wired in fabrick

judopay

Judopay Token Import

Import File Specification

January 2026

DISCLAIMER

This document is provided for informational purposes only. The information contained herein is derived from Judopay's publicly available API documentation and is subject to change without notice.

Merchants are solely responsible for ensuring that:

- They are authorised to transfer and process the card data provided;
- All data supplied complies with applicable PCI DSS, data protection, and other relevant regulatory obligations;
- The data provided for import is accurate, complete, and suitable for its intended purpose

Judopay will process any imported data in accordance with applicable security standards, the parties' existing contractual arrangements, and its own compliance obligations.

Judopay maintains certifications to recognised industry standards, including PCI DSS and ISO standards, in relation to its own operations. Nothing in this document shall be construed as transferring or extending such certifications or compliance obligations to any third party.

Judopay accepts no liability for issues arising from data supplied by the merchant or third parties.

This document does not form part of, nor does it amend, any contractual agreement between the merchant and Judopay.

CONTENTS

Judopay Token Import – Import File Specification

1. Overview

When migrating stored card tokens from a previous payment provider to Judopay, merchants **must supply a single encrypted file** containing the card data to be imported. This import file is processed by the Judopay Import Tool.

This document explains:

- The attributes expected by the Judopay Import Tool.
- Format and validation rules applied during import.
- How migrated tokens are used for future Merchant Initiated Transactions (MITs).

---

2. File delivery and security requirements

2.1 Encryption

- The import file must be encrypted using Judopay's PGP public key before delivery.
- Encryption is **mandatory** to maintain PCI DSS compliance.
- Judopay will provide the correct public key and secure transfer instructions ahead of migration.

2.2 File format

- Preferred format: CSV
- Structure:
    - One row per card token
    - Header row required
- JSON formats may be supported by exception, but **CSV is strongly recommended**.

---

3. Field name attributes (per row)

Each row in the file represents one stored card and must include the following **mandatory** fields.

Note: The file from the previous provider does not have to use these field names, but a mapping from the field names they do use to the ones we use must be provided.

3.1 `cardToken`

A unique identifier for the stored card.

Note: It is not mandatory (typically this will be provided by the merchant, but if not provided Judopay can import and generate a new `cardToken` associated with the consumer reference and card details).

Format restrictions:

- Length: 10–50 characters
- Allowed characters: alphanumeric (**a–z, A–Z, 0–9**), hyphen (**-**), underscore (**_**)
- Must not be a 13–19 digit numeric sequence that passes a Luhn check (to avoid being mistaken for a PAN)

  If numeric identifiers are used, they must be prefixed (for example, **Token_123456**).

Typical source:

- Previous provider token ID (for example, Opayo **TokenID** GUID)

3.2 `yourConsumerReference`

A reference identifying the consumer with the merchant.

Format restrictions:

- Length: 1–50 characters
- Must not contain control characters (**<, \\, \r, \n, \f, \b**)
- Should not resemble a card number (Luhn-valid sequence)

Important notes:

- Should not be derived from cardholder name (PCI restriction)
- Must remain stable for all future MIT payments

Common approaches:

- Reuse the previous provider token ID
- Use a Judopay-generated static consumer reference (if agreed in advance)

3.3 `cardNumber`

The full Primary Account Number (PAN).

Format restrictions:

- Digits only, or digits with spaces / dashes
- Will be tokenised and never stored in plaintext

3.4 `expiryDate`

Card expiry date.

Expected format:

- Month and year provided
- Internally normalised to **MM/YY** (for example, **12/30**)

Note: **Expired cards** should be **excluded** from the import.

---

4. Optional attributes

4.1 `cardHolderName`

- Max length: 45 characters
- **Must not** contain control characters
- **Must not** resemble a card number

4.2 Billing address fields

All address fields **must not** resemble PANs and **must not** contain control characters.

Field	Notes
`address1`	Optional
`address2`	Optional
`town`	Optional
`postCode`	**Required** if other address fields are present. max 20 chars; regex **^[0-9a-zA-Z -]*$**
`state`	Optional ISO 2-char state (US/CA only)
`countryCode`	Optional ISO 3166-1 numeric (3-digit) e.g. **826**

---

5. Merchant Initiated Transactions (MIT) after migration

For recurring or follow-on payments, merchants **must** submit MIT requests using three references:

1. `cardToken`
2. `yourConsumerReference`
3. `RelatedPaymentNetworkTransactionId`

5.1 `RelatedPaymentNetworkTransactionId`

- Sourced from the previous provider's export
- Represents the original network transaction identifier
- Judopay does not generate this value

---

6. Merchant lookup table (**required**)

Merchants **must** maintain an internal lookup table mapping:

- Internal recurring payment reference
- → `cardToken`
- → `yourConsumerReference`
- → `RelatedPaymentNetworkTransactionId`

This table is used to populate MIT requests after migration.
Judopay can provide a post-import mapping extract, but ongoing maintenance remains the merchant's responsibility.

---

7. Token volume considerations

- Judopay will import all supplied tokens that are **not associated** with expired cards (and have the mandatory elements supplied - consumer reference, PAN, expiry date)
- There is no technical disadvantage to importing unused tokens.
- Merchants may optionally restrict the file to active tokens only.

---

8. Example CSV (illustrative)

```
None
cardToken,yourConsumerReference,cardNumber,expiryDate,car
dHolderName,cardAddress.address1,cardAddress.town,cardAdd
ress.postCode,cardAddress.countryCode,
F86EA57D-50E1-4450-943F-84A2D183843B,4111 1111 1111
1111,12/30,John Smith,1 High Street,London,SW1A
1AA,826,V483338411387348
9EF3CCB3-BEA3-4028-9BC0-91E516F4D73D,9EF3CCB3-BEA3-4028-9
BC0-91E516F4D73D,5555-5555-5555-4444,06/29,Jane Doe,10
Market Road,Manchester,M1 1AA,826,MN99287466234109
```

Note: Values above are **examples only**. Real files must be encrypted before transfer.

---

9. Pre-migration prerequisites

Before migration can proceed, the following **must** be supplied separately:

- Merchant Judo ID (supplied by Judopay Onboarding)
- Production API token with Register Cards permission (supplied by Judopay onboarding)
- Confirmation of consumer reference strategy (supplied by merchant)
- Encrypted file transfer details (supplied by merchant)

---

10. Summary

To ensure a smooth migration:

- Supply one encrypted CSV file
- Ensure all mandatory fields meet validation rules
- Use stable, non-PCI consumer references
- Maintain a merchant-side lookup table for MIT payments

Once validated, Judopay will **securely** import the tokens and confirm completion.


wired in fabrick
judopay