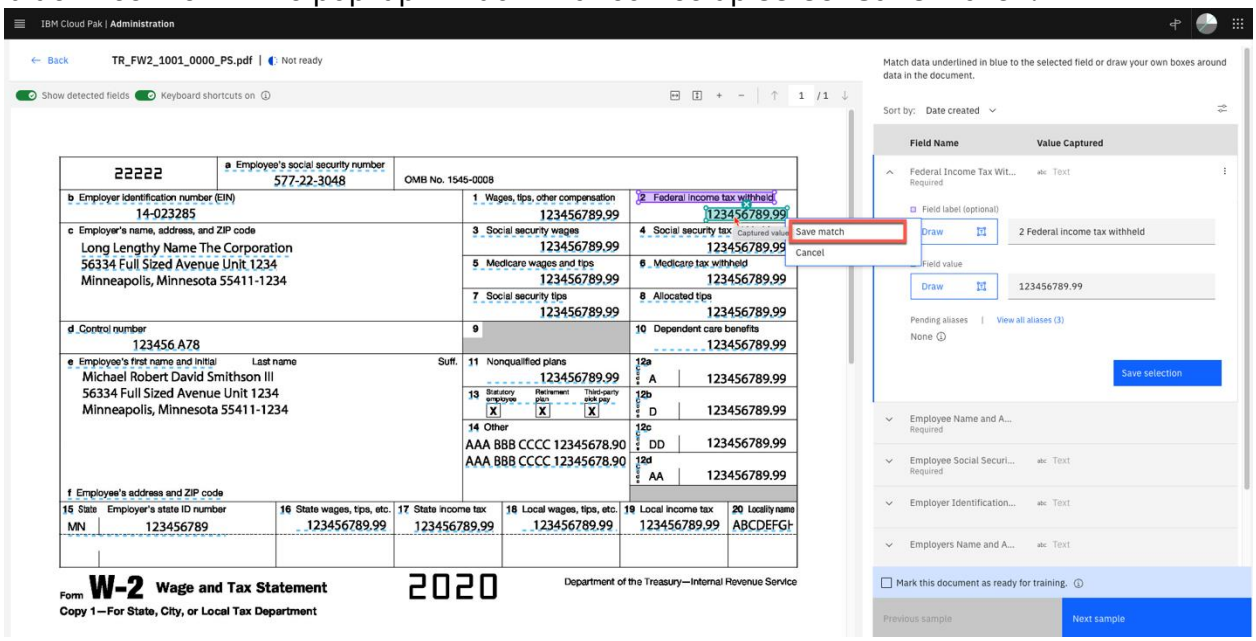
Let's start with the Federal Income Tax withheld field (i.e., the first one in the 'Fields to extract' list). Again, you may see different results based on your forms and how the different algorithms behaved on that particular document during extraction.

_1. ADP may have already preselected the first field like in the first screen shot below. But ADP can also show the characters it recognized on the page with blue lines (second screen shot below) If your result is like the first screen shot then **Click** blue button **Save section**. Otherwise, if you got blue lines **Click** on the **number** below the heading **"Federal Income tax withheld"** in the image.

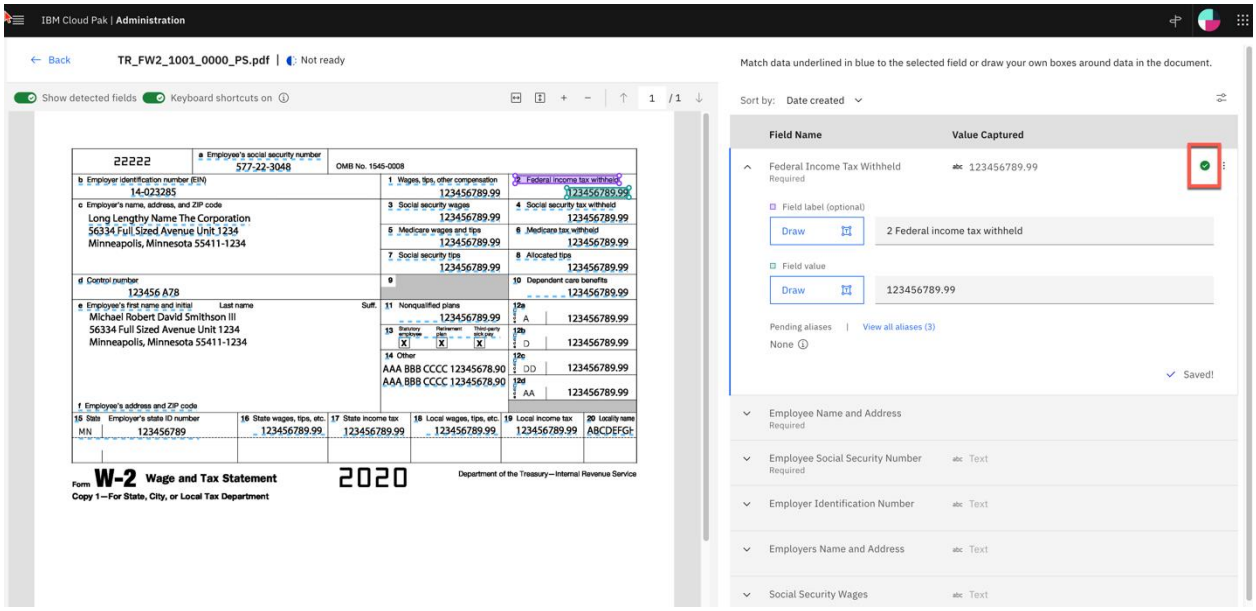




_2. Again, depending on your specific results. If ADP was able to find the field and will ask if you want to save match of value captured along with the field label. **Select Save Selection.** Otherwise, if your results were the recognized characters with blue lines then in the pop-up window that comes up **select Save match**.



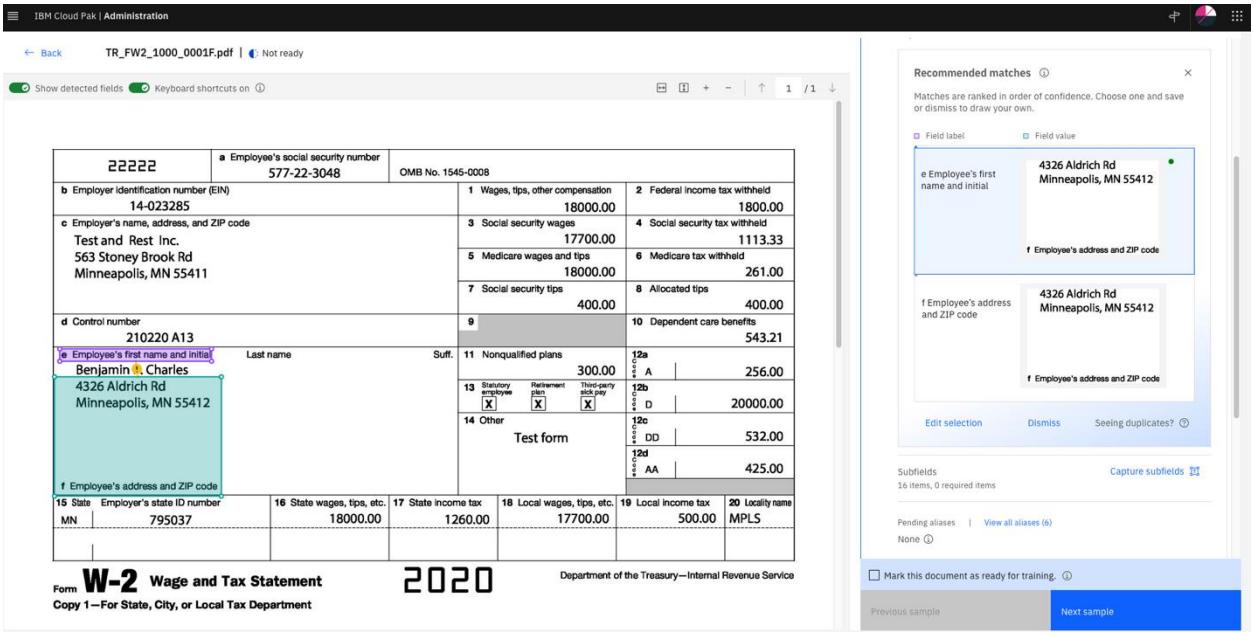
Notice a green check mark signifies this field is complete.



The 3 ellipses next the green check mark allow you to clear the data or update ADP to there is no field with this data in the current view.

_3. Move to Employee Name and Address field by clicking in the grey area on that field name. In our two possible outcomes depending on the algorithms. ADP did pick up the name but missed the address. Or the algorithm may have picked up the address and not the name. Or it may have gotten the correct field.
If the field is not correct **Click** on the **Dismiss** button.

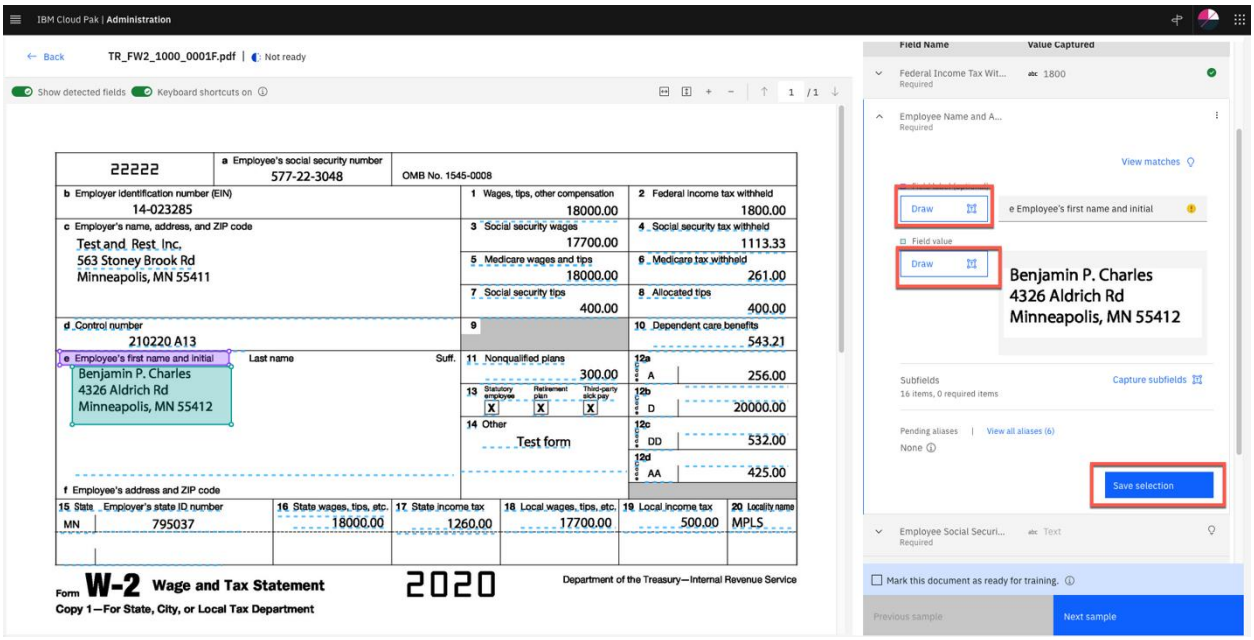
Now under the Field label **select Draw** button and using your mouse grab or lasso around "**Employee's first name and initial**".



If you got the blue lines, you would notice that only the “e Employee’s first name and initial” have blue marks. In this case the values for name and address where not located. Using Draw button and using your mouse grab or lasso around “**Employee’s first name and initial**”.

We are interested in getting the “Employee’s First Name” data and address for the field value. **Click** on the **Draw** button under Field value. Using your mouse select the appropriate values for Name and address (green box), then **Click Save selection.**

_4.



_5.

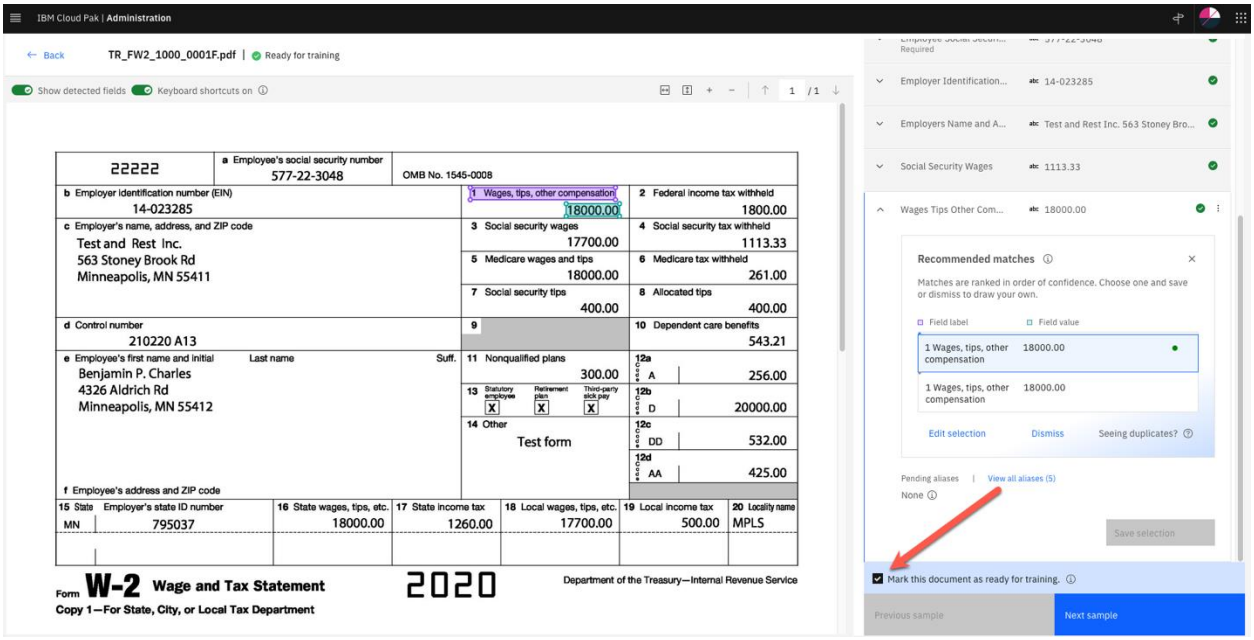
For the Employee Social Security field if it looks good, **Click** on **Save selection**. Or

_6. if the blue lines are present instead **select** the value displayed to populate the field and **Click Save match** then **Click** on **Save selection**.

Continue to process for the remaining fields, using either method as described above, clicking on the *Save* selection if ADP picked up the correct field label and

_7. field value or select the blue line values to populate both the field label and field value or finally if both fields are wrong use the *Dismiss* and use blue lines if Key Value Pair (KVP) is correct or drawing a box around needed label or value.

Once complete **check the box** next to “**Mark this document as ready for training**” at the bottom



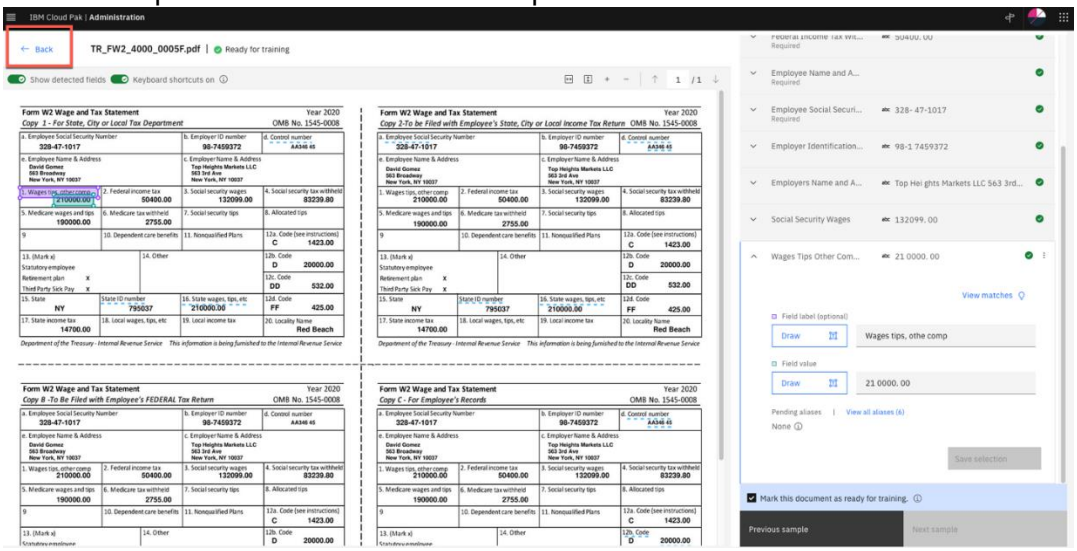
_8. Review ALL other fields carefully. **Do not leave any incorrect values**. You can adjust or delete values as needed by clicking on Edit selection. If you leave incorrect values, the system will assume they are correct and LEARN them as if they were good values.

_9. Repeat **steps for Next Sample**

Over the course of next few samples, you may find that ADP has extracted the wrong results, perhaps getting a value that is above when it should have been below. If this is the case and you pick you a blue underline data, but the results are wrong. Simply use the draw box for the Field Label and Field Value.

Note: *When completing the remaining documents, you may run across ADP finding the fields but perhaps on the second image or third image on the page. Try to keep all Key Value Pairs (KVP) on the same image.*

Once complete review of all the sample documents **Click** on the **Back link**

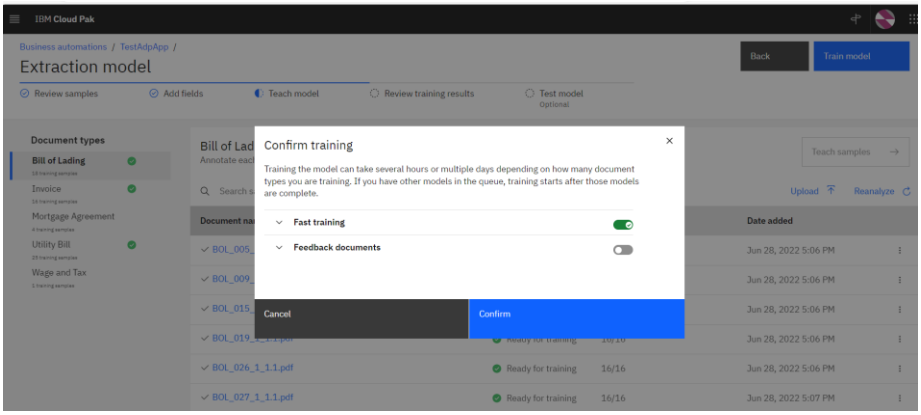
_10. 

8.2 Train extraction model

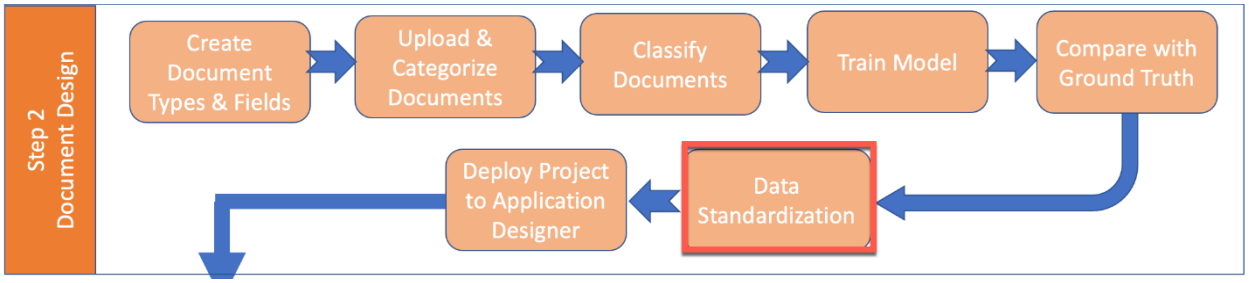
We will be performing the **fast training** in this lab due not having a GPU available in the environment. A GPU is only needed in a development environment and is not needed in either a production or runtime environment. The Deep Learning capabilities have been disabled on this training environment. You can find instructions in the Appendix for when you have access to a server with it enabled.

_1. **Click Train model** button

In the **Confirm training** dialog coming up, switch **Fast training!!** on before clicking the **Confirm** button. Then the training will take several minutes (good time for a break). If fast training is not switched on it could take days without a GPU.

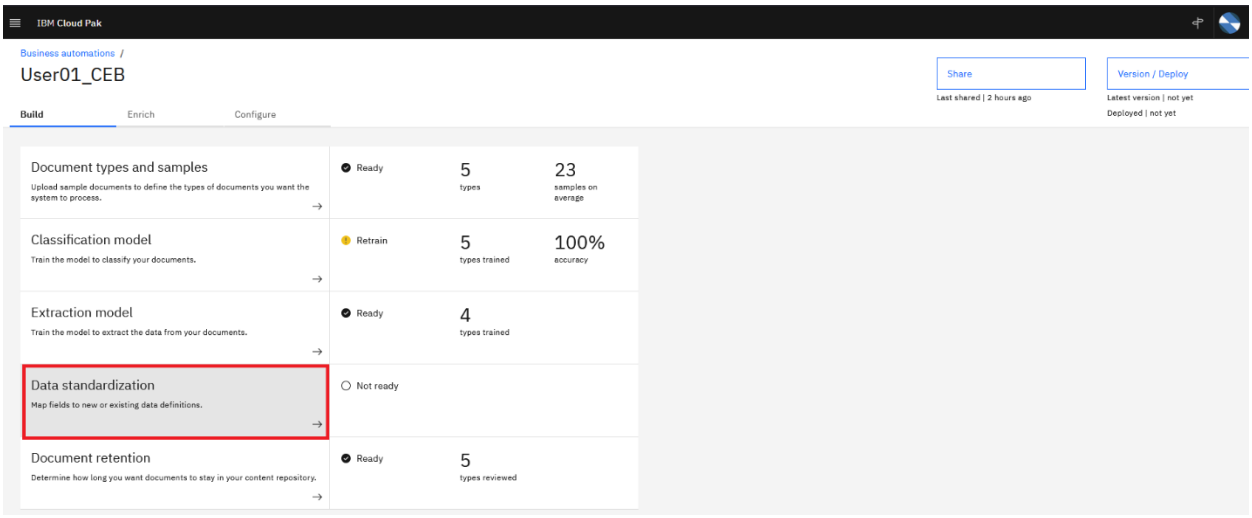


9 Data standardization

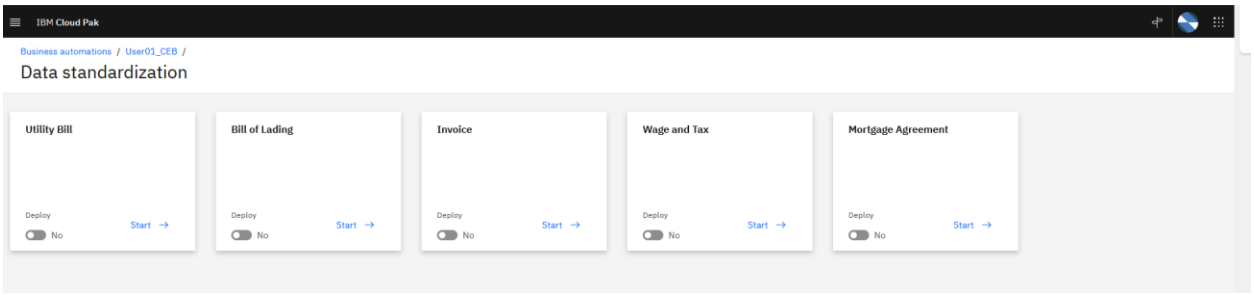


Next, we may need to standardize the data that will be presented in the user interface and how it will be stored in the FileNet repository for example. Data standardization is the process of defining attributes for a data field in a standardized way. This is done using data definitions. These definitions can be used across projects, and across different applications within the Cloud Pak for Automation. Each data definition has a title, description, and a datatype. We can also set a data definition as required or not. When a document is ingested into ADP, it results in a list of Key Value Pairs' (KVP) for that document. The Designer maps some of these KVPs to fields and teaches the model on how to extract the fields from the full list of KVPs. The designer then maps some of those fields to data definitions for a particular document type. Only the fields that have been mapped to data definitions will become Content Process Engine properties.

1. Return to the guided configuration flow and **click** anywhere in the **Data standardization** box

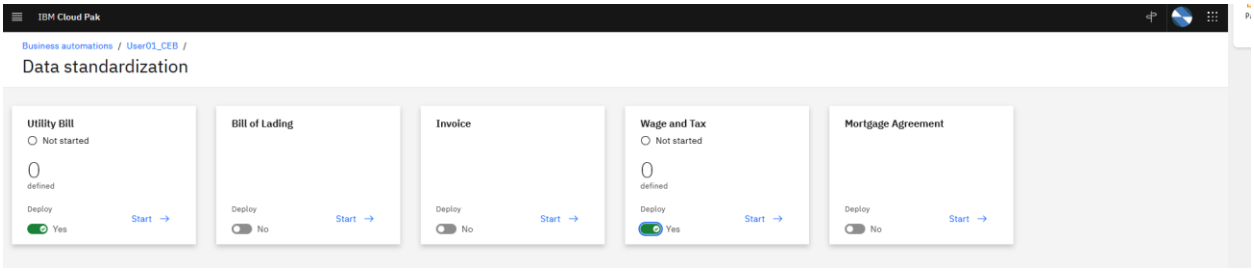


Here, you will see a list of available document types. Only the ones which have ***Deploy*** turned on will be visible in the verify interface and will have fields stored in FileNet.



Ensure the **Utility Bill** and **Wages and Tax** are toggled to **Yes**

_2.

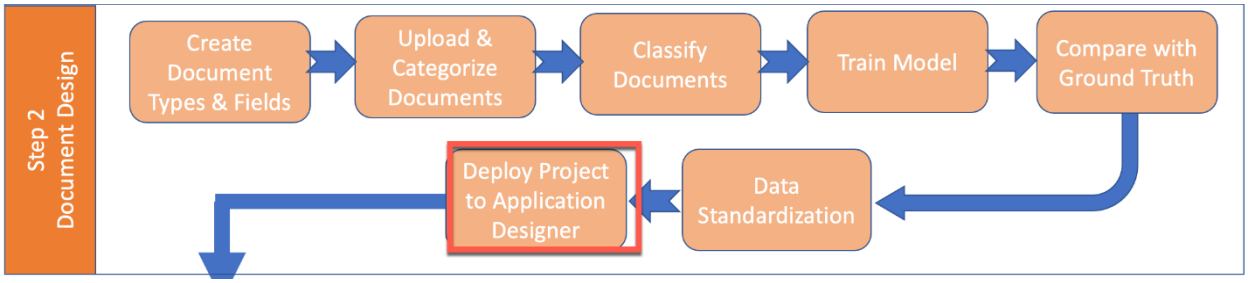


Click on **Start** → on either of the selected deployments

_3. This is where we begin defining the data filed attribute definitions. You could create a new data definition and configure them. We will NOT be creating/defining any data fields for this lab.

_4. Return to the guided configuration screen by **Clicking** on **<your project>** name at the top of the screen

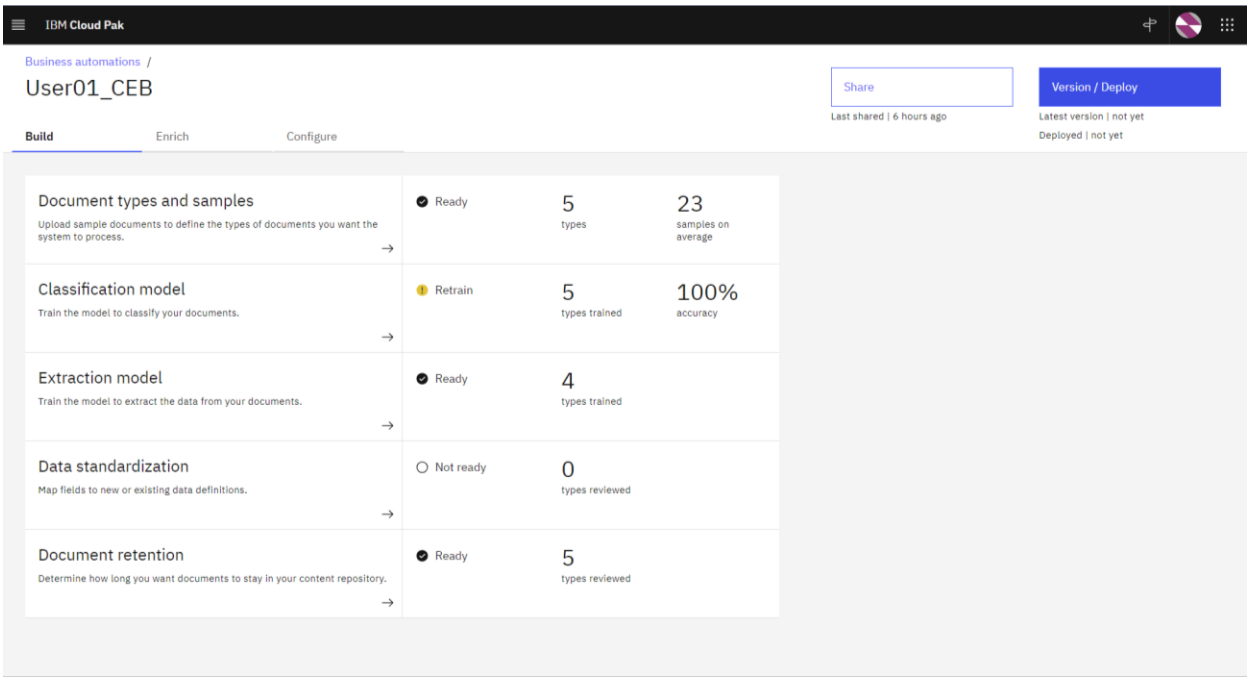
10 Version and deploy your project



At this point in our project, we have defined a document type, labeled the fields we want from the document, trained (classified) the system to recognize the document type, reviewed the extracted fields we wanted and standardized (mapped) the document fields to our output.

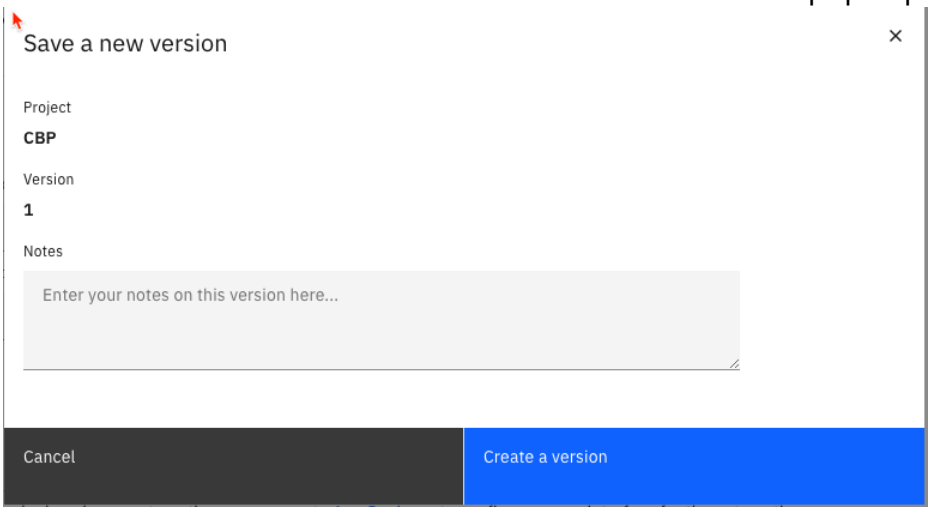
Now that we completed the configuration of the content extraction project, we need to save and deploy the design project to the application side. This will allow you to test your project using a client runtime interface.

1. If not already there, return to the guided home screen by clicking on your project name. Then **Click Version / Deploy**.



Click **Save a new version.** A *Save a new version* window pops up.

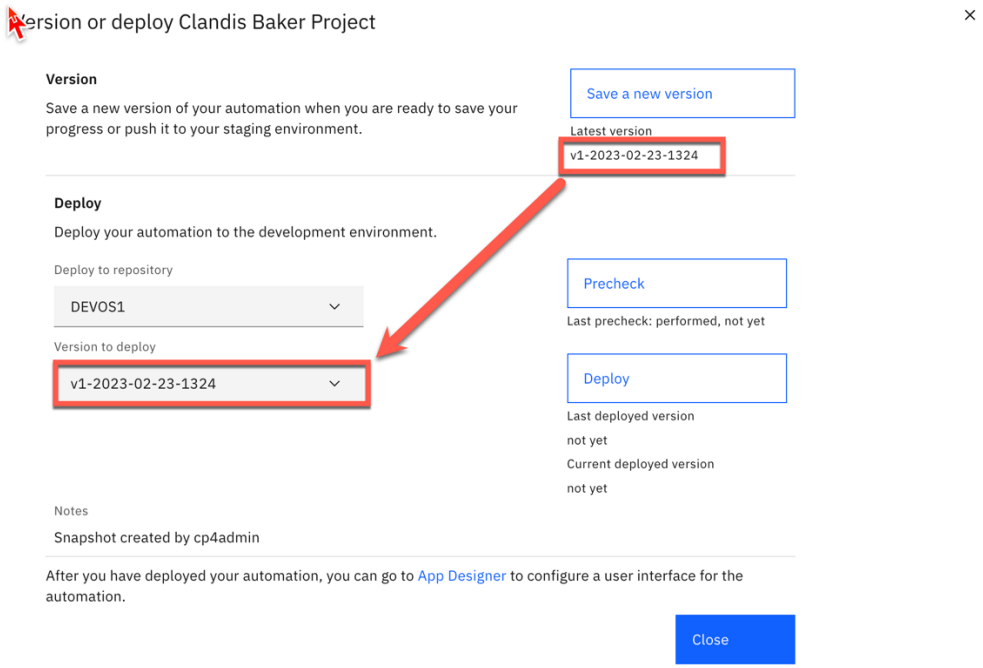
_2.



Click on **Create a version**

_3. Once the version is saved, you should see the version in the Version to deploy drop

_4. down list

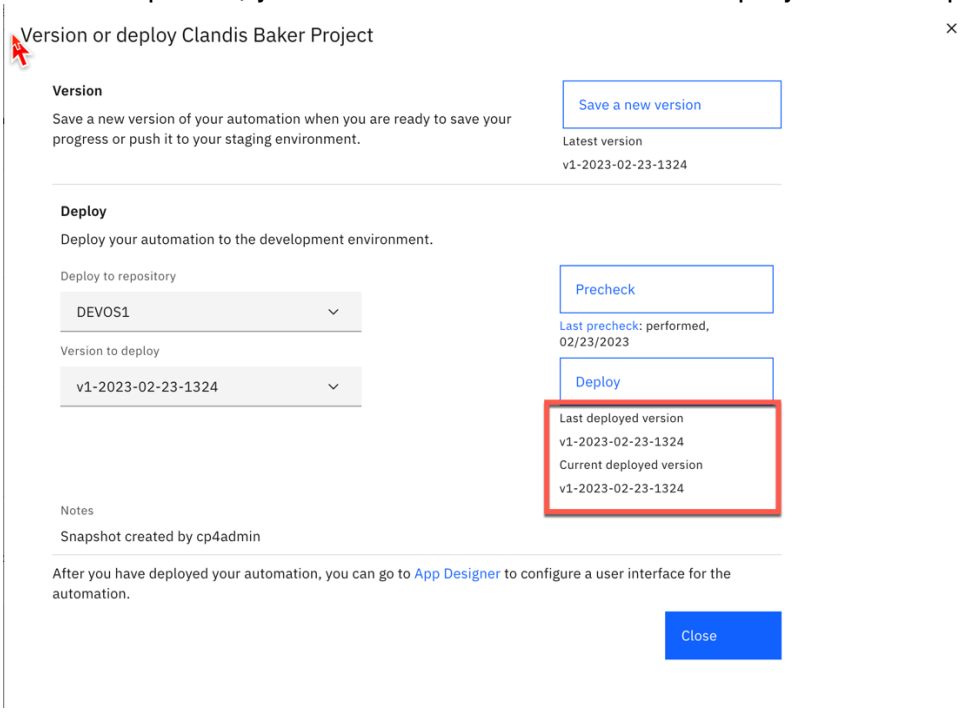


_5.

... also, in the top corner has the “Latest Version”

Click on the **Deploy button**. This will also take a minute or two to deploy.

Once completed, you should have a notice that the project was deployed.

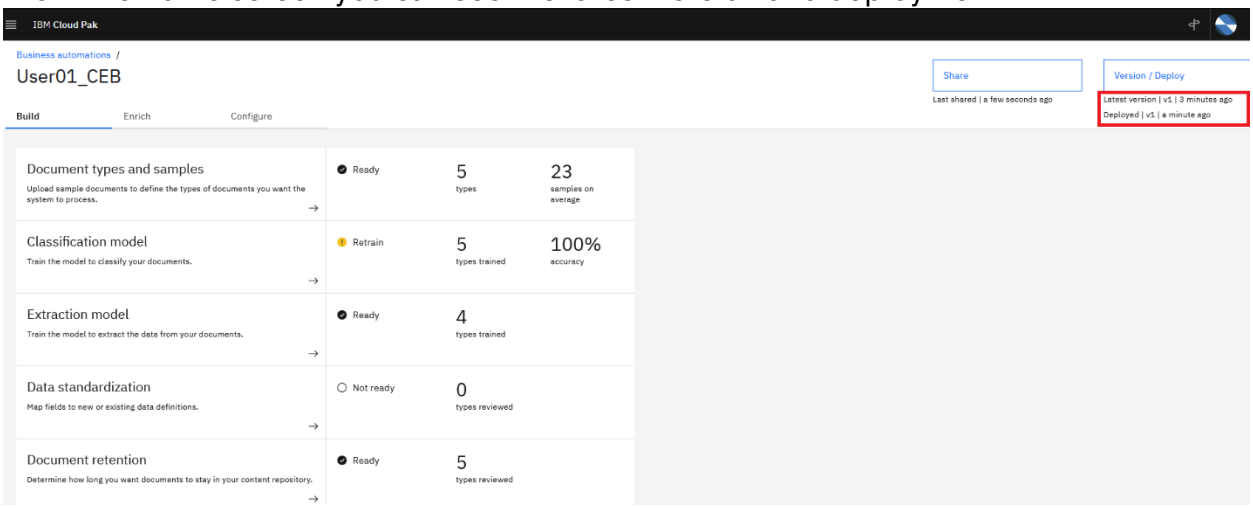


Note that you do not have to remain in the deploy screen while it is versioning or deploying. You can always click the button and then go back into any other screen if you like. It will run in the background. If you do this, just keep an eye on the top right of your screen for deployment status.

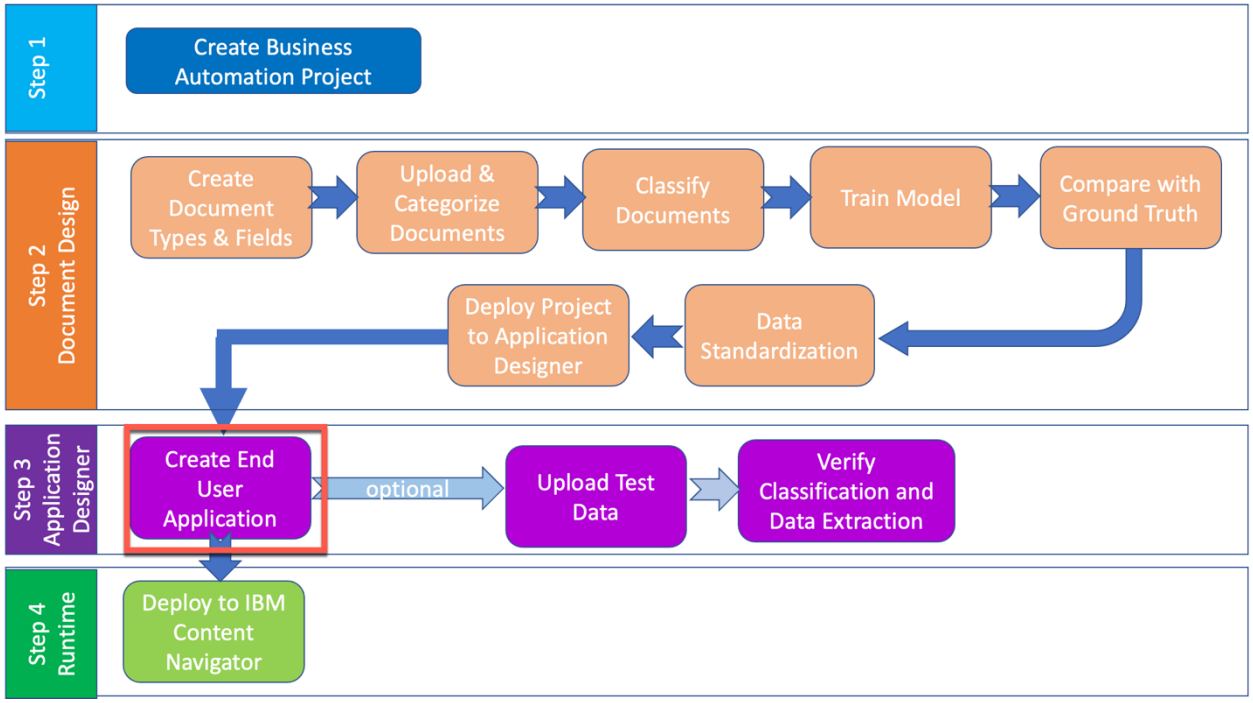
_6. **Click Close** button

Once deployed, proceed to the next steps.

From the home screen you can see the latest version and deployment



11 Application designer



At this point we have designed or built a project that consists of document types, data or filed types and methods to extract the desired data. The next major section of this lab is to build the user interface using the Application Designer. IBM provides two application templates for Document Processing

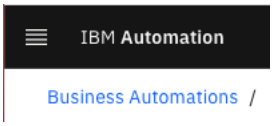
1. Batch Document Processing template – used to process batches of documents
2. Document Processing Template – used to process single documents

The lab will have you create a new batch processing application. We will quickly explore the various tabs in the interface, preview what the IBM Content Navigator (ICN) client would look like using the Preview feature and then publish our application to ICN where we will process a batch of documents.

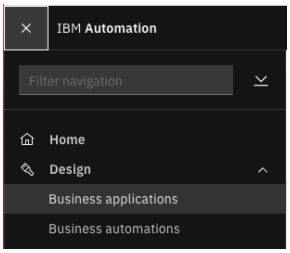
_1. Changes to the application itself will not be in the scope of this lab.

11.1 Create your Runtime Application.

Return to the starting screen by **clicking** the **hamburger** in the top left

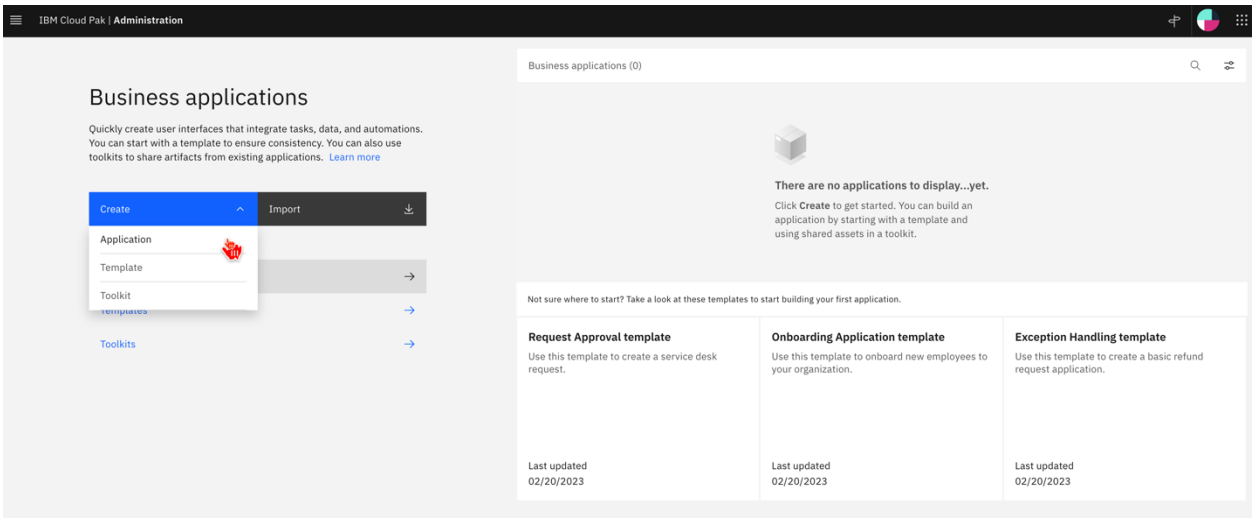


and selecting **Business Applications**

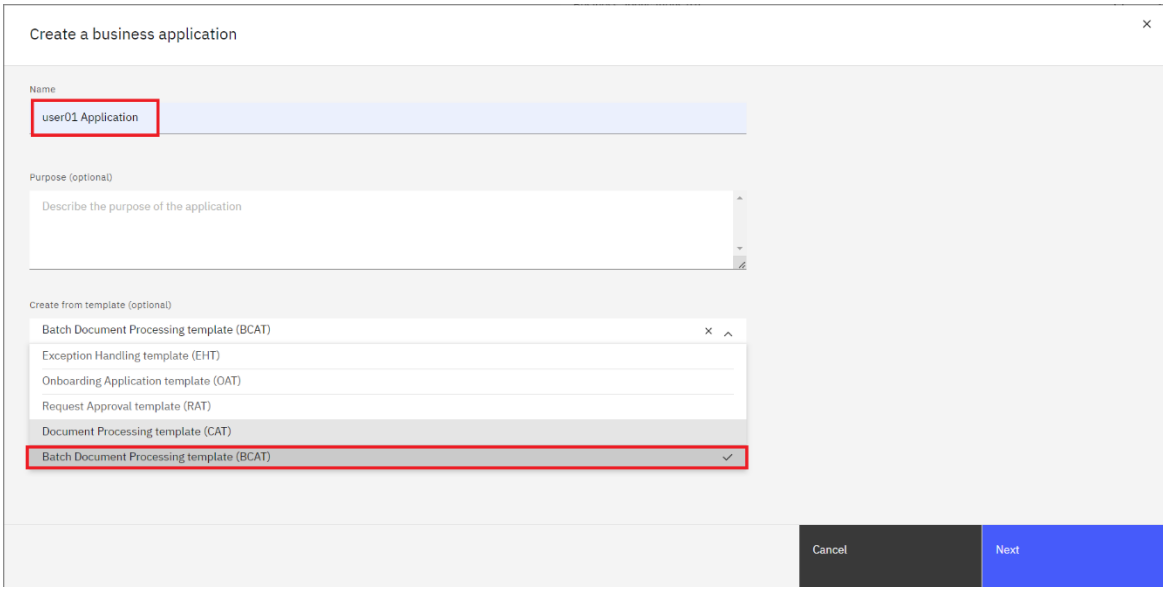


From the **Create** drop down list, **select Application**

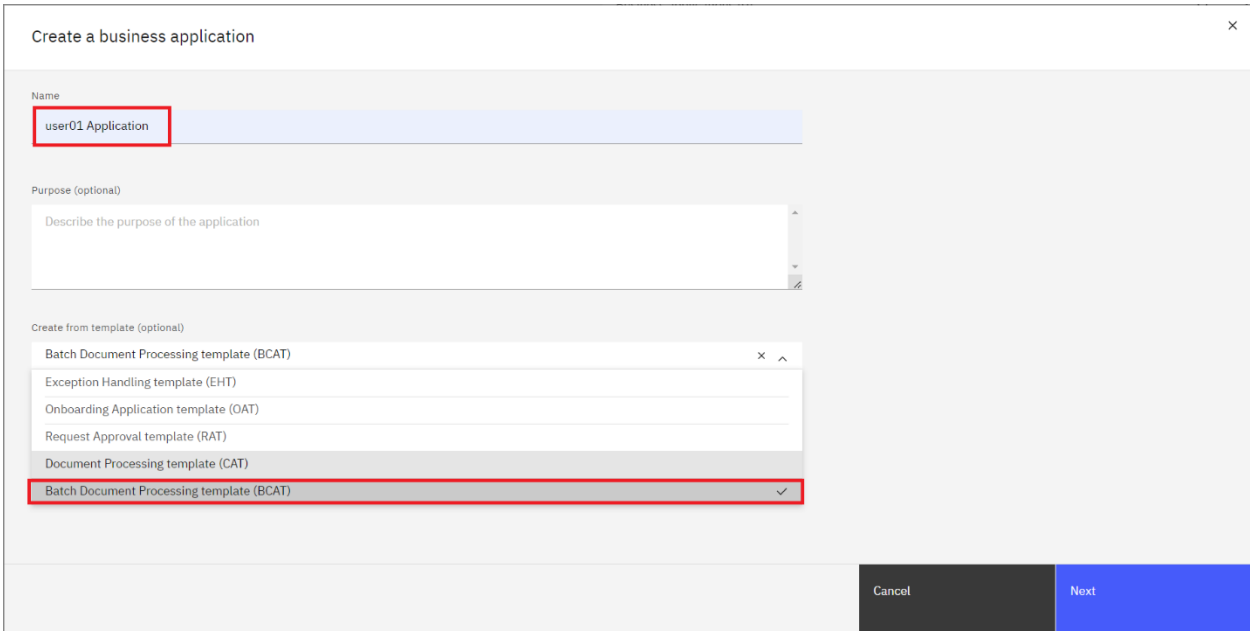
_2.



_3. Select **Enter your <*application name*>** in the Name field



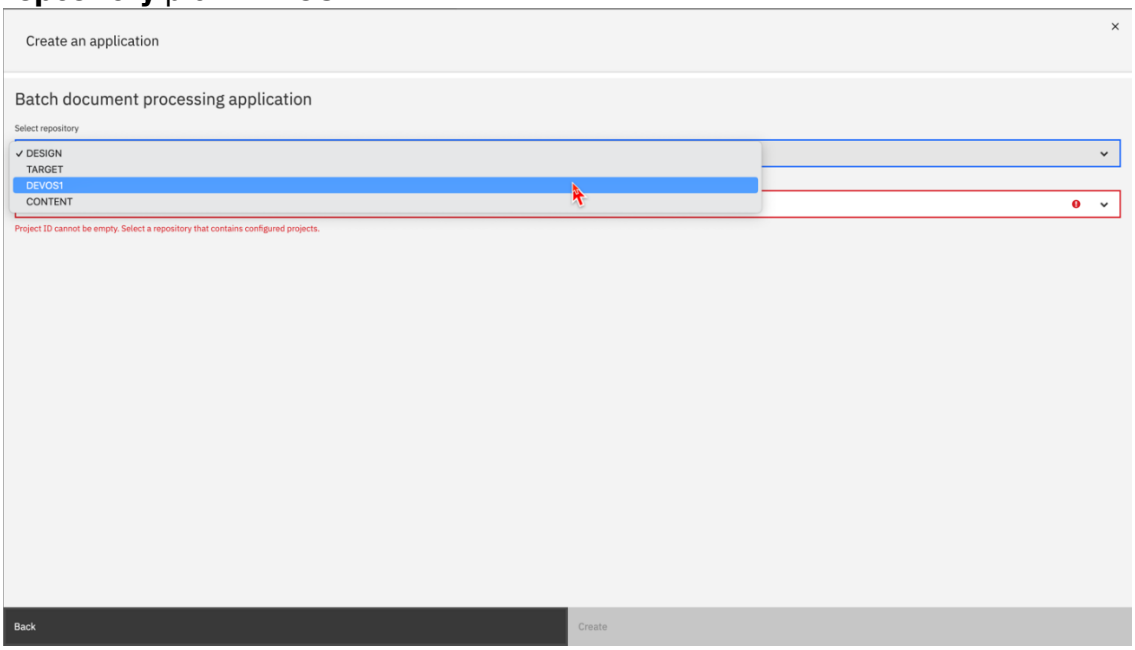
_4. In the Create Form Template in drop down **select Batch Document Processing template (BCAT)**



You could have selected the Document Processing Template if you only wanted to process a single document at a time, but in this lab, you will process several documents in a batch.

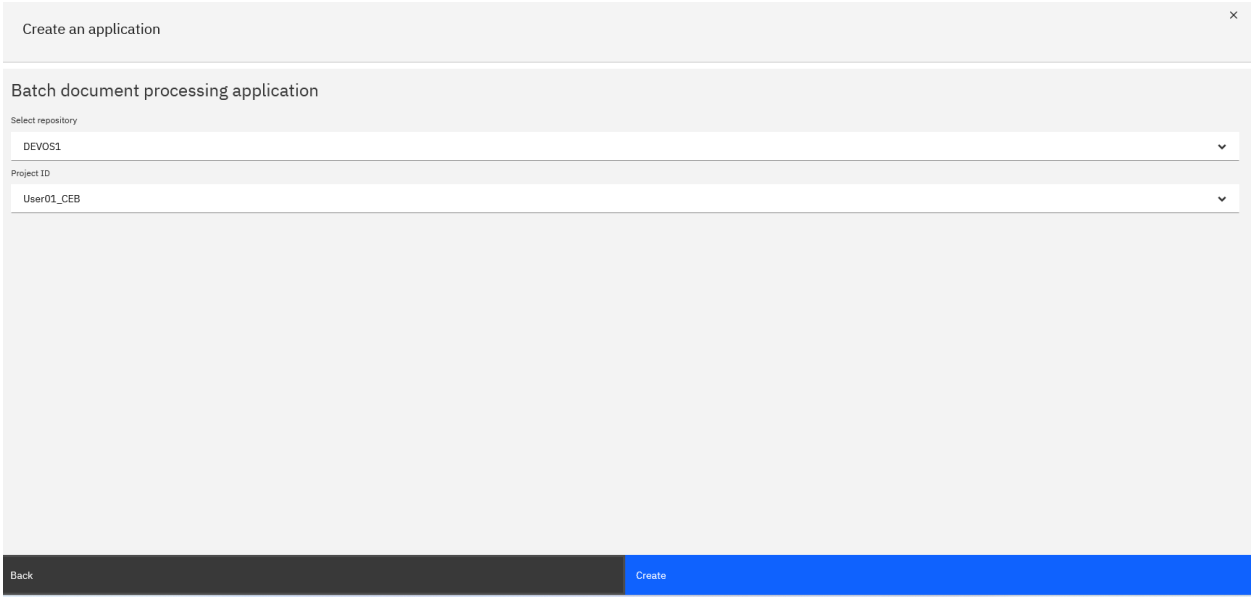
_5. Click **Next**

_6. You will be presented with the Create an application window. In the **Select repository** pick **DEVOS1**



In the Project ID drop down **pick *<your project name>*.**

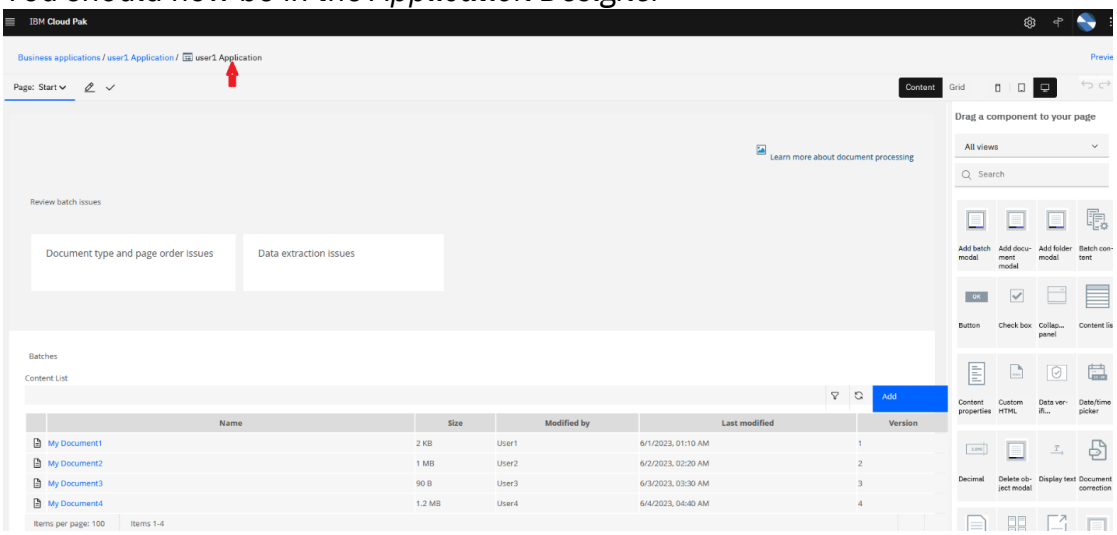
_7.



Note: *It may take a minute or two before this update and you can see your project.*

_8. Click **Create**

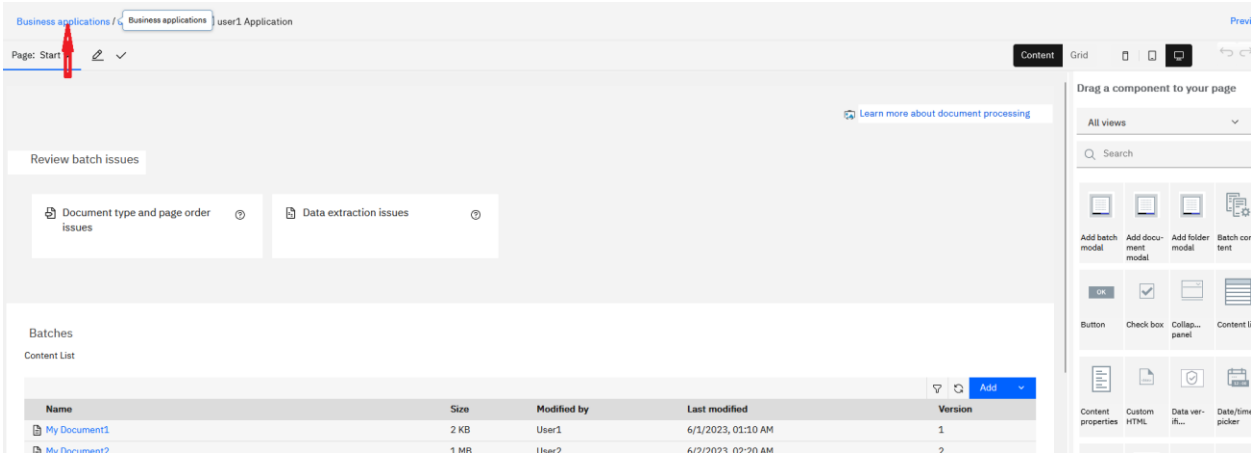
You should now be in the *Application Designer*



Batch Document Processing template (BCAT) has all the necessary pages and configuration to start using the application. Using this designer user interface, you have the option to further customize the application, such as its page design or actions, to fit your requirements.

Click on **Business applications** breadcrumb at the top

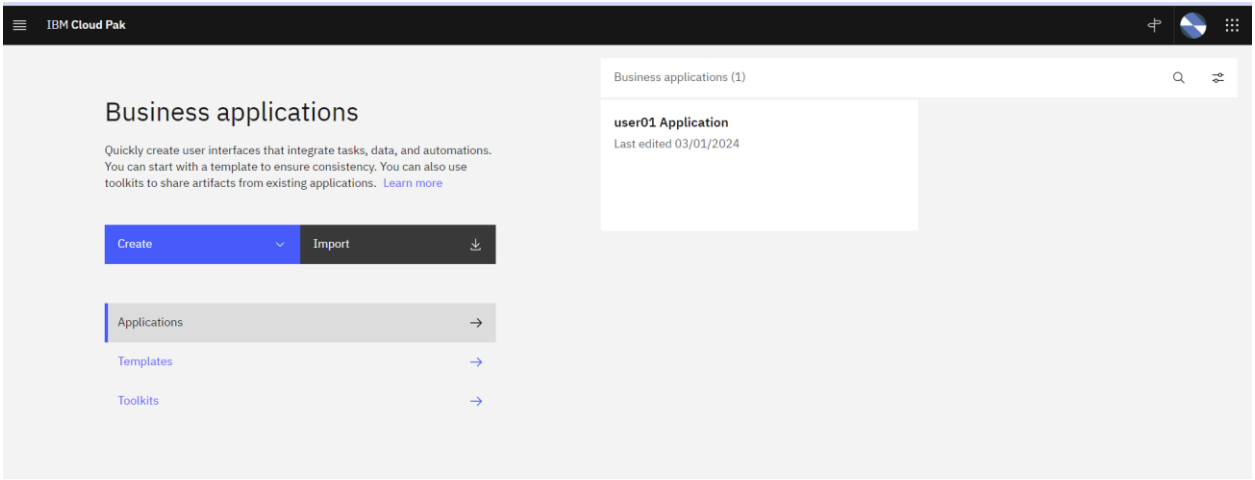
_9.



> ***Note:*** *It may take several seconds up to multiple minutes to build and display the current configuration of the interface. In case the screen does not load properly the first time, try to reload the whole browser window.*

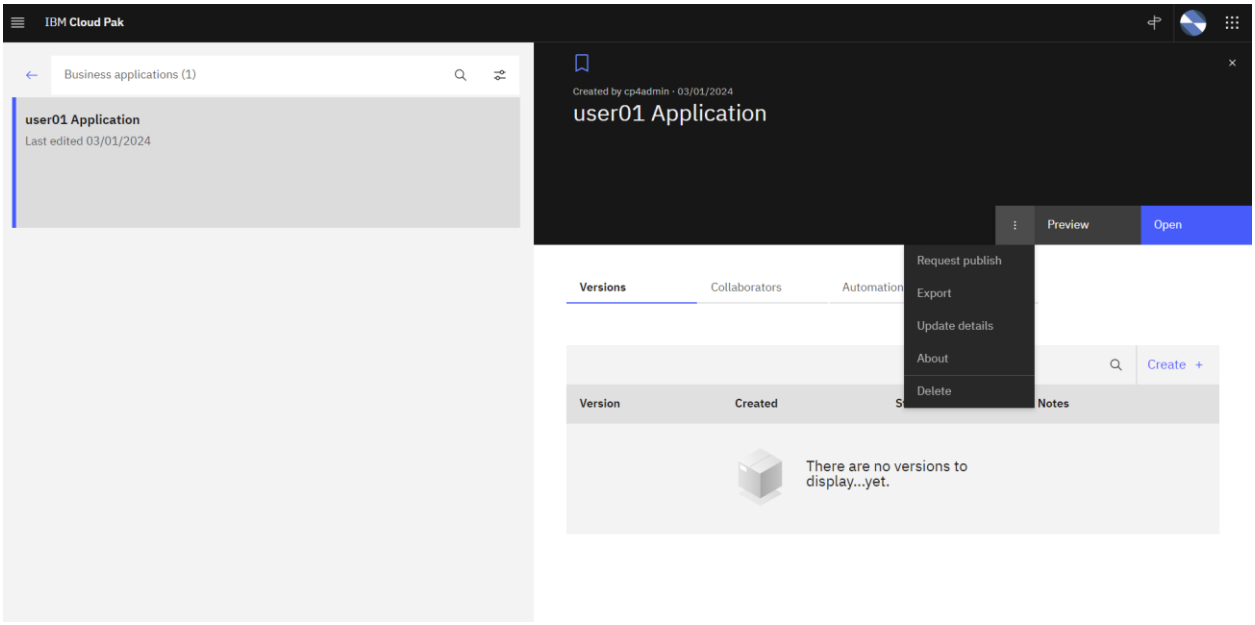
_10. If you hover over any of the applications on the right, the respective box will turn grey, and a Preview and Open link will become visible. Clicking Preview would let you test the pre-configured interface. Clicking Open would open the designer for the application where you can modify the look and feel and modify its features.

Click anywhere into the grey box, but not the Preview or Open link. This brings you to the details of the application.

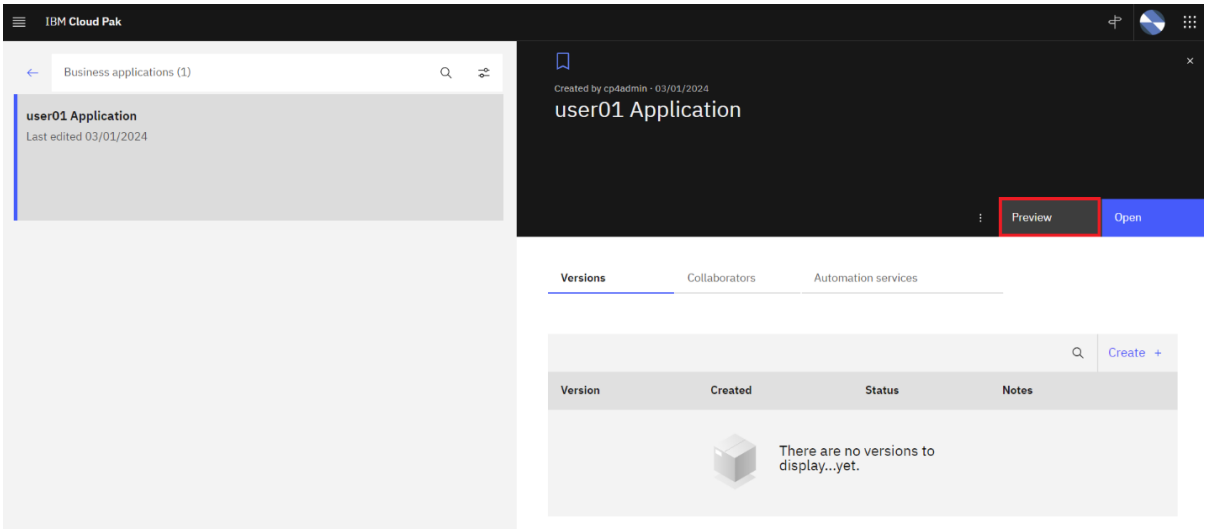


From this screen if you **click** on the **3 dots** you could for example export the application or delete it

_11.



_12. Now **click** on **Preview**



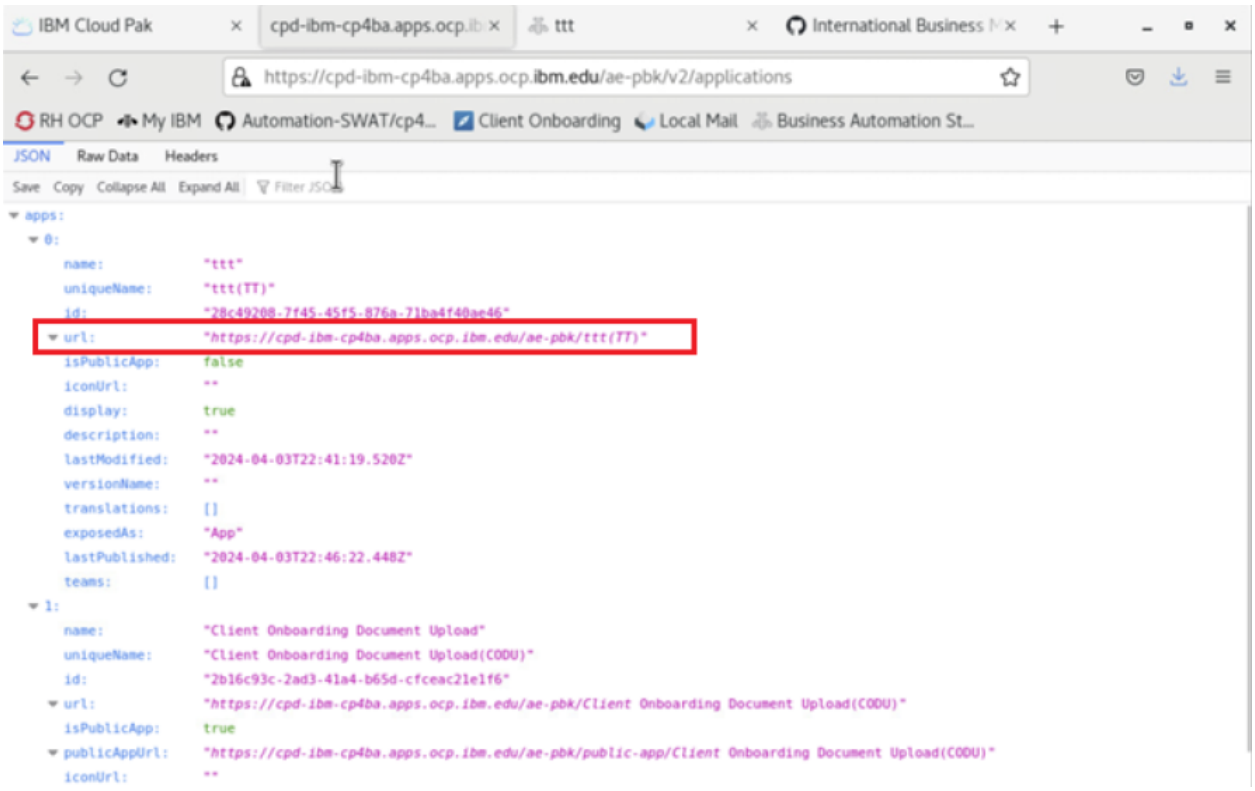
Note: *You may have a popup blocker turned on in your browser. Your browser will need to have this option off for the Preview.*

The Preview allows you to validate the execution behavior of your application.

Previewing your application is a vital step in the creation process. You can preview your application at various points throughout your development. Maybe you want to preview a small interaction within your application or test the entire experience of your application after you complete development.

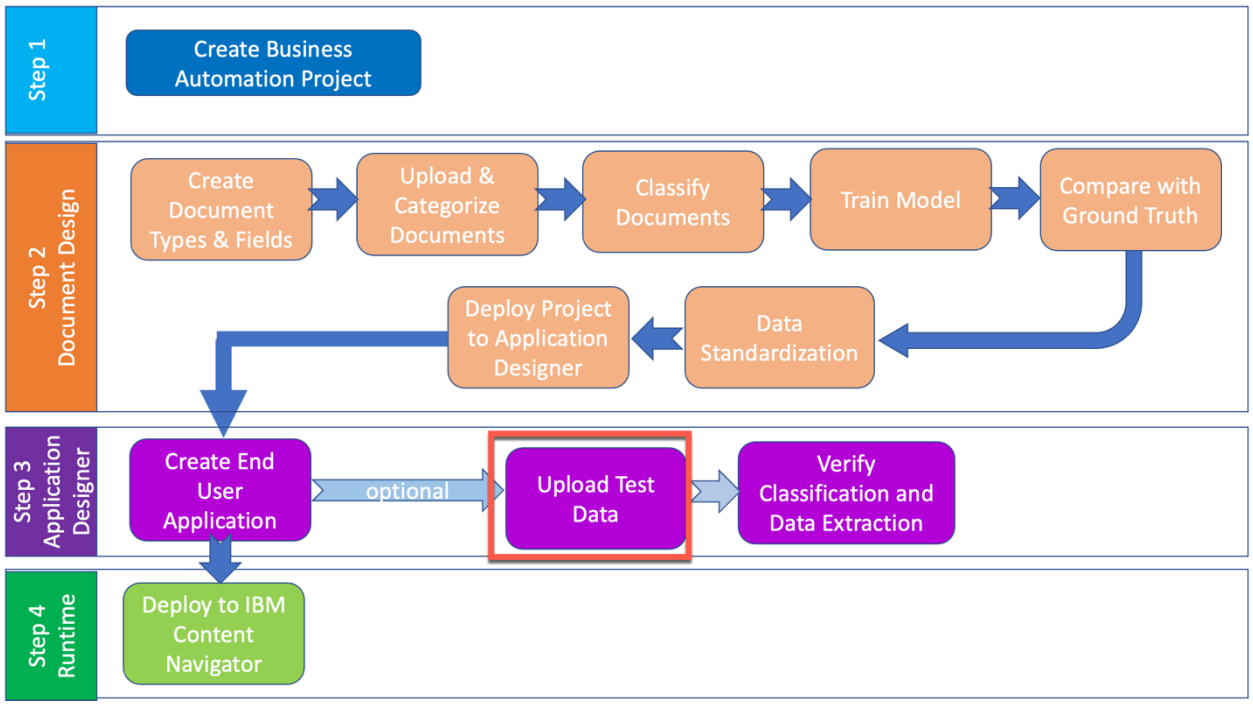
_13. In case the Preview takes time more than about 9 minutes or throws an exception like “Unable to connect to server”, open a new browser tab, copy the link from the Studio page (e.g. https://SERVER/bas/BAStudio/build/index.jsp...), and replace everything after server with **ae-pbk/v2/applications** (e.g. https://SERVER/ae-pbk/v2/applications).

This should look like below

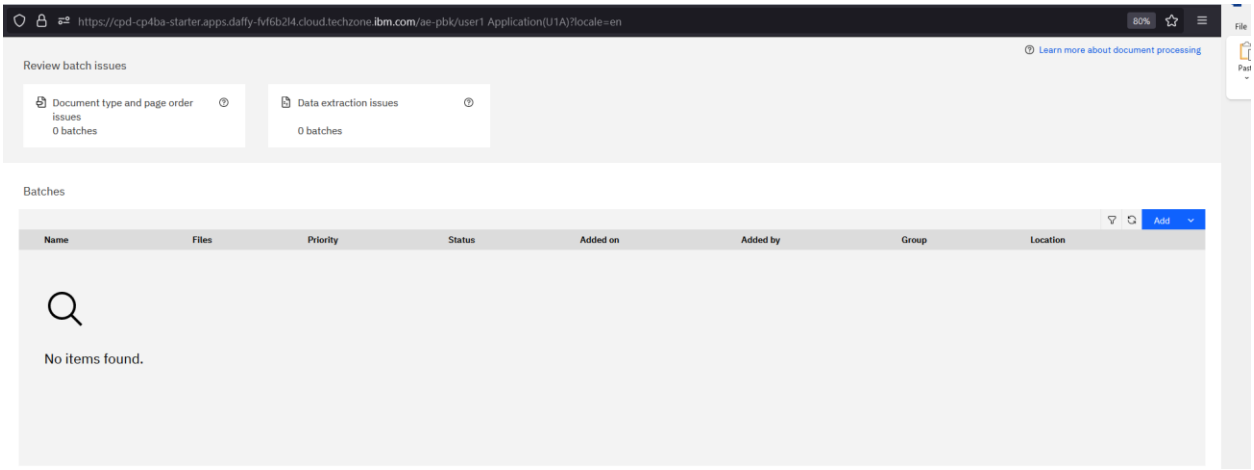


Here you can observe the ADP application that you created. When you refer to the above snapshot, you can see that there is an application called “ttt” and notice the url. Copy the url and paste it into a new browser window.

11.2 Upload documents for processing

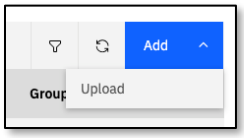


_1. Below pasted snapshot is the preview of the application. Normally, this preview should work in “incognito mode” of Chrome or “In Private mode” window of an Edge browser. Additionally, any popup blocker must be disabled or configured to allow open the pop-up window. You should be in the default application user interface for ADP. It opens a new tab/window in your browser.



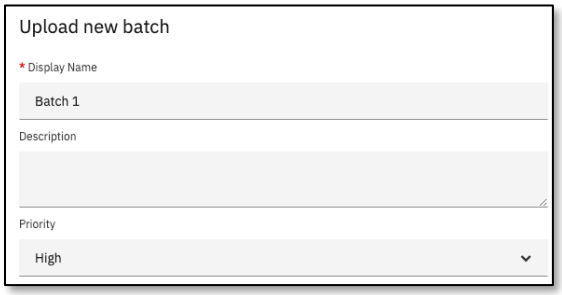
There are two key screens you will work with: “*Document type and page order issues*” and the “*Data extraction issues*”. First, we need to upload some test documents and have them processed.

Click on **Add**, then **Upload**



_2. Enter a ***name*** for your batch in the **Display Name** field and set the **Priority** to **High** as seen in the image below

_3.

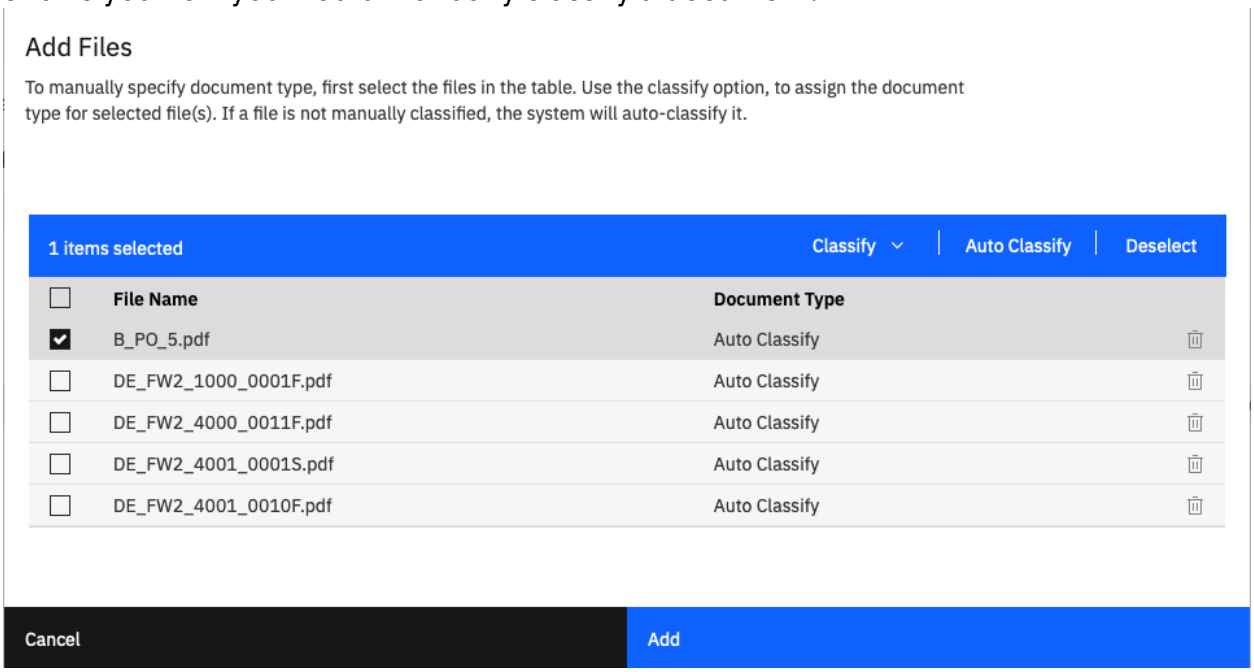


Click **Select files**

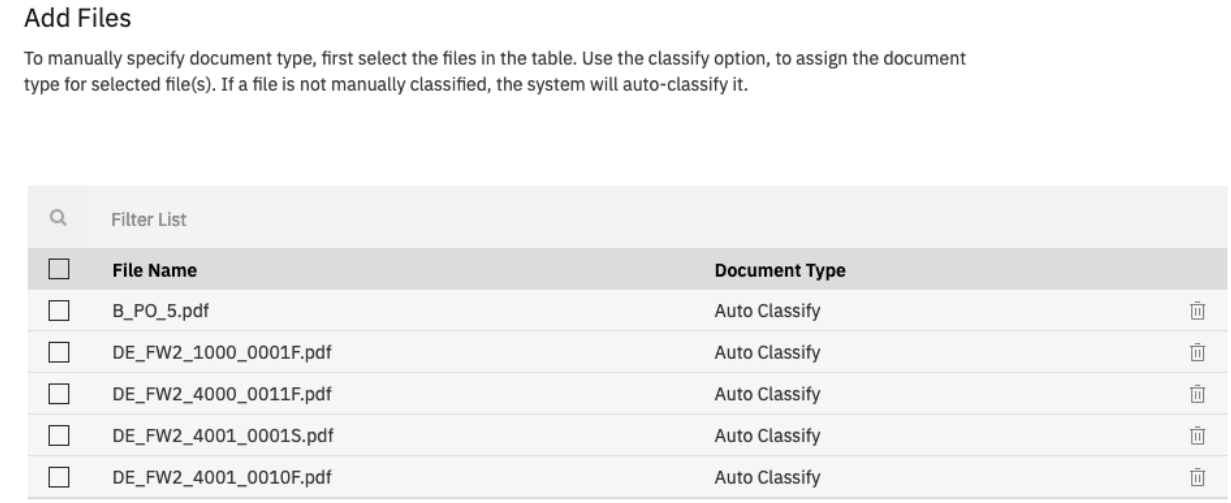
_4. Navigate to the samples folder previously downloaded from Section 2 and use the ***Group 3 - Runtime Demo Set*** folder documents. **Select all the files** in the folder.

_5. Click **Open**

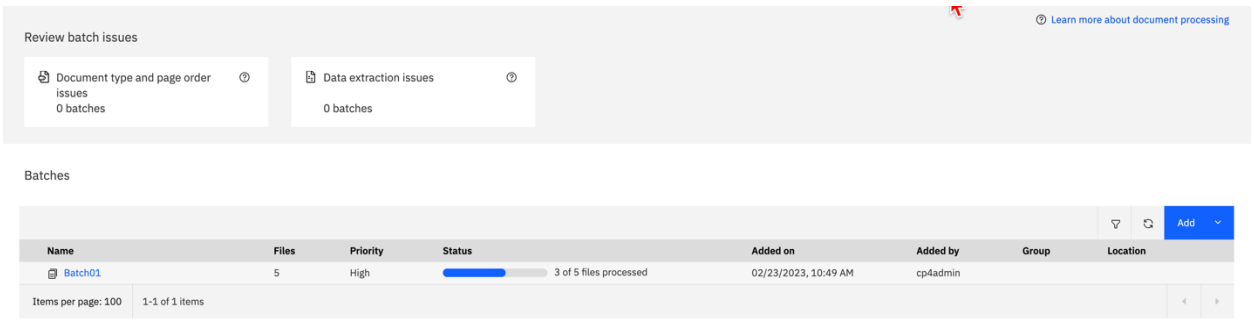
You will see a window that will give the operator a chance to manually classify the documents before they are ingested. By clicking on one of the files you will be presented with an option to manually classify the documents. The example below shows you how you would manually classify a document.



We are not going to do this but instead let ADP auto classify them.



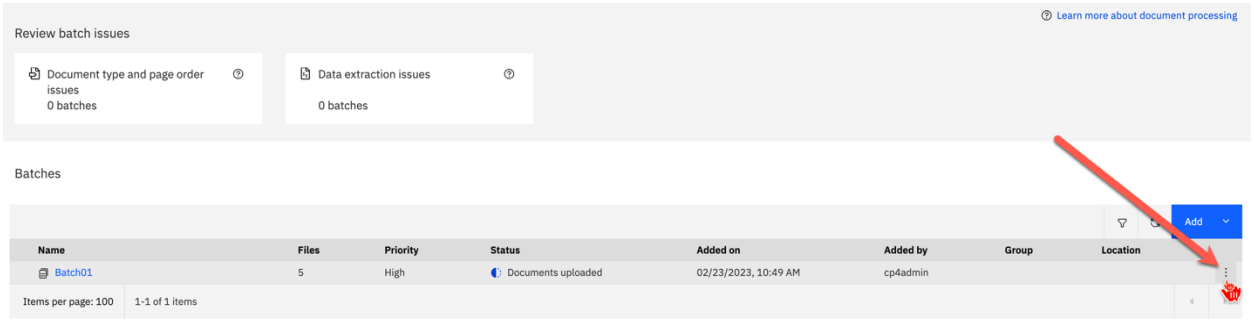
_6. **Click** on the **Add button**



_7. A progress bar will be displayed indicating when all documents have been uploaded/processed.

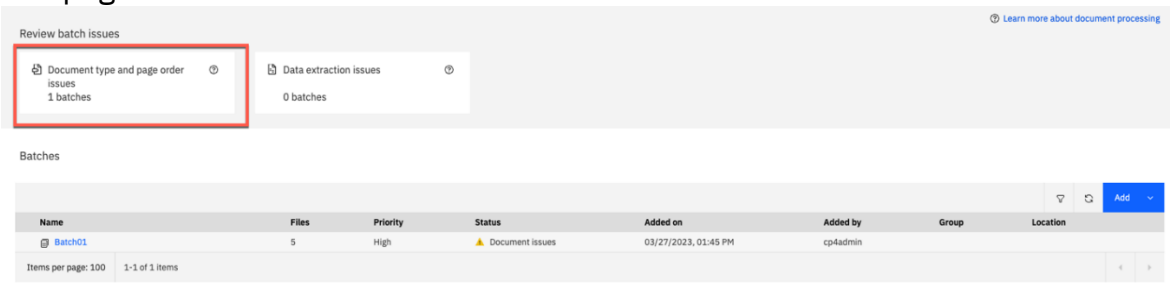
Click the 3 dots at the end of the line

_8.



Click Submit

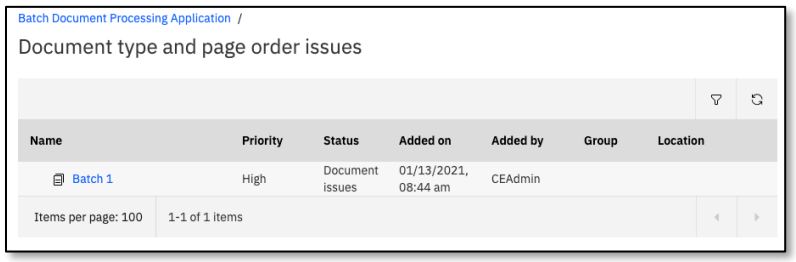
In the screen shot below, you see the status of the batch job is marked as having Document issues. Matching with that we now have 1 batch in the “Document type and page order issue” tile.



11.3 Correct any classification errors

Click on the **Document type and page order issues** tile to get to the respective batches

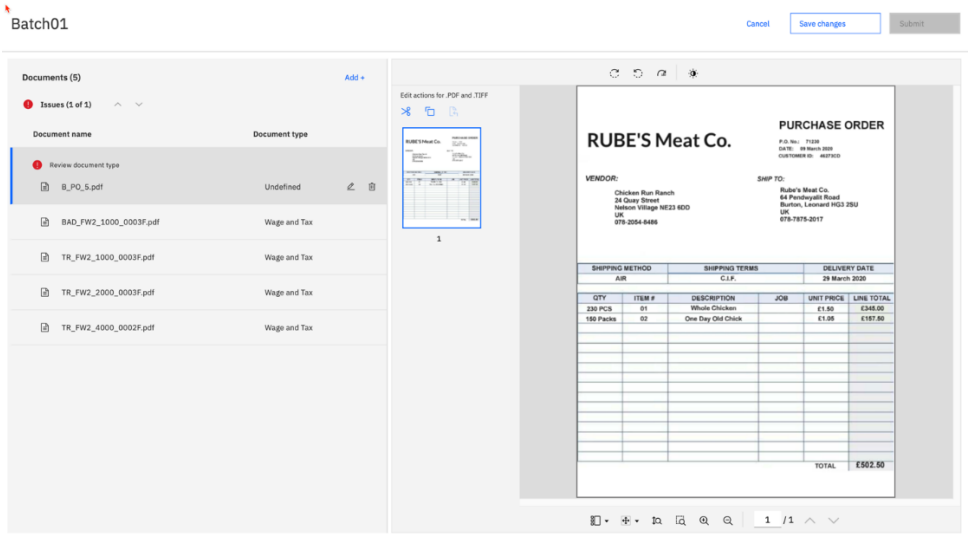
_1.



_2. **Click** on ***<your batch name>*** to open it

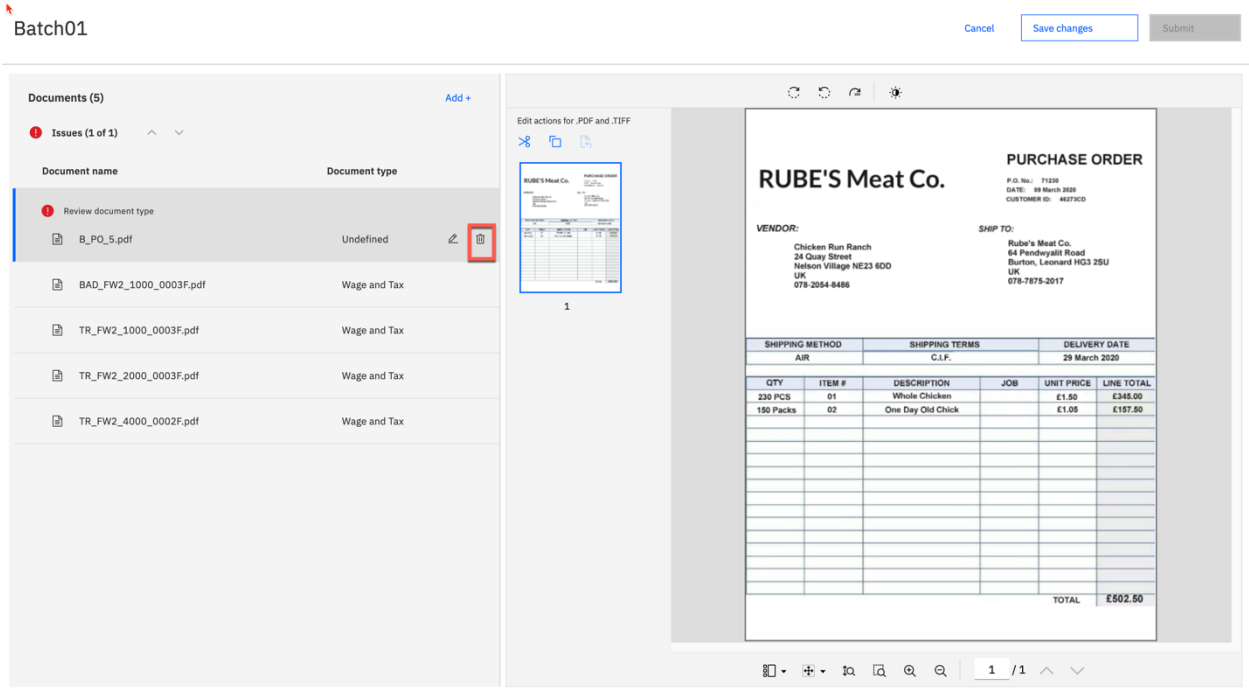
You should now see all the documents you uploaded in your batch. The ones with issues will have

- a **red checkmark** for documents that have a **low confidence** document type
- a **red exclamation** mark for documents that **could not be classified**



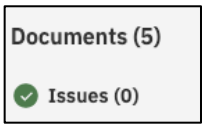
Most of the document types are correct but it looks like a Purchase order (PO) got mixed into our batch. **Click** on the **Trash can** to delete it from the batch and **select OK** to finally delete it.

_3.



_4. Review all documents to ensure everything is correct. If the system no longer detects any issues, you should see a green checkmark near the top of the document list.

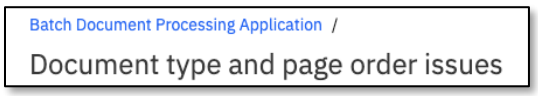
_5.



Click Save Changes and then **Submit** to save your changes and have the batch processed

_6. The system will start reprocessing the documents now that they have been classified correctly.

Click on the blue **Batch Document Processing Application** link at the top to return to the previous preview menu.



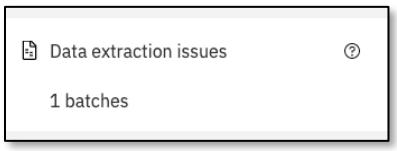
11.4 Correct extraction issues

The following instructions are based on a pre-trained sample application. Not what you will see in your untrained application.

> ***Important Note:*** *The project you are using for this has been configured but NOT run through the training (Deep Learning). So, the results will not reflect what they should be. IN A NORMAL SCENARIO, ON A CLUSTER WITH GPU AND DEEP LEARNING ENABLED, YOU WOULD HAVE TRAINED YOUR MODEL BEFORE DEPLOYING IT AND WOULD BENEFIT FROM HIGHER EXTRACTION RATES. The purpose of this lab is to teach you the tools but won't show you the trained results.*

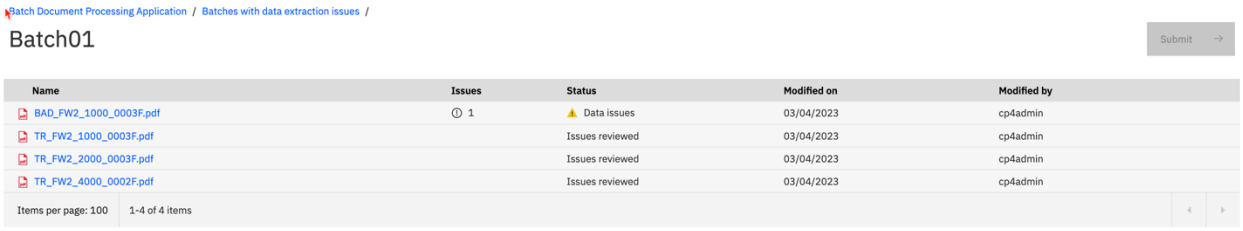
It may take a few seconds for your batch to advance to the next step. If your batch needs further attention, you will see it appear in the Data extraction issues tile.

_1. **Click** on the ***Data extraction issues*** tile to open it

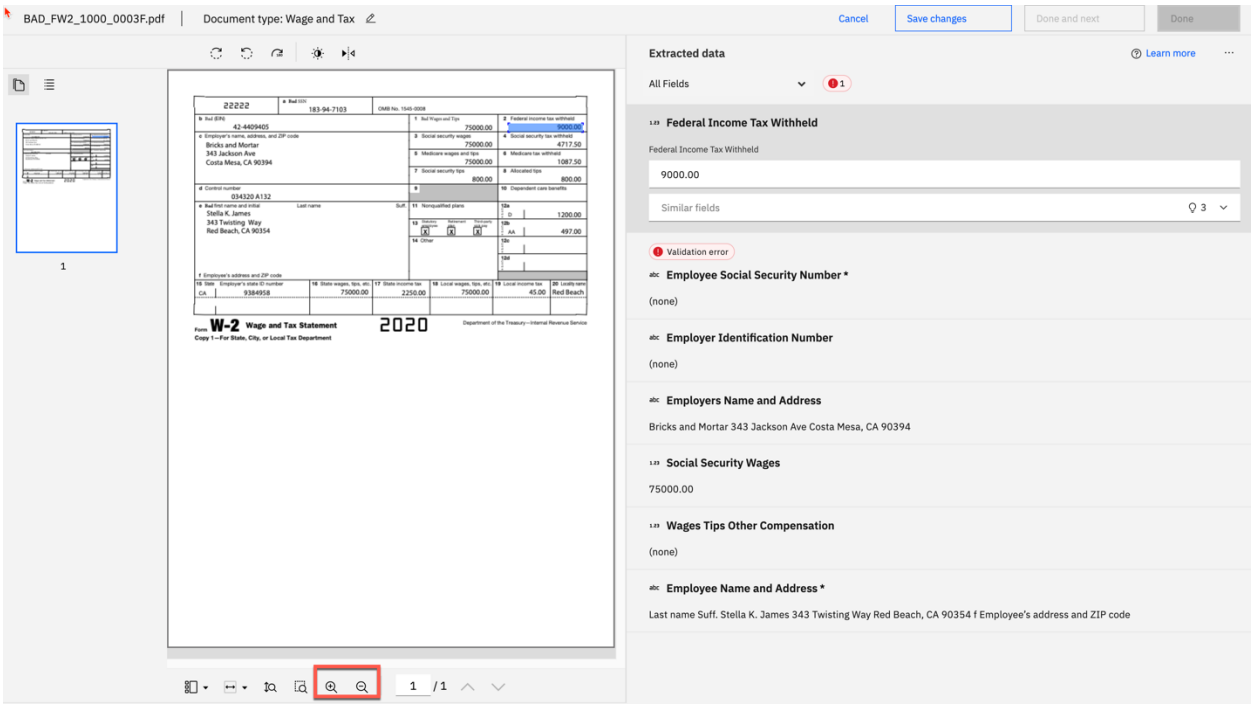


_2. **Click** on ***<your Batch name>*** to open

After opening we see all the documents that have been processed but one looks to have extraction issues.

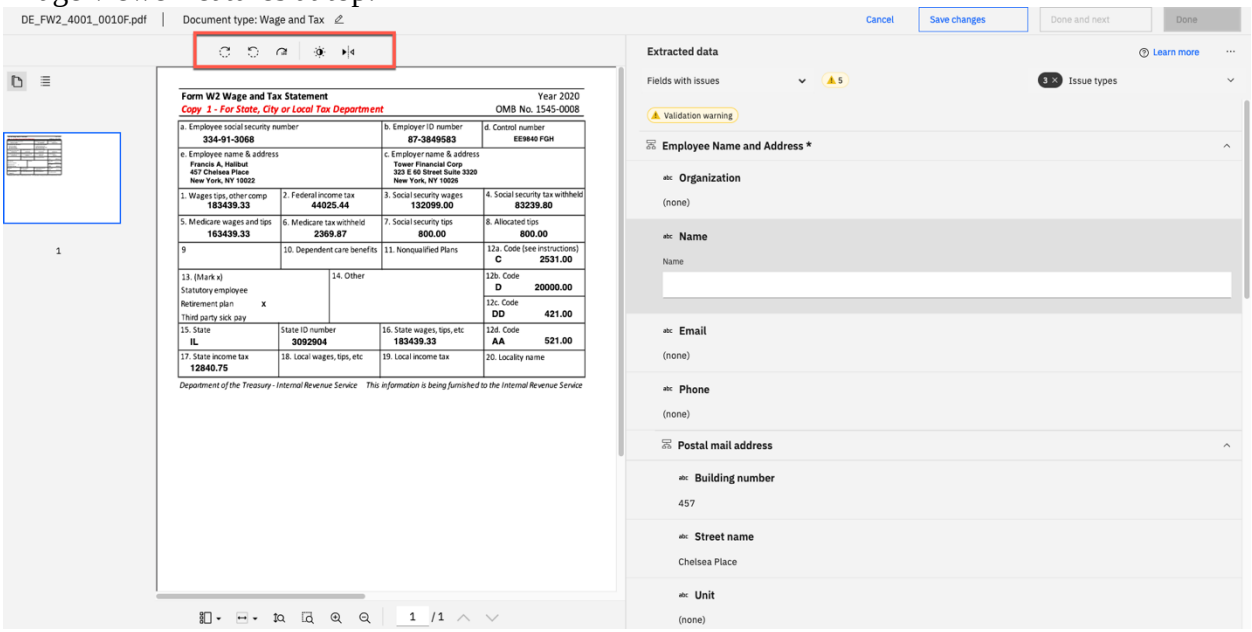


_3. **Click** on the bad document to open it. Zoom in a bit to get a better picture of the document.



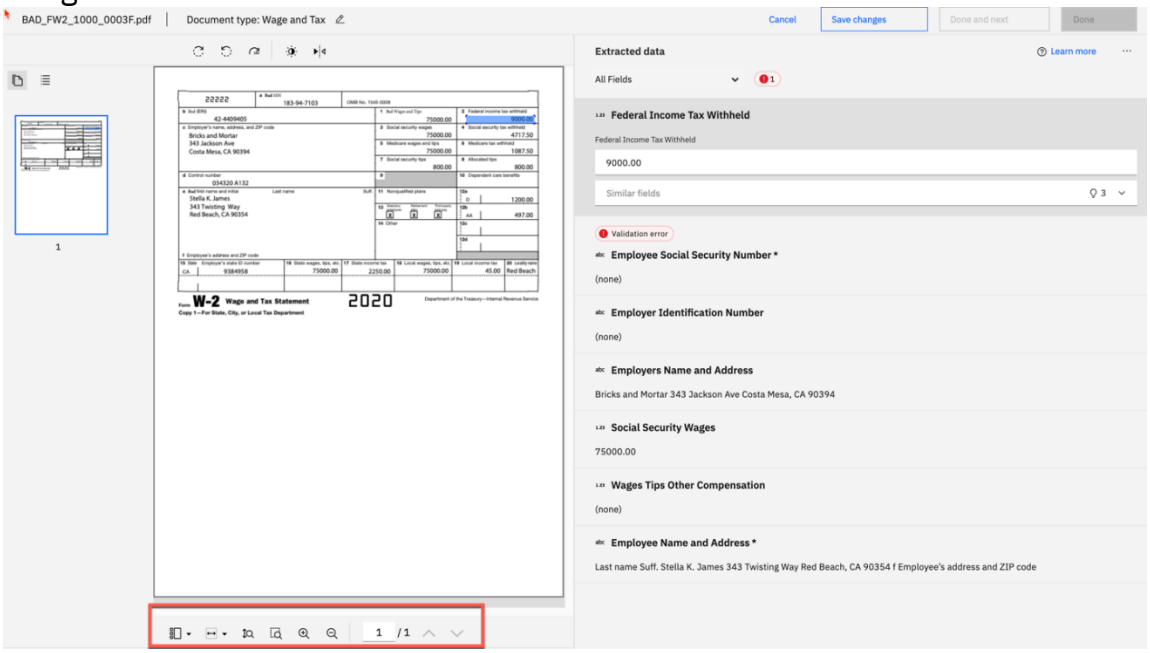
Take a moment to discover the image viewer features.

Image viewer features at top:



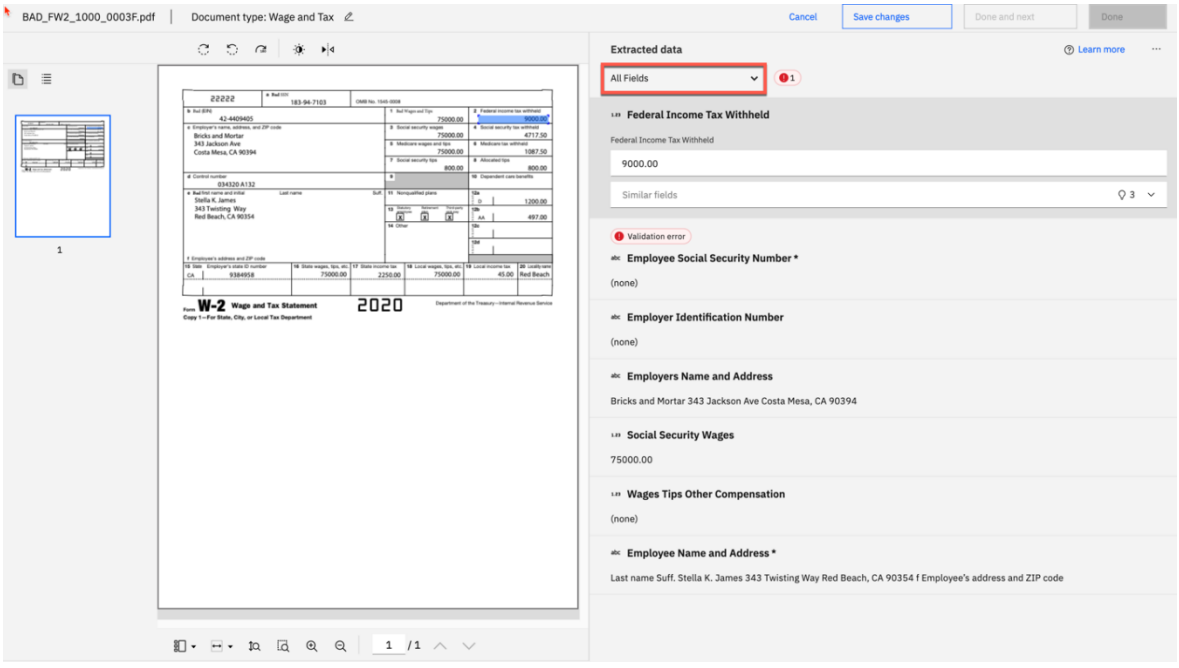
- Rotate image
- Visual effect adjustment
- Invert

Image viewer features at bottom:



- Page and thumbnail's view
- Fit to window
- Zoom and Magnify

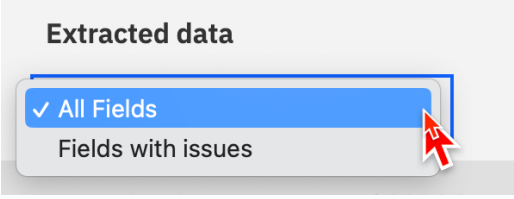
Field features



- Show all fields.
- Show fields with issues.

Also note that fields that that do have issues have a notification icon next to them. For example, Wages Tips Other Compensation field picked up correctly but has a low confidence based on the extraction results.

_4. Under Extracted data **click** on the **drop down twisty**



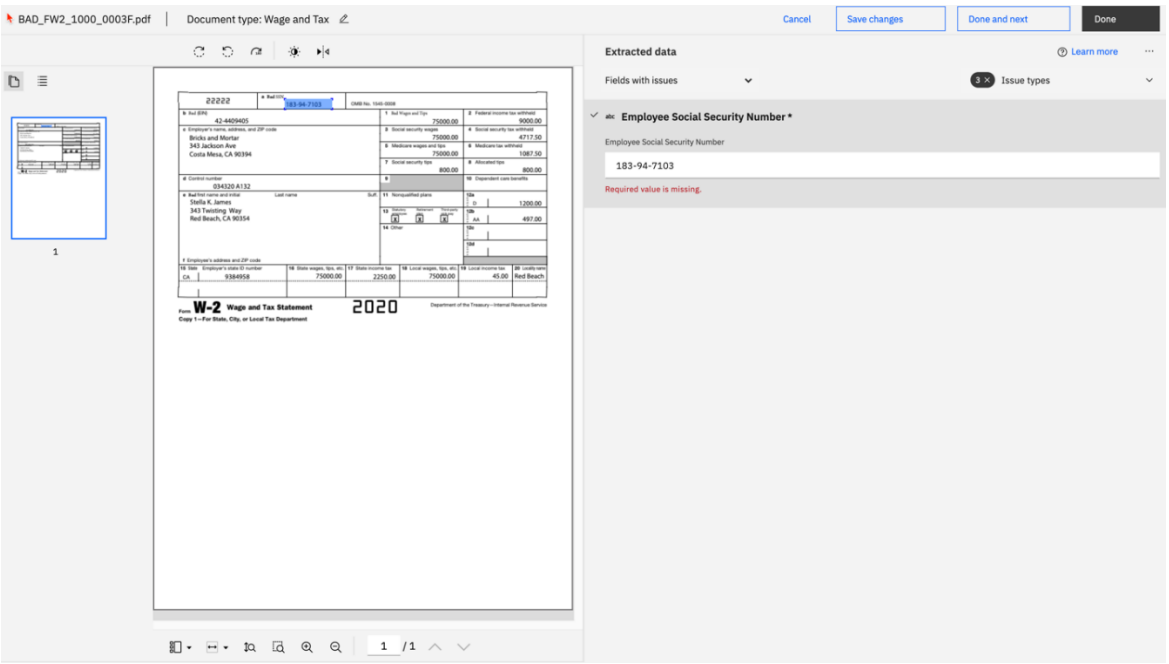
_5. **Click** on the **All Fields**

This view shows all the fields that we defined earlier. Fields with an asterisk are mandatory fields.

Change the Extracted data back to **Fields with issues**

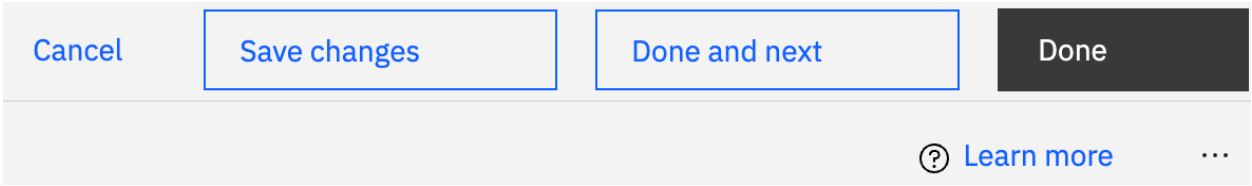
The Employee Social Security Number is a mandatory field. For purposes of this lab, it was changed to "Bad SSN". Since you did not make that phrase an alias ADP was not able to pick it up.

_6. **Click** on **Employee Social Security Number** and with your mouse **select** the **SSN** under **"Bad SSN"**



Also, the Wages Tips Other Compensation did not have a correct alias defined. But since it was not a required field, you can continue to process.

_7. **Click** on **Save Changes** box at the top



_8. For the remaining fields there are no extraction issues that ADP picked up for mandatory fields. You may see some low confidence characters. If so, **Click** on Dismiss for each field with a yellow validation warning.

_9. **Click** on **Done and next**

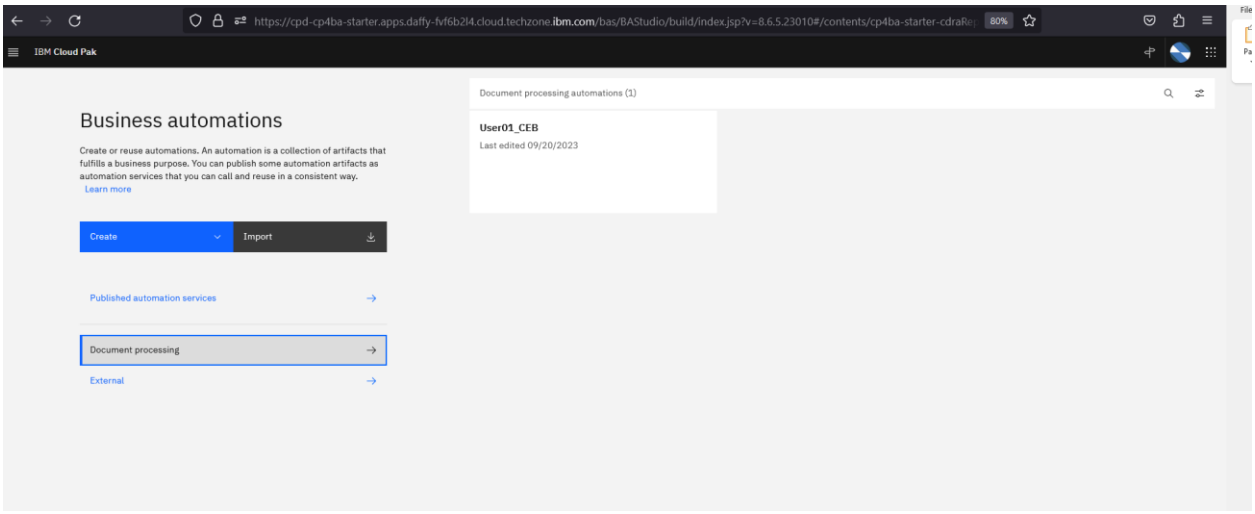
_10. All documents have been processed **Click** on **Submit** → at the top to complete the batch

12 Export/Import Project (Optional)

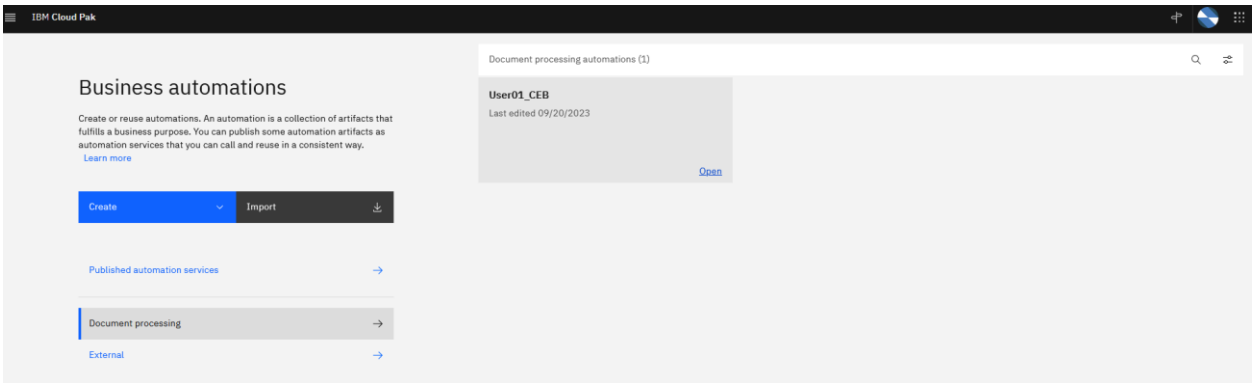
If you would like to save your project and perhaps use it later, you can perform the steps in this chapter.

From the Business Automations screen:

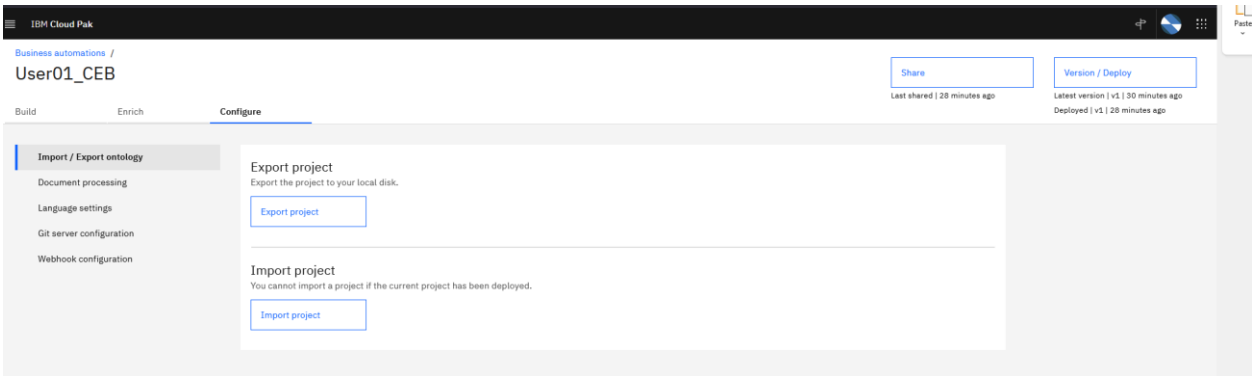
_1. **Select Document Processing**



_2. **Select <your project name>. Click open**

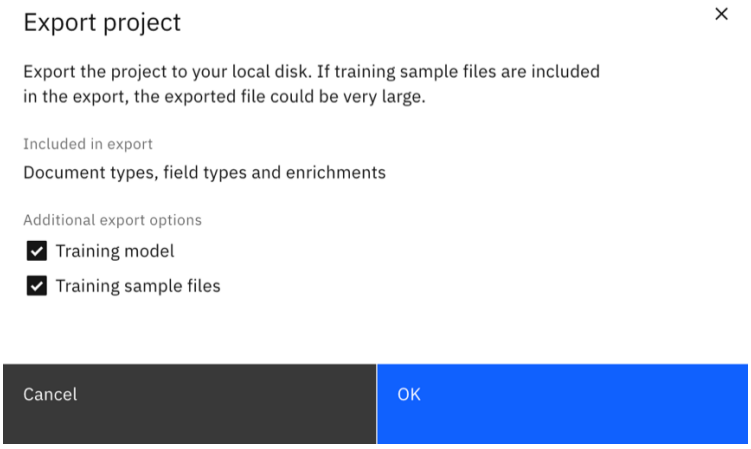


_3. From the main screen **select the Configure tab**



_4. **Select Export Project**

_5. In the Export Project window, **check Training model** and **Training sample files**



_6. **Click** on **OK**

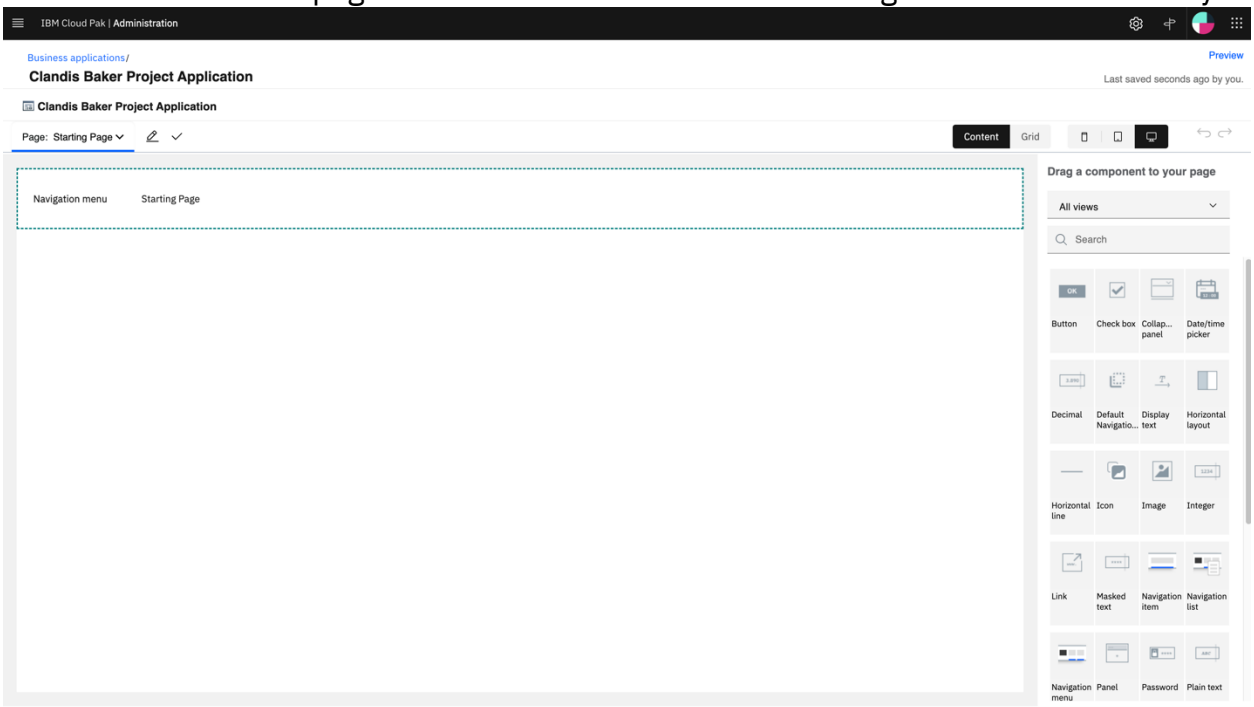
_7. A project-export-*<date-time>*.zip will be download via browser to local machine.

You have successfully completed the Automation Document Processing lab.
Congratulations and well done!

Appendix A - Troubleshooting

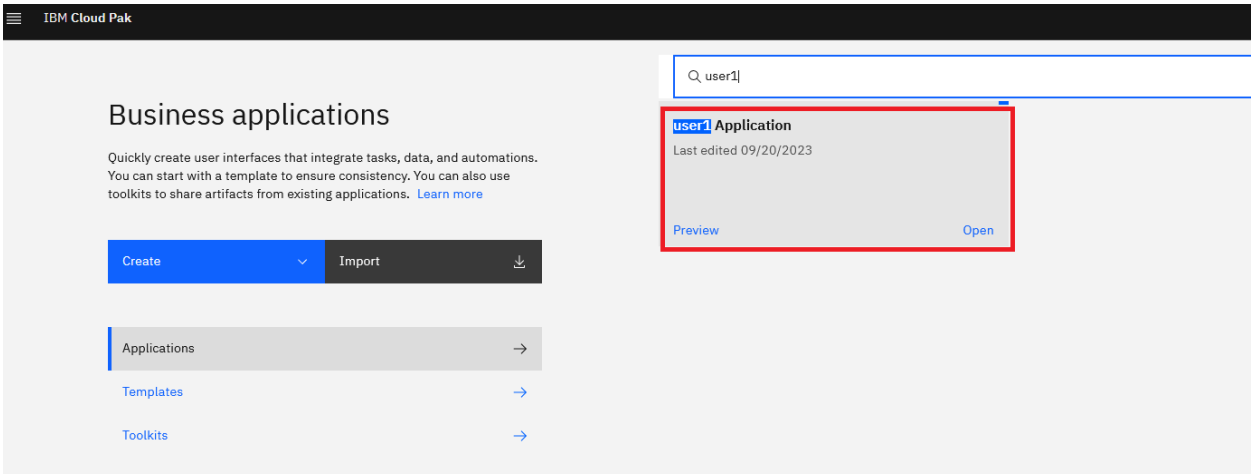
Blank Business Automation Application

After the creation of the Business Application, when you open the project for the first time after the Starter page remains blank or shows the loading animation indefinitely.

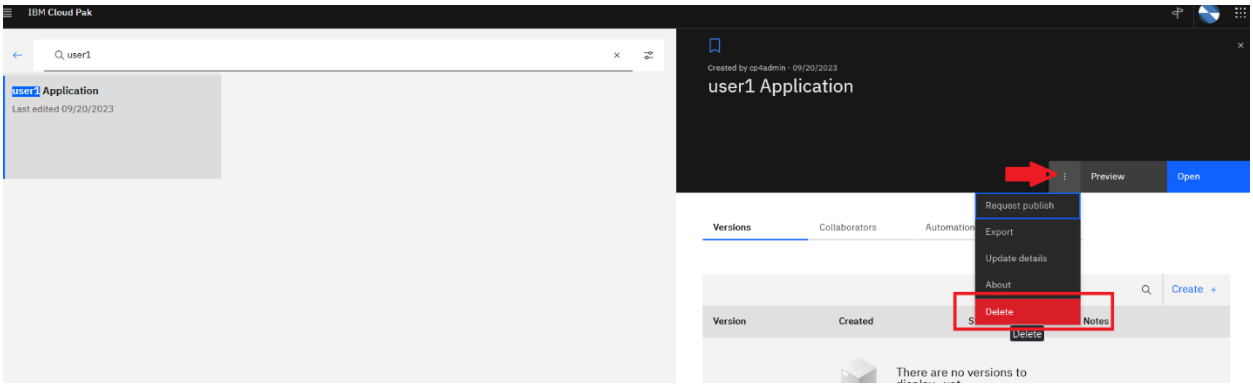


First try to reload the whole editor page and wait for the UI to be loaded.

If this remains unsuccessful, delete the application and try again. To delete the application, Click on the Application tile

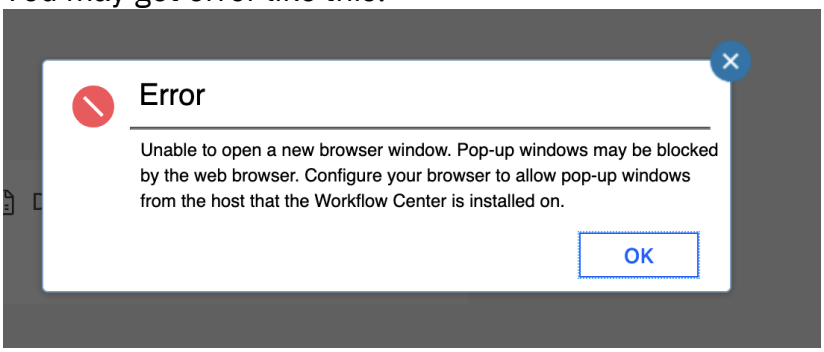


Then click on the 3 dots and select Delete.



Popup blocked when trying to Preview Application

You may get error like this:



You will need to grant access to pop up windows in your browser.

Appendix B - BAW & ADP Integration Sample

For the End-to-End demo, BAW was integrated with ADP. This link explains how to accomplish https://github.com/IBM/baw-adp-integration-sample.