

100% OFFLINE · 440 RULES · 218 COUNTRIES

# Global TIN Validator

## User Guide & Administrator Handbook

How to validate Tax Identification Numbers — single checks, bulk files, reading results, and maintaining the country rule set.

THEFAXMINATOR 2.0

# What the program does

1

## Checks TINs against rules

Validates a Tax Identification Number against country-specific regular-expression patterns.

2

## 100% offline

Runs entirely in your browser. No internet, no server, no data ever leaves your machine.

3

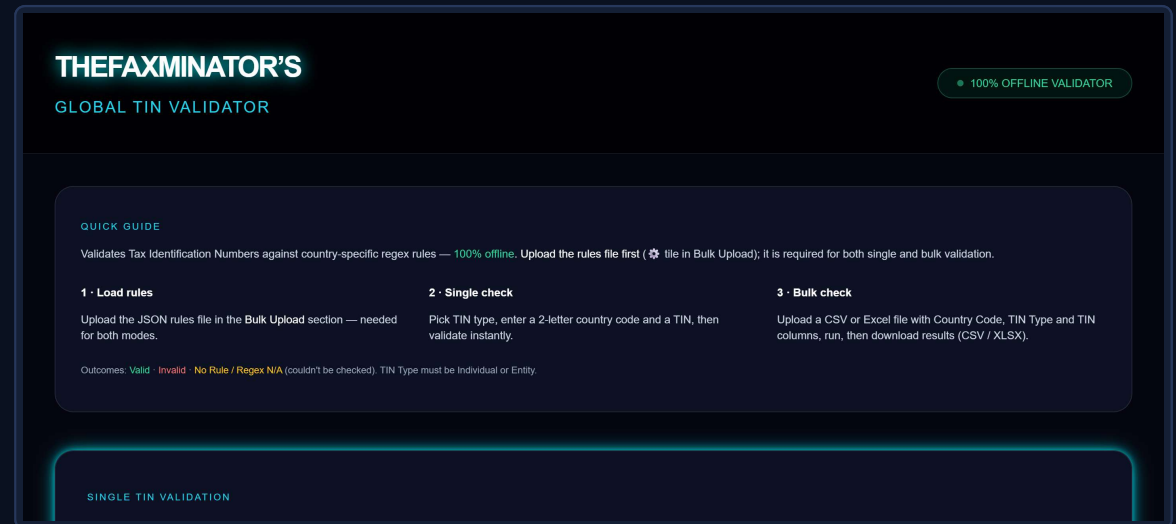
## Two ways to work

Validate one TIN instantly, or upload a whole spreadsheet for bulk validation.

4

## Covers the world

440 rules across 218 countries, for both Individual and Entity taxpayer types.



The app on launch — title, quick guide, and the single-validation form.

# What you need to run it

Keep these three files together in the same folder. Nothing else is required.

1

`Global-Tin-Validator-Offline.html`

The application. Double-click to open it in any modern web browser.

2

`xlsx.full.min.js`

Excel engine. Must sit next to the HTML — it reads and writes .xlsx files.

3

`validation-rules.json`

The rule set (440 rules). You upload this inside the app each session.

**Tip:** a sample dataset (`tin-data-test.xlsx`) is included so you can try bulk validation right away.

# How it works — three steps

The same rule set powers both single and bulk validation.

**1**

## Load the rules

Open the app and upload validation-rules.json via the gear (⚙️) tile. You will see "440 rule(s) loaded".

**2**

## Validate

Check a single TIN in the form, or upload a CSV / Excel dataset and run bulk validation.

**3**

## Read & export

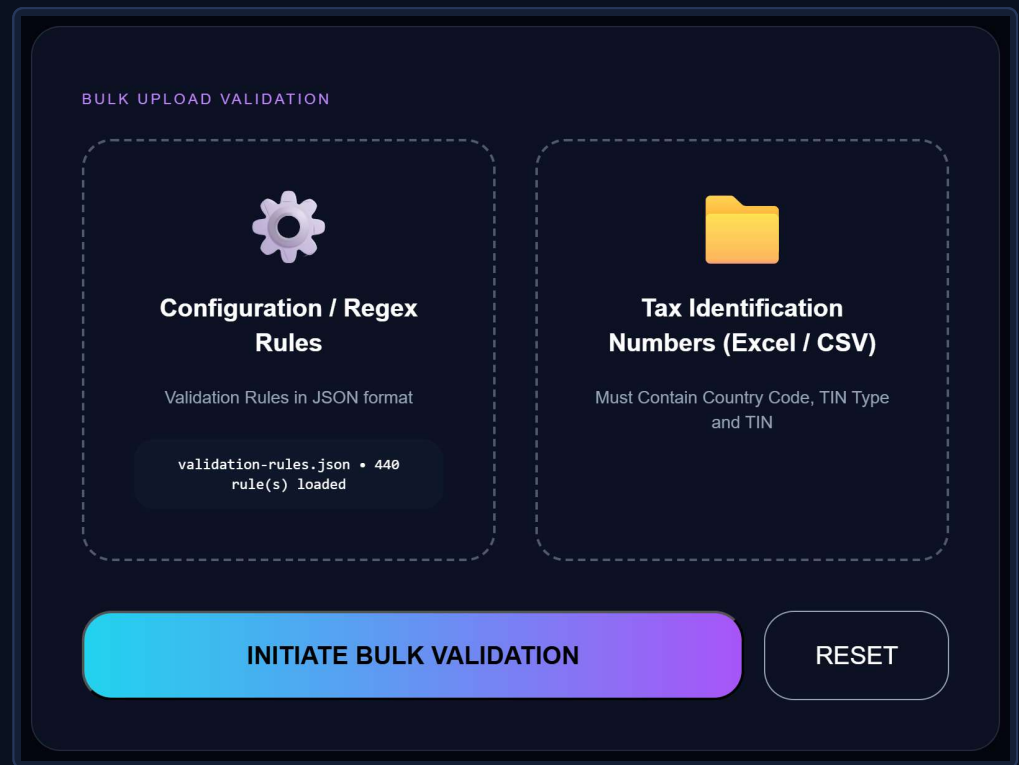
See Valid / Invalid / No-Rule outcomes and statistics. Download bulk results as CSV or XLSX.

# Step 1 — Load the rules file

Required before anything works, for BOTH single and bulk validation.

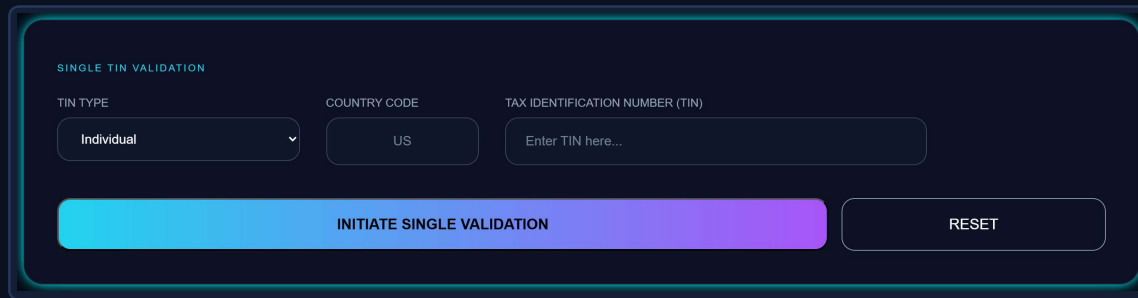
- 1 Scroll to the BULK UPLOAD VALIDATION section.
- 2 Click the gear tile — "Configuration / Regex Rules".
- 3 Choose validation-rules.json from the folder.
- 4 Confirm the tile shows "440 rule(s) loaded".

⚠ **Until rules are loaded**, every check returns "Validation rule missing." The rules are not saved — re-upload them each time you open the app.



# Single TIN validation

Check one number instantly — ideal for spot checks.



The screenshot shows a web form titled "SINGLE TIN VALIDATION". It contains three input fields: "TIN TYPE" with a dropdown menu showing "Individual", "COUNTRY CODE" with a text input showing "US", and "TAX IDENTIFICATION NUMBER (TIN)" with a text input showing "Enter TIN here...". Below these fields are two buttons: a large blue button labeled "INITIATE SINGLE VALIDATION" and a smaller white button labeled "RESET".

**1**

## TIN Type

Pick Individual or Entity — rules differ by type.

**2**

## Country Code

Enter the 2-letter ISO code (e.g. DE, US, NO).

**3**

## TIN

Type the number, then click INITIATE SINGLE VALIDATION.

# Reading the result

Every check returns one of four clearly colour-coded outcomes.

The screenshot shows the 'SINGLE TIN VALIDATION' form with the following inputs: TIN TYPE: Individual, COUNTRY CODE: DE, TAX IDENTIFICATION NUMBER (TIN): 12345678901. The 'INITIATE SINGLE VALIDATION' button is highlighted in blue. Below the button, a green checkmark icon is displayed next to the text 'VALID TIN'. Further details include 'Country: Germany (DE) • Type: Individual' and '12345678901'. At the bottom, it states 'Matched rule: \*([0-9]{11})\$'.

## VALID

Matches a rule pattern.

The screenshot shows the 'SINGLE TIN VALIDATION' form with the following inputs: TIN TYPE: Individual, COUNTRY CODE: DE, TAX IDENTIFICATION NUMBER (TIN): DE-123-ABC. The 'INITIATE SINGLE VALIDATION' button is highlighted in blue. Below the button, a red 'X' icon is displayed next to the text 'INVALID TIN'. Further details include 'Country: Germany (DE) • Type: Individual' and 'DE-123-ABC'. At the bottom, it states 'Checked against 1 configured pattern(s)'.

## INVALID

Format does not match.

The screenshot shows the 'SINGLE TIN VALIDATION' form with the following inputs: TIN TYPE: Individual, COUNTRY CODE: ZZ, TAX IDENTIFICATION NUMBER (TIN): 12345. The 'INITIATE SINGLE VALIDATION' button is highlighted in blue. Below the button, a document icon is displayed next to the text 'VALIDATION RULE MISSING'. Further details include 'Validation rule missing for ZZ (Individual)' and 'Country Code: ZZ • Type: Individual'.

## NO RULE

No pattern for that country/type.

A fourth outcome — **REGEX N/A** — appears in bulk runs: a rule exists for the country but has no usable pattern, so the TIN cannot be checked. **Both NO RULE and REGEX N/A mean "not checked", not "invalid".**

# Bulk validation — prepare your file

A CSV or Excel (.xlsx) file. Column names are matched flexibly (case & spacing ignored).

| Column                    | Required? | Purpose / accepted names                   |
|---------------------------|-----------|--------------------------------------------|
| Country Code              | Required  | 2-letter code. Also: Country, Jurisdiction |
| TIN Type                  | Required  | Individual or Entity. Also: Type, Category |
| Tax Identification Number | Required  | The number. Also: TIN, Tax ID              |
| Record ID                 | Optional  | Row reference. Also: ID, Record            |
| Entity                    | Optional  | Name shown in preview. Also: Name, Company |

Only the three required columns are checked. Extra columns are ignored.

## EXAMPLE ROW

Record ID                      **REC-1001**

Entity                              **Private Person**

Country Code                      **NO**

Tax Identification No.              **82117318333**

Type                                  **Individual**

The included `tin-data-test.xlsx` has  
10,000 ready-made rows.

# Bulk validation — load & run

Rules + dataset loaded; the preview confirms the first records before you run.

BULK UPLOAD VALIDATION



### Configuration / Regex Rules

Validation Rules in JSON format

validation-rules.json • 440 rule(s) loaded



### Tax Identification Numbers (Excel / CSV)

Must Contain Country Code, TIN Type and TIN

tin-data-test.xlsx • 10000 record(s) loaded

INITIATE BULK VALIDATION

RESET

PREVIEW • FIRST 10 RECORDS

| Record ID | Entity           | Country Code | Tax Identification Number | Type       |
|-----------|------------------|--------------|---------------------------|------------|
| REC-1001  | Private Person   | NO           | 82117318333               | Individual |
| REC-1002  | Private Person   | NO           | 15108479589               | Individual |
| REC-1003  | Private Person   | NO           | 28108822409               | Individual |
| REC-1004  | Nordic Logistics | NO           | 163254636                 | Entity     |
| REC-      | Global           |              |                           |            |

1

## Load rules

Upload validation-rules.json in the gear tile (if not already loaded).

2

## Upload dataset

Click the folder tile and pick your CSV / XLSX file.

3

## Check preview

The right panel shows the first 10 records and detected columns.

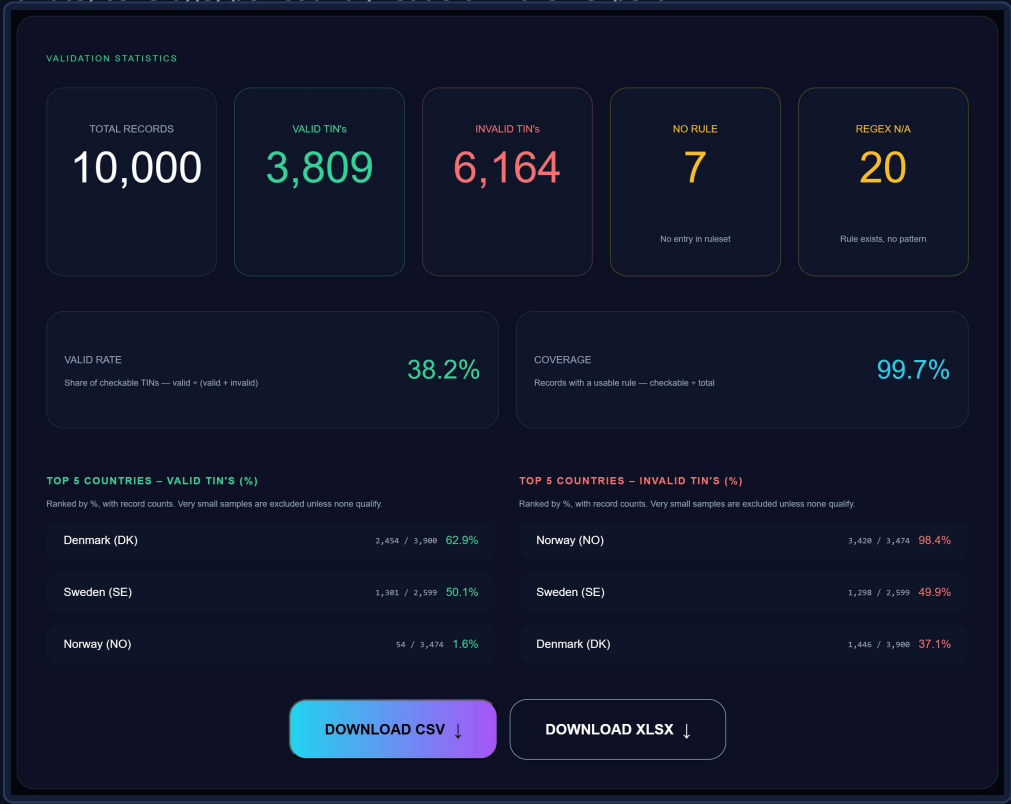
4

## Run

Click INITIATE BULK VALIDATION.

# Bulk results & statistics

Totals, valid rate, coverage, per-country leaders — then export.



### Valid / Invalid

Counts of TINs that matched or failed their pattern.



### No Rule / Regex N/A

Records that could not be checked (no usable rule).



### Valid rate

$\text{valid} \div (\text{valid} + \text{invalid})$  — quality of checkable TINs.



### Coverage

$\text{checkable} \div \text{total}$  — how much of the file had rules.

↓ **Export:** DOWNLOAD CSV or DOWNLOAD XLSX saves every row with its result.

# Admin — inside the rules file

validation-rules.json is one "validators" array. Each entry is one country + type.

```
{
  "validators": [
    {
      "CountryCode": "DE",
      "Country": "Germany",
      "Region": "Europe",
      "Type": "Individual",
      "TaxIdentificationNumber": {
        "regex": [
          "^(?!([0-9])\\1{10})\\d{11}$"
        ]
      }
    }
  ]
}
```

## CountryCode

2-letter ISO code that must match the input.

## Type

Individual or Entity — each is a separate entry.

## Country / Region

Labels shown in results (not used to match).

## regex

Array of patterns. A TIN is VALID if it matches ANY one.

# Admin — adding or editing a rule

Edit the JSON in any text editor. No build step — just save and re-upload.

**1**

## Find or add the entry

Search for the CountryCode + Type. To add a country, copy an existing block and change the values.

**2**

## Write the pattern

Put one or more regular expressions in the regex array. Remember JSON needs double back-slashes: `\\d`, `\\1`.

**3**

## Validate the JSON

Make sure commas and brackets are intact — one syntax error stops the whole file from loading.

**4**

## Save & re-upload

Save the file, then upload it again in the app. The tile shows the new rule count.

# Troubleshooting & tips

## ● Every TIN says "rule missing"

The rules file is not loaded. Upload validation-rules.json in the gear tile first.

## ● Bulk button does nothing / errors

Check both tiles are filled (rules + dataset) and the dataset has the 3 required columns.

## ● Lots of "No Rule" results

Country codes may be wrong (use 2-letter ISO), or those countries have no rule yet.

## ● Excel export fails

Make sure `xlsx.full.min.js` is in the same folder as the HTML file.

## ● Valid number flagged invalid

The rule may be too strict — check/adjust the country regex (see admin section).

## ● Rules disappear after reopening

By design — the app never stores data. Re-upload the rules each session.

## QUICK REFERENCE

# You're ready to validate

- **Open** — Double-click Global-Tin-Validator-Offline.html (keep xlsx.full.min.js beside it).
- **Load rules** — Upload validation-rules.json in the gear tile — every session.
- **Single check** — Type + country code + TIN → INITIATE SINGLE VALIDATION.
- **Bulk check** — Upload CSV/XLSX (Country Code, TIN Type, TIN) → run → download results.
- **Admin** — Edit regex in validation-rules.json, save, re-upload.