DocuScrape — Reading a Refugee ID Card QR Code

A step-by-step guide. No programming knowledge is needed to follow it; developers can jump straight to the grey code boxes.

What this does

Uganda refugee identity cards have a large square **QR code** on the back. Inside that code is the card holder's information: name, date of birth, ID numbers, dates and a small photo. You send DocuScrape a picture of the QR code, and DocuScrape sends back all that information as neat, labelled fields.

Nothing is guessed or "read from the printed text". DocuScrape opens the QR code and reports exactly what is stored in it, so the values are exact.

What you need

- **Your API key.** A long secret code DocuScrape gives you. It works like a password. Never share it.
- **Your docket id.** The name of your account folder at DocuScrape. In this guide it is `docket-910`.
- **A picture of the QR code.** A JPEG or PNG file (see section 2 for how to take a good one).
- **A way to send the request.** Either a terminal window (this guide shows the exact lines to paste), a tool such as Postman, or your own developer's software.

1. Words you will see

Word	What it means
API	The way one computer program talks to another. Here, your program talks to DocuScrape.
Request	The message you send to DocuScrape (your picture plus a few details).
Response	DocuScrape's reply (the card's information).
curl	A free tool, already on most computers, that sends a request from a terminal window.
Docket	Your folder at DocuScrape. Yours is `docket-910`.
Sub-docket	A smaller folder inside a docket for one type of document. For these cards it is `refugee-identity-qr`.
Parameter	A short label you attach to a request to say what kind of document it is. For these cards: `cardtype=refugee`.
Base URL	The web address of DocuScrape: `https://docuscrape.com`.

2. Take a good picture of the QR code

This step decides whether it works. The QR on this card is very dense, so the picture must be clear.

- Photograph or crop **only the QR code**, so it fills most of the picture. A picture of the whole card is usually too small for the code to be read.
- Make it **about 1000 × 1000 pixels or larger**. (A test that worked used a 1080 × 1080 picture.)
- Hold the card flat, in good light. Avoid glare, shadows and blur.
- Keep the white border around the code inside the picture.
- Save it as a **JPEG or PNG** file, up to 25 MB. Remember where you saved it.

3. Check your account is ready (one minute, optional)

Paste this into a terminal window (replace `YOUR-API-KEY` with your key). On Windows, use the PowerShell version in section 4.

```
curl -s "https://docuscrape.com/v1/dockets/docket-910/parameters" \
  -H "Authorization: Bearer YOUR-API-KEY"
```

You should see the word `refugee` in the reply. If you see it, your account is ready. If you see an "UNAUTHORIZED" message, your API key was typed wrongly.

4. Send the QR picture — Method A (recommended)

This is the simplest way. You tell DocuScrape the document type with `cardtype=refugee` and attach your picture under the name `qr`.

Mac or Linux (Terminal):

```
curl -sS -w "\nHTTP %{http_code}\n" -X POST \
  "https://docuscrape.com/v1/dockets/docket-910/documents/extract" \
  -H "Authorization: Bearer YOUR-API-KEY" \
  -F "documentId=MY-TEST-001" \
  -F "cardtype=refugee" \
  -F "qr=@/path/to/qr-picture.jpg"
```

Windows (PowerShell). Use `curl.exe` (with the `.exe`) and a backtick at the end of each line:

```
curl.exe -sS -X POST "https://docuscrape.com/v1/dockets/docket-910/documents/extract" `
  -H "Authorization: Bearer YOUR-API-KEY" `
  -F "documentId=MY-TEST-001" `
  -F "cardtype=refugee" `
  -F "qr=@C:\Users\you\Pictures\qr-picture.jpg"
```

What to change before you run it:

Part of the line	Replace it with
`YOUR-API-KEY`	The API key DocuScrape gave you
`MY-TEST-001`	Any name you like for this card, e.g. a case number. You get it back unchanged.
`/path/to/qr-picture.jpg`	The real location of your picture. **Keep the `@` in front of it.**

Leave everything else exactly as it is. Press Enter. After a few seconds you should see the reply (section 6) and, on Mac/Linux, the line `HTTP 200`, which means success.

Optional: add the line `-H "Idempotency-Key: ANY-UNIQUE-TEXT"`. If you send the same request twice by accident, DocuScrape will treat it as one (and charge once).

5. Send the QR picture — Method B (by sub-docket name)

This does the same job but names the sub-docket instead of using a parameter. Use it only if DocuScrape told you to. The picture is attached under the sub-docket's name, `refugee-identity-qr`.

```
curl -sS -w "\nHTTP %{http_code}\n" -X POST \
  "https://docuscrape.com/v1/dockets/docket-910/documents/extract" \
  -H "Authorization: Bearer YOUR-API-KEY" \
  -F "documentId=MY-TEST-001" \
  -F "subDockets=refugee-identity-qr" \
  -F "refugee-identity-qr=@/path/to/qr-picture.jpg"
```

On Windows PowerShell, use the same layout as Method A with these two lines instead of the `cardtype` and `qr` lines:

```
  -F "subDockets=refugee-identity-qr" `
  -F "refugee-identity-qr=@C:\Users\you\Pictures\qr-picture.jpg"
```

Both methods give you the same card information. The only differences are the parameter you send and the name the results are filed under in the reply (`qr` for Method A, `refugee-identity-qr` for Method B). With Method B you may also see a harmless `warnings` note saying a parameter was not used; ignore it.

6. What you get back

DocuScrape answers with a block of text called JSON. It looks technical but reads like a list. This is a **Method A** reply. The values are made up, and the photo is shortened:


```
{
  "requestId": "dsreq-XXXXXXXXXXXXXXXXXXXXXXXX",
  "docketId": "docket-910",
  "documentId": "MY-TEST-001",
  "status": "COMPLETED",
  "parameters": {"cardtype": "refugee"},
  "sides": {
    "qr": {
      "status": "COMPLETED",
      "fields": [
        {"name": "fullName", "value": "TESTNAME ONE, TWO",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "dateOfBirth", "value": "01/02/1990", "isoValue": "1990-02-01",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "sex", "value": "Female",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "countryOfOrigin", "value": "Testland",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "individualId", "value": "T1X-000000001",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "documentNumber", "value": "T1X-00-0000001",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "issueDate", "value": "01/03/2026", "isoValue": "2026-03-01",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "expiryDate", "value": "01/03/2031", "isoValue": "2031-03-01",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "issuingOffice", "value": "Testland - City : TST - V1 IDCARD",
         "confidence": 100.0, "status": "EXTRACTED"},
        {"name": "photo", "value": "UklGR... (long text, shortened)",
         "dataType": "IMAGE", "mimeType": "image/webp",
         "confidence": 100.0, "status": "EXTRACTED"}
      ],
      "qr": {"decoded": true, "readable": true,
             "layout": "Uganda refugee ID card QR (14 fields)"}
    }
  }
}
```


A **Method B** reply is the same, with two small differences: there is a line `"subDockets": ["refugee-identity-qr"]` instead of `"parameters"`, and the results sit under `"refugee-identity-qr"` instead of `"qr"`:


```
{
  "requestId": "dsreq-XXXXXXXXXXXXXXXXXXXXXXXX",
  "status": "COMPLETED",
  "subDockets": ["refugee-identity-qr"],
  "sides": {
    "refugee-identity-qr": {
      "status": "COMPLETED",
      "fields": [ ... the same 10 fields as above ... ],
      "qr": {"decoded": true, "readable": true,
             "layout": "Uganda refugee ID card QR (14 fields)"}
    }
  }
}
```


How to read it

Item in the reply	What it means
`"status": "COMPLETED"`	DocuScrape finished. Success.
`"documentId"`	The name you gave this card, sent back unchanged.
`fullName`	The card holder's name, exactly as stored in the QR.
`dateOfBirth`	Date of birth as day/month/year. `isoValue` is the same date as year-month-day, which computers prefer.
`sex`	Sex, as stored in the QR.

Item in the reply	What it means
`countryOfOrigin`	Country the holder is from.
`individualId`	The holder's individual number stored in the QR.
`documentNumber`	The card's document number stored in the QR.
`issueDate`, `expiryDate`	When the card was issued and when it expires (day/month/year, plus `isoValue`).
`issuingOffice`	The issuing office and card type as stored in the QR.
`photo`	The holder's small photo, written as a long piece of text (see below).
`confidence: 100`	The value was read straight from the QR code, so it is exact. It is always 100 for a successful read.
`"decoded": true`	The QR code was opened successfully.

Good to know:

- The QR does not label its values. The names above follow the order in which the QR stores them. If a name ever looks wrong against the printed card, please tell DocuScrape.
- DocuScrape reads what is in the QR. It does **not** check that the card is genuine and does not look the card up in any government record. Treat it as fast, accurate data entry, not as proof of identity.

Turning the photo text into a picture

The photo arrives as a long piece of text. Copy everything between the quotation marks after `"value":` in the `photo` field, then turn it into a picture file:

Mac or Linux (save the whole reply first as `response.json`; needs the free tool `jq`):

```
jq -r '.sides.qr.fields[] | select(.name=="photo") | .value' response.json \
  | base64 -d > photo.webp
```

Windows (PowerShell) (paste the long text where it says `PASTE-HERE`):

```
[IO.File]::WriteAllBytes("photo.webp", [Convert]::FromBase64String("PASTE-HERE"))
```

Then open `photo.webp` (most browsers can open it). It is a small picture, about 50 × 50 pixels. Developers: it is a base64-encoded WebP image.

7. If something goes wrong

What you see	What it means	What to do
`"decoded": false` and `QR_NOT_DECODED`	DocuScrape could not open the QR code. The picture is usually too small, blurry or has glare.	Take a new close-up (section 2) and send it again.
`HTTP 401` / `UNAUTHORIZED`	The API key is missing, mistyped or no longer valid.	Check the key, keeping the word `Bearer` and a space before it.
`HTTP 404` / `DOCKET_NOT_FOUND`	The docket name is wrong.	Use `docket-910` exactly.
`PARAMETER_REQUIRED` or `INVALID_PARAMETER_VALUE`	`cardtype` is missing or has the wrong value.	Send exactly `cardtype=refugee`.
`MISSING_DOCUMENT`	DocuScrape did not receive the picture.	Check the `@` in front of the file path, and that the part is named `qr` (Method A).
`HTTP 415` / `UNSUPPORTED_MEDIA_TYPE`	The file is not a JPEG or PNG.	Save the picture as JPEG or PNG.
`HTTP 413` / `FILE_TOO_LARGE`	The picture is over 25 MB.	Use a smaller file.
`HTTP 402`	Your account is out of credits.	Ask DocuScrape to add credits, then try again.
`HTTP 409` / `IDEMPOTENCY_CONFLICT`	You reused an `Idempotency-Key` with a different picture.	Use a new key for each new card.
`HTTP 202` / `QUEUED` or `PROCESSING`	Still working.	Wait a moment, then ask again (section 8).

When the QR cannot be opened, the reply looks like this (shortened). The fields come back as `NOT_FOUND`, and the overall status may read `COMPLETED_WITH_WARNINGS`:

```
"qr": {
  "decoded": false,
  "error": {
    "code": "QR_NOT_DECODED",
    "message": "No QR code could be decoded from the image. Send a sharp, ..."
  }
}
```

8. If the reply says it is still processing

Most requests finish in a few seconds. If you see `"status": "QUEUED"` or `"PROCESSING"`, copy the `requestId` from the reply and ask again until it says `COMPLETED`:

```
curl -s "https://docuscrape.com/v1/extractions/REQUEST-ID-HERE" \
  -H "Authorization: Bearer YOUR-API-KEY"
```

9. Quick checklist

1. Take a close-up, sharp picture of the QR only (section 2).
2. Run the check in section 3 and confirm you see `refugee`.
3. Run the request in section 4 with your key and your picture.
4. Look for `HTTP 200`, `"status": "COMPLETED"` and `"decoded": true`.
5. Read the card's fields from the reply, and save the photo if you need it (section 6).
6. Try one blurry picture to see how an error looks, so your team knows what to expect (section 7).

For developers (one-minute summary)

Item	Value
Endpoint	`POST https://docuscrape.com/v1/dockets/docket-910/documents/extract`
Auth	Header `Authorization: Bearer <key>`
Body	`multipart/form-data`: `documentId`, `cardtype=refugee`, file part `qr` (JPEG or PNG, up to 25 MB)
Result	`sides.qr.fields[]` (name, value, confidence, status, plus `isoValue` for dates and `mimeType` for the photo), `sides.qr.qr`
Credits	1 per image
Polling	`GET /v1/extractions/{requestId}` while `status` is `QUEUED` or `PROCESSING` (HTTP 202)
Retries	Send an `Idempotency-Key` header so a retry never creates or charges twice