UTILITIES

Ask

Cropper

What is Cropper?

Cropper enables users to retrieve the coordinates of cropped images from a document through our APIs.

Cropper can be used in two ways:

- On any API by adding a parameter.
- By using the stand-alone API Cropper.

How it Works

The cropper feature can be used on any API on the prediction route:

```
https://api.mindee.net/v1/products/<account_name>/<api_name>/<api_
version>/predict?cropper=true
```

`?cropper=true` is the additional parameter to add to your API calls.

Cropper Output

The cropper results are located at the page-level when retrieving document prediction. The cropper results are returned in the following JSON object `document.inference.pages[].extras.cropper.cropping[].xxxxxx`. Most of the time, taking the first element of the cropper output is sufficient for most use case.

The results show the polygon vertices at the page level, for each possible cropped document within the image:

- **bounding_box**: straight rectangle (always inside canvas).
- **rectangle**: rectangle that may be oriented (can go beyond the canvas, i.e. vertices coordinate < 0 and > 1).
- **quadrangle**: free polygon with 4 vertices (always inside canvas).
- **polygon**: free polygon with up to 30 vertices (always inside canvas).

The vertices format are a list of (x, y) relative coordinates to your document, always in clockwise order. Starting vertex may not be fixed (depends if this is a rectangle or a free polygon).

Example:

```
...
"rectangle": [
    [0, 0.996],
    [0.002, -0.002],
    [0.996, 0],
    [0.994, 0.998]
]
...
```

The coordinates returned by the cropper take in consideration the rotation of the document. You can find this information in `document.inference.pages[].orientation` in the form of clockwise degrees. The value is the clockwise rotation to apply on the source page document to get the page upright.

Example With Receipts V5

Using this sample receipt document↗ we append the cropper additional param `?cropper=true` to the receipts API URL.

Python | Node.js | PHP | Ruby | Bash

```python
from mindee import Client, PredictResponse, product

# Init a new client
mindee_client = Client(api_key="my-api-key-here")

# Load a file from disk
input_doc =
mindee_client.source_from_path("/path/to/the/file.ext")

# Load a file from disk and parse it.
result: PredictResponse = mindee_client.parse(
    product.ReceiptV5,
    input_doc,
    # Add the cropper param to the API call
    cropper=True
)

# Print a summary of the API result
print(result.document)

# Print the document-level summary
# print(result.document.inference.prediction)
```

Our result shows the different polygon vertices (`bounding_box`, `rectangle`, `quadrangle`, `polygon`) at the page level.

```
{
    ...
    "document": {
        "id": "62c6198c-e0a8-4ee7-b922-7b5b98bedb79",
        "inference": {
            "extras": {},
            "finished_at": "2022-03-25T16:17:18+00:00",
            "is_rotation_applied": true,
```

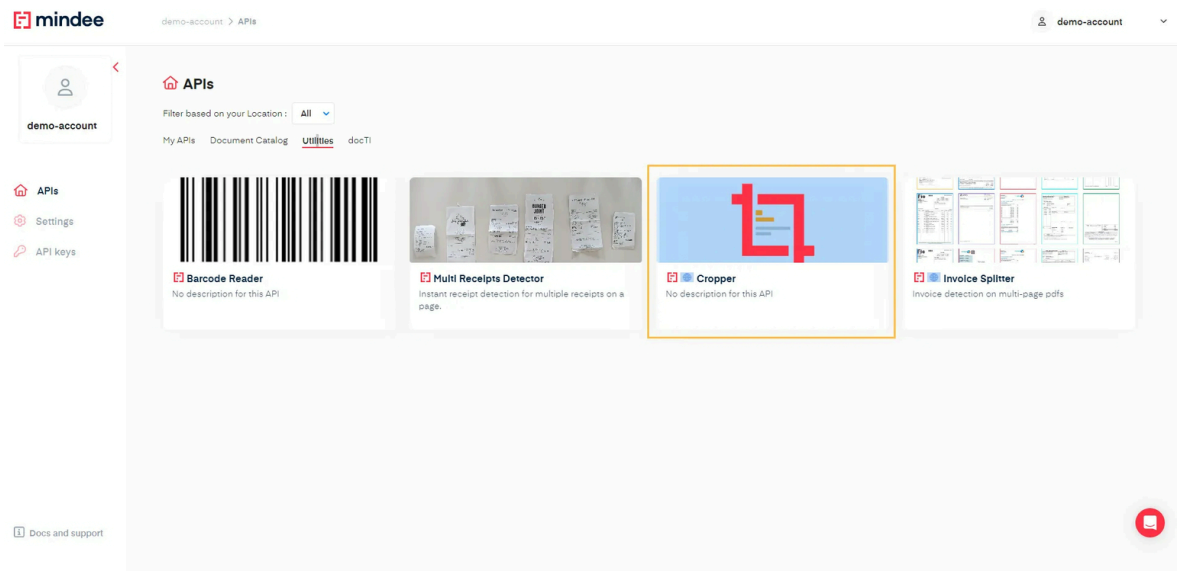

- Replace **my-api-key-here** with your new API key, or use the **select an API key** feature and it will be filled automatically.
- Copy and paste the sample code of your desired choice in your application, code environment, terminal etc.
- Replace `/path/to/the/file.ext` with the path to your input document.

> Remember to replace with your V1 API key.

Cropper standalone API

Setup the API

1. Access your Cropper API by clicking on the **Cropper** card in the Utilities tab:



2. From the left navigation, go to documentation **> API Reference**, you'll find sample code in popular languages and command line.

Python | Node.js | PHP | .NET | Ruby | Java | Bash

```
from mindee import Client, PredictResponse, product

# Init a new client
mindee_client = Client(api_key="my-api-key-here")

# Load a file from disk
input_doc =
mindee_client.source_from_path("/path/to/the/file.ext")

# Load a file from disk and parse it.
# The endpoint name must be specified since it cannot be
determined from the class.
result: PredictResponse =
mindee_client.parse(product.CropperV1, input_doc)

# Print a summary of the API result
print(result.document)

# Print the document-level summary
# print(result.document.inference.prediction)
```

- Replace **my-api-key-here** with your new API key, or use the **select an API key** feature and it will be filled automatically.
- Copy and paste the sample code of your desired choice in your application, code environment, terminal etc.
- Replace `/path/to/the/file.ext` with the path to your input document.

Remember to replace with your V1 API key.

API Response

Here is the full JSON response you get when you call the API:

```
{
  "extras": {},
  "finished_at": "2023-04-18T14:15:13.557525",
  "is_rotation_applied": null,
  "pages": [
    {
      "extras": {},
      "id": 0,
      "orientation": {
        "value": null
      },
      "prediction": {
        "cropping": [
          {
            "bounding_box": [
              [
                0.547,
                0.096
              ],
              [
                0.883,
                0.096
              ],
              [
                0.883,
                0.977
              ],
              [
                0.547,
                0.977
              ]
            ],
            "polygon": [
              [
                0.605,
                0.332
              ],
              [
                0.613,
                0.188
              ],
              [
                0.645,
                0.15
              ],
              [
                0.672,
                0.168
```

```
        ],
        [
          0.766,
          0.166
        ],
        [
          0.791,
          0.152
        ],
        [
          0.869,
          0.158
        ],
        [
          0.871,
          0.234
        ],
        [
          0.859,
          0.277
        ],
        [
          0.859,
          0.402
        ],
        [
          0.836,
          0.658
        ],
        [
          0.562,
          0.885
        ],
        [
          0.568,
          0.654
        ],
        [
          0.594,
          0.514
        ],
        [
          0.596,
          0.445
        ],
        [
          0.609,
          0.406
```

```
        ]
      ],
      "quadrangle": [
        [
          0.625,
          0.096
        ],
        [
          0.882,
          0.157
        ],
        [
          0.803,
          0.975
        ],
        [
          0.549,
          0.9
        ]
      ],
      "rectangle": [
        [
          0.625,
          0.096
        ],
        [
          0.886,
          0.121
        ],
        [
          0.803,
          0.975
        ],
        [
          0.542,
          0.949
        ]
      ]
    },
    {
      "bounding_box": [
        [
          0.195,
          0
        ],
        [
          0.482,
          0
```

```
        ],
        [
          0.482,
          0.988
        ],
        [
          0.195,
          0.988
        ]
      ],
      "polygon": [
        [
          0.201,
          0.439
        ],
        [
          0.191,
          0.281
        ],
        [
          0.197,
          0.291
        ],
        [
          0.191,
          0.117
        ],
        [
          0.197,
          0.041
        ],
        [
          0.254,
          0.037
        ],
        [
          0.33,
          0.006
        ],
        [
          0.436,
          0.006
        ],
        [
          0.467,
          0.301
        ],
        [
```

```
            0.213,
            0.986
          ],
          [
            0.215,
            0.797
          ],
          [
            0.186,
            0.793
          ],
          [
            0.197,
            0.744
          ],
          [
            0.193,
            0.689
          ],
          [
            0.211,
            0.629
          ],
          [
            0.197,
            0.576
          ]
        ],
        "quadrangle": [
          [
            0.197,
            0.019
          ]
```


Previous
Nutrition facts Label

Next
Barcode Reader OCR

Last updated 2 months ago

Was this helpful?