Automate > Creating apps

Extracting data from documents

To process documents, you must specify which data points, or *fields*, you want to extract. If your project includes different document types, like a mix of passports and driver's licenses, create a *class* for each document type and specify fields for each class.

You can create up to 250 classes per project, and up to 100 fields per class.

The classes and fields in your project form the project *schema*: the blueprint of information you want to extract from documents.

Autogenerating a project schema Agent mode

Projects in agent mode can use a model to autogenerate classes and fields based on uploaded documents.

Autogenerated schemas create up to 20 fields using basic field types like text extraction and list extraction. This approach works well for straightforward extraction needs or as a starting point for more complex schemas that you can refine.

✓ **Before you begin**
You must have uploaded a set of files that represent the types of documents you want to process. Five or so files of each type is a good start.

1. In the editing panel, click **Autogenerate schema**.

   Commercial & Enterprise If your project includes multipage files, you're prompted to optionally enable splitting files.

   It might take several minutes for the schema to generate.

   You can use the schema as-is, modify it, or click the **Undo schema** icon ↩ to clear the autogenerated schema and manually generate your project schema.

Manually generating a project schema

To manually create your project schema, create classes for each document type in your project, if necessary, then create fields for the data points you want to extract. This approach gives you complete control over your schema.

Manual schema generation works across all processing modes and provides access to all field types.

✓ **Before you begin**
You must have uploaded a set of files that represent the types of documents you want to process. Five or so files of each type is a good start.

Creating classes

Organization members can import *prebuilt classes* from a library of established schemas, such as paystubs, invoices, bank statements, and utility bills.

In projects with classification, a default class called *other* is assigned to documents that can't be classified. You can't delete or modify this class.

1. In the editing panel, click the **Create classes** icon, then select one of these options based on your AI Hub subscription and project requirements:

   - **Create classes** — Lets you create a custom class without any fields. If you select this option, enter a succinct name for your document type, then click ← to exit the class editing panel.

   - **Browse prebuilt classes** Commercial & Enterprise — Lets you add common document types and their associated fields based on a library of available schemas. If you select this option, choose the prebuilt classes that you want to add and click **Add to project**.

2. Use the **Create classes** icon to add more classes as needed.

3. When you're done creating classes, click **Classify documents**.

   Commercial & Enterprise If your project includes multipage files, you're prompted to optionally enable splitting files. You can split by document—which lets the model determine where document breaks occur—or split each page—which creates a new document at every page break.

   Classes are assigned to your documents and documents are grouped by class in the document list. Any documents that can't be classified are assigned the *other* class.

4. Verify classification. If documents weren't classified as expected, edit classes to improve your results.

   1. In a class that wasn't identified accurately, click the overflow icon •••, then select **Edit class**.

   2. Enter a description to help the model more accurately identify documents in the class, then click ← to exit the class editing panel.

      > ☆ Effective descriptions include unique identifying details about a document class.
      >
      > In projects that don't use file splitting, you can reference file extensions to help classify documents. For example, the description for an *Images* class might be *Files with image file extensions, like JPEG, PNG, and TIF*.
      >
      > As a best practice, limit class descriptions to 1,000 characters (4,000 maximum).

   3. Use the overflow icon to edit more classes as needed.

   4. When you're done editing classes, click **Classify documents**.

› 🎓 **Visual tutorial: Creating classes**

Creating fields

Create fields for each of the data points you want to identify.

1. In the editing panel, click **Add field**.

3. Do one of the following, based on whether your result is accurate:
    - Accurate result — Click ← to exit the field editing panel and continue adding fields.
    - Inaccurate result — Edit the field. When you're done editing, click ← to exit the field editing panel and continue adding fields.

🎓 **Visual tutorial: Creating fields**

Editing fields

If a field doesn't return the results you expect, you have options to fine-tune the results.

Access the field editor for an existing field by hovering over the field and clicking the edit icon ✎.

In the field editor, first choose the field type appropriate for the data you want to identify.

With a suitable field type selected, if necessary, provide a more detailed description or prompt describing the information you're looking for. As a best practice, keep field and attribute names under 48 characters and use a description or prompt for longer content up to 1,000 characters (4,000 maximum). For best practices, see Writing effective prompts.

Legacy mode

For most field types, you can change the model using the model selector dropdown.

- Use the **standard model** for straightforward fields that perform basic text extraction or calculations. The standard model tends to perform best on shorter documents less than 50 pages. Its faster processing is suitable when speed is your priority.
- Use the **advanced model** for specialized fields that perform multistep reasoning or complex math. The advanced model performs better on longer documents and those with challenging formatting, and it's required for visual reasoning fields. Its more deliberate processing is suitable when accuracy is your priority.

For details about model capabilities, see Choosing a model.

When you're done editing a field, click **Run** to see results and further refine your edits if needed.

🎓 **Visual tutorial: Editing fields**

Field types

Choose the field type appropriate for the data you want to identify.

Field type	Commercial+	Used to...
Text extraction		Extract strings of text or numbers, such as address, account balance, or filing status.
Table extraction	✓	Extract structured tabular data from documents.
List extraction	✓	Extract multiple similar items with optional attributes, such as transactions or line items.

reasoning		deduction, summarization, or calculation.
Visual reasoning	✓	Analyze visual and stylistic elements including images, watermarks, layout, and formatting. Requires the advanced model.
Derived	✓	Generate values based on other fields in the class.

For more guidance, see Choosing field types.

Custom function fields Commercial & Enterprise

The custom function field type lets you use a Python function to compute values or import data to your project schema.

For example, you might use a custom function to calculate total invoice amount using existing subtotal and tax rate fields:

```python
subtotal = float(subtotal)
tax_rate = float(tax_rate) / 100
tax_amount = subtotal * tax_rate
total_amount = subtotal + tax_amount

return round(total_amount, 2)
```

Custom function fields accept these parameters:

Parameter	Required?	Description
`context`	Required	Stores metadata about the document.
`context['document_text']`	Optional	Retrieves the entire text of the document.
`context['file_path']`	Optional	Retrieves the path to the uploaded file.
`keys`	Optional	Access custom variables and organization secrets. Use `keys['custom']['<key-name>']` for custom keys and `keys['secret']['<key-name>']` for secret keys.
`<additional-field-name>`	Optional	When writing custom functions in automation projects, click **Add argument** to select additional fields in the class to use in the function.

For additional guidance about custom functions, see Writing custom functions.

Viewing results across documents

To quickly scan or compare results, click the **Results table** icon ⊞ in the **Documents** header.

The results table corresponds to the current view in the editing panel, so the results you see change depending on your current task.

If the editing panel shows...	Then the results table displays...
Classes	Final results for all fields, across all classes.
Field editor	Final result and, if applicable, confidence threshold validation result for the selected field.

Validations with no rule selected	Validation results for all fields, across all classes.
Validations with a rule	

Reordering fields

To change the order of fields in the field editor, use the up and down arrows that display when you hover over a field.

Reordering fields can be necessary when creating derived fields, which can reference fields that precede it in the field editor. Additionally, reordering fields can be helpful to speed up reviews or support downstream integrations, because fields are displayed in processed results in the same order as in the field editor.

> If you have derived fields or custom functions in your project that reference preceding fields, be aware that reordering fields can break the reference.

Hiding fields

Commercial & Enterprise

Hiding intermediate or computational fields can help simplify human review and downstream integration output.

Consider hiding fields that are used exclusively as input for derived fields or custom functions. For example, you might extract individual date components in separate hidden fields, then combine them into a final formatted date field that reviewers and downstream systems actually need.

To mark a field as hidden, open the field editor and enable **Hide field**.

Hidden fields can't have validation rules, because validations on hidden fields could create confusing review scenarios. If you hide a field with an active validation rule, the rule is removed. If you later unhide the same field, any previous validation rules are restored.

Hidden fields use processing resources and count toward field limits, but their visibility varies across different AI Hub interfaces:

Interface	Hidden field behavior
App run results (UI)	Hidden by default, can be unhidden with human review field filters
App run results (exported)	Unhidden
Accuracy tests	Hidden by default, can be unhidden via test configuration
Human review	Hidden by default, can be unhidden with human review field filters
Deployment run results (exported)	Unhidden
Downstream integrations	Hidden by default, can be unhidden via deployment configuration
API & SDK results	Hidden by default, can be unhidden via deployment configuration

Was this page helpful? Yes No