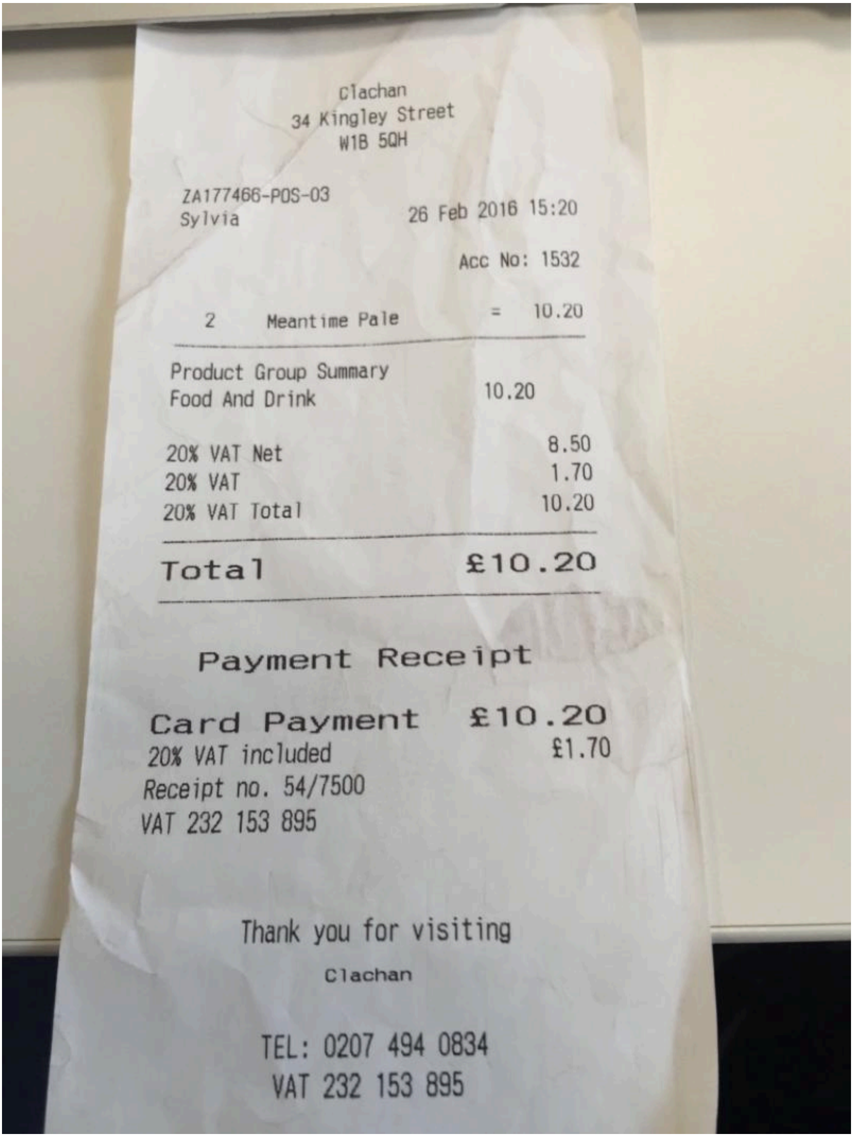
Documentation

Receipt OCR Node.js

Receipt OCR Node.js

The Node.js OCR SDK supports the Receipt API.

Using the sample below, we are going to illustrate how to extract the data that we want using the OCR SDK.



Clachan
34 Kingley Street
W1B 5QH

ZA177466-POS-03
Sylvia
26 Feb 2016 15:20

Acc No: 1532

2	Meantime Pale	=	10.20

Product Group Summary	
Food And Drink	10.20
20% VAT Net	8.50
20% VAT	1.70
20% VAT Total	10.20
Total	£10.20

Payment Receipt

Card Payment	£10.20
20% VAT included	£1.70

Receipt no. 54/7500
VAT 232 153 895

Thank you for visiting
Clachan

TEL: 0207 494 0834
VAT 232 153 895

Quick-Start

JavaScript

```
const mindee = require("mindee");
// for TS or modules:
// import * as mindee from "mindee";

// Init a new client
const mindeeClient = new mindee.Client({ apiKey: "m

// Load a file from disk
const inputSource = mindeeClient.docFromPath("/path

// Parse the file
const apiResponse = mindeeClient.parse(
  mindee.product.ReceiptV5,
  inputSource
);

// Handle the response Promise
apiResponse.then((resp) => {
  // print a string summary
  console.log(resp.document.toString());
});
```

Output (RST):

rst

```
########
Document
########
:Mindee ID: ce41e37a-65d8-4de1-b34b-1c92ab04b1ae
:Filename: default_sample.jpg

Inference
```

```
#########
:Product: mindee/expense_receipts v5.0
:Rotation applied: Yes

Prediction
==========
:Expense Locale: en-GB; en; GB; GBP;
:Purchase Category: food
:Purchase Subcategory: restaurant
:Document Type: EXPENSE RECEIPT
:Purchase Date: 2016-02-26
:Purchase Time: 15:20
:Total Amount: 10.20
:Total Net: 8.50
:Total Tax: 1.70
:Tip and Gratuity:
:Taxes:
  +---------------+--------+----------+---------
  | Base          | Code   | Rate (%) | Amount
  +===============+========+==========+=========
  | 8.50          | VAT    | 20.00    | 1.70
  +---------------+--------+----------+---------
:Supplier Name: CLACHAN
:Supplier Company Registrations: 232153895
```

Field Types

Standard Fields

These fields are generic and used in several products.

Basic Field

Each prediction object contains a set of fields that inherit from the generic `Field` class.
A typical `Field` object will have the following attributes:

- **value** (`number | string`): corresponds to the field value. Can be `undefined` if no value was extracted.
- **confidence** (`number`): the confidence score of the field prediction.
- **boundingBox** (`[Point, Point, Point, Point]`): contains exactly 4 relative vertices (points) coordinates of a right rectangle containing the field in the document.
- **polygon** (`Point[]`): contains the relative vertices coordinates (`Point`) of a polygon containing the field in the image.
- **pageId** (`number`): the ID of the page, is `undefined` when at document-level.
- **reconstructed** (`boolean`): indicates whether an object was reconstructed (not extracted as the API gave it).

> **Note:** A `Point` simply refers to an array of two numbers (`[number, number]`).

Aside from the previous attributes, all basic fields have access to a `toString()` method that can be used to print their value as a string.

Amount Field

The amount field `AmountField` only has one constraint: its **value** is a `number` (or `undefined`).

Classification Field

The classification field `ClassificationField` does not implement all the basic `Field` attributes. It only implements **value**, **confidence** and **pageId**.

> Note: a classification field's `value is always a` string`.

Company Registration Field

Aside from the basic `Field` attributes, the company registration field `CompanyRegistrationField` also implements the following:

- **type** (`string`): the type of company.

Date Field

Aside from the basic `Field` attributes, the date field `DateField` also implements the following:

- **dateObject** (`Date`): an accessible representation of the value as a JavaScript object.

Locale Field

The locale field `LocaleField` only implements the **value**, **confidence** and **pageId** base `Field` attributes, but it comes with its own:

- **language** (`string`): ISO 639-1 language code (e.g.: `en` for English). Can be `undefined` .
- **country** (`string`): ISO 3166-1 alpha-2 or ISO 3166-1 alpha-3 code for countries (e.g.: `GRB` or `GB` for "Great Britain"). Can be `undefined` .

- **currency** (`string`): ISO 4217 code for currencies (e.g.: `USD` for "US Dollars"). Can be `undefined` .

String Field

The text field `StringField` only has one constraint: its **value** is a `string` (or `undefined`).

Taxes Field

Tax

Aside from the basic `Field` attributes, the tax field `TaxField` also implements the following:

- **rate** (`number`): the tax rate applied to an item can be expressed as a percentage. Can be `undefined` .
- **code** (`string`): tax code (or equivalent, depending on the origin of the document). Can be `undefined` .
- **base** (`number`): base amount used for the tax. Can be `undefined` .

> Note: currently `TaxField` is not used on its own, and is accessed through a parent `Taxes` object, an array-like structure.

Taxes (Array)

The `Taxes` field represents an array-like collection of `TaxField` objects. As it is the representation of several objects, it has access to a custom `toString()` method that can render a `TaxField` object as a table line.

Specific Fields

Fields which are specific to this product; they are not used in any other product.

Line Items Field

List of line item details.

A `ReceiptV5LineItem` implements the following attributes:

- `description` (string): The item description.
- `quantity` (number): The item quantity.
- `totalAmount` (number): The item total amount.
- `unitPrice` (number): The item unit price.

Attributes

The following fields are extracted for Receipt V5:

Purchase Category

category (ClassificationField): The purchase category among predefined classes.

JavaScript

```
console.log(result.document.inference.prediction.ca
```

Purchase Date

date (DateField): The date the purchase was made.

JavaScript

```
console.log(result.document.inference.prediction.da
```

Document Type

documentType (ClassificationField): One of: 'CREDIT CARD RECEIPT', 'EXPENSE RECEIPT'.

JavaScript

```
console.log(result.document.inference.prediction.do
```

Line Items

lineItems (ReceiptV5LineItem[]): List of line item details.

JavaScript

```
for (const lineItemsElem of result.document.inferen
  console.log(lineItemsElem.value);
}
```

Expense Locale

locale (LocaleField): The locale detected on the document.

JavaScript

```
console.log(result.document.inference.prediction.lo
```

Purchase Subcategory

subcategory (ClassificationField): The purchase subcategory among predefined classes for transport and food.

JavaScript

```
console.log(result.document.inference.prediction.su
```

Supplier Address

supplierAddress (StringField): The address of the supplier or merchant.

JavaScript

```
console.log(result.document.inference.prediction.su
```

Supplier Company Registrations

supplierCompanyRegistrations (CompanyRegistrationField[]): List of company registrations associated to the supplier.

JavaScript

```
for (const supplierCompanyRegistrationsElem of resu
  console.log(supplierCompanyRegistrationsElem.valu
}
```

Supplier Name

supplierName (StringField): The name of the supplier or merchant.

JavaScript

```
console.log(result.document.inference.prediction.su
```

Supplier Phone Number

supplierPhoneNumber (StringField): The phone number of the supplier or merchant.

JavaScript

```
console.log(result.document.inference.prediction.su
```

Taxes

taxes (TaxField[]): List of tax lines information.

JavaScript

```javascript
for (const taxesElem of result.document.inference.p
  console.log(taxesElem.value);
}
```

Purchase Time

time (StringField): The time the purchase was made.

JavaScript

```javascript
console.log(result.document.inference.prediction.t
```

Tip and Gratuity

tip (AmountField): The total amount of tip and gratuity.

JavaScript

```javascript
console.log(result.document.inference.prediction.t
```

Total Amount

totalAmount (AmountField): The total amount paid: includes taxes, discounts, fees, tips, and gratuity.

JavaScript

```
console.log(result.document.inference.prediction.to
```

Total Net

totalNet (AmountField): The net amount paid: does not include taxes, fees, and discounts.

JavaScript

```
console.log(result.document.inference.prediction.to
```

Total Tax

totalTax (AmountField): The total amount of taxes.

JavaScript

```
console.log(result.document.inference.prediction.to
```

Questions?

Join our Slack

Updated 5 months ago

← Invoice OCR Node.js

Financial Document OCR Node.js →

Did this page help you? 👍 Yes 👎 No