Prediction

Prediction

✎

Prediction Endpoint

Prediction is the main endpoint of Mindee API to extract information from your document. Use the Prediction endpoint by selecting the API you want to use and upload your document. You will then receive JSON predictions at document level or even page level for the available fields.

URL

To make a prediction, select the document API `<account_name>/<api_name>/<api_version>` where:

- `<account_name>` refers to the username or organization name of the account that created the API,
- `<api_name>/<api_version>` refers to the name and selected version as described in [API Documentation](#).

Then use the URL:

HTTP

```
POST https://api.mindee.net/v1/products/<account_name>/<api_name>/<api_version>/pred
```

Off-the-shelf APIs

Mindee ready-to-use APIs are accessible on the account name `mindee`. You can browse all of them in the [API Store](#).

> 📘 Off-the-shelf APIs use a major version convention. A new major version may not be fully backward compatible and bring new features and better performance.
>
> Examples:
>
> - <u>Invoice</u>: `mindee/invoices/v3`
> - <u>Passport</u>: `mindee/passport/v1`

- **Expense Receipt**: `mindee/expense_receipts/v3`

Custom APIs

When creating a custom [document parsing API](#) with the API Builder, you must train the API before making your first predictions. As the training is progressing, a new minor version is created for each new model deployed:

- `v1.0` - no model / no predictions
- `v1.1` - first model
- `v1.2` - second model
- ... etc

📘 Select the version `v1` to always have the latest and best model.

Example: `bob/form_456/v1`

Payload

The Prediction endpoint can handle three types of payload in order to send your document:

- a binary file
- a base64 encoded file
- a URL

See [Document inputs](#) for more information on supported files.

Send a Binary File

Use a `multipart/form-data` encoding to send your document

Example:

cURL   Python   C# (.NET)

```
curl -X POST
  https://api.mindee.net/v1/products/<account_name>/<api_name>/<api_version>/predict
  -H 'Authorization: Token my-token'
  -F document=@/path/to/your/file.png
```

🚧 The `@` in the `curl` command is very important as it tells curl that you aren't passing a data but a file.

Send a Base64 Encoded File

Prepare a JSON payload:

JSON

```json
{
  "document": "/9j......"
}
```

Send your request with an `application/json` encoding:

Shell

```shell
curl -X POST \
  https://api.mindee.net/v1/products/<account_name>/<api_name>/<api_version>/predict
  -H 'Authorization: Token my-api-key-here' \
  -H 'Content-Type: application/json' \
  -d 'document="/9j..."'
```

Send a URL

Prepare a JSON payload:

JSON

```json
{
  "document": "https://mydomain.com/my_file.pdf"
}
```

Send your request with an `application/json` encoding:

Shell

```shell
curl -X POST \
  https://api.mindee.net/v1/products/<account_name>/<api_name>/<api_version>/predict
  -H 'Authorization: Token my-token' \
  -H 'Content-Type: application/json' \
  -d '{"document":"https://mydomain.com/my_file.pdf"}'
```

> **!** Only a public HTTPS URL is accepted.

JSON Response

See *Endpoints* for general description of Mindee's REST API response format.

Description

When calling the prediction endpoint, the parsed information from your documents can be found in the `document` key.

JSON

```json
{
  "api_request": { .. },
  "document": {
    "id": "ac668055-e7db-48f2-b81f-e5ba9a6a6b8f",
    "name": "myfile.pdf",
    "n_pages": 2,
    "inference": {
      "started_at": "2021-03-24T09:14:27+00:00",
      "finished_at": "2021-03-24T09:14:28+00:00",
      "processing_time": 1.087,
      "is_rotation_applied": true,
      "extras": {},
      "prediction": { .. },
      "pages": [
        {
          "id": 0,
          "orientation": {"value": 0},
          "extras": {},
          "prediction": { .. }
        },
        {
          "id": 1,
          "orientation": {"value": 0},
          "extras": {},
          "prediction": { .. }
        }
      ]
    }
  }
}
```

Document

Describes the uploaded document

key	type	description
id	*string*	a unique identifier
name	*string*	the filename
n_pages	*number*	the number of pages
inference	*object*	a JSON object with the content of your inference (prediction)

Document > Inference

Contains the whole inference data (predictions)

key	type	description
started_at	*string*	the date & time the inference has started in ISO 8601 format
finished_at	*string*	the date & time the inference was finished in ISO 8601 format
processing_time	*number*	the request processing time in seconds
is_rotation_applied	*boolean* or *null*	**true**: polygons are *already* rotated given the page orientation **false**: polygons are *never* rotated **null**: the API has no orientation information
extras	*object*	a JSON object with document-level extras predictions
prediction	*object*	a JSON object with the document-level API prediction
pages	*list[object]*	a JSON object with the page-level inference data

Document > Inference > Pages[ ]

Contains the page-level specific inference data (predictions)

key	type	description
id	*number*	the page index
orientation.value	*number*	the clockwise rotation to apply to get the page upright Examples: **0**, **90**, **180**, **270**
extras	*object*	a JSON object with page-level extras predictions *Example: the [Cropper](#) feature*
prediction	*object*	a JSON object with the page-level API prediction

Prediction example

Each API can describe several fields within its `prediction` object. Depending on the field properties, you will find values, a confidence score or polygons.

```JSON
{
  "prediction": {
    "locale": {
      "country": "CA",
      "currency": "CAD",
```

```
      "language": "en",
      "value": "en-CA",
      "confidence": 0.85
    },
    "date": {
      "value": "2020-07-03",
      "confidence": 0.99,
      "polygon": [[0.273, 0.355], [0.289, 0.355], [0.289, 0.373], [0.273, 0.373]]
    },
    "total_incl": {
      "value": 14.32,
      "confidence": 0.98,
      "polygon": [[0.581, 0.485], [0.696, 0.485], [0.696, 0.503], [0.581, 0.503]]
    }
  }
}
```

👍 **Success**

To know more about your document parsing API response, especially the prediction object's structure, you can access the Documentation part of your API on Mindee's platform.

Questions?

🟦 Join our Slack

🕑 Updated 9 months ago

Did this page help you?   👍 Yes   👎 No