

DLRL Koha Patron API

Complete User Manual

Base URL: value of `app.baseUrl` in `.env` (example: `http://14.139.92.12/dlrl_koha_api/`)

Authentication: HTTP Basic Auth — from `.env` (`api.basic.username` / `api.basic.password`)

Default credentials: DLRL / DLRL@2026

Content-Type: `application/json`

Clean URLs: Do **not** include `index.php` in the URL

Production config file: `.env` only — change settings there (no PHP file edits needed)

1. Table of Contents

- 1. API Overview
- 2. Production Configuration (.env)
- 3. Authentication (all APIs)
- 4. Image File Format (imagefile / mimetype)
- 5. Attribute Codes & Category Codes
- 6. API 1 — Upsert Patron
- 7. API 2 — View Patron
- 8. API 3 — Overdue List
- 9. Common Error Messages
- 10. Checklist

2. API Overview

Method	Endpoint	Purpose
POST	/api/patron/upsert	Insert new patron or update existing (single or multiple)
GET	/api/patron/view	Get all patrons
GET	/api/patron/view?cardnumber=XXX	Get one patron by cardnumber
GET	/api/patron/overdue	Get all overdue issues
GET	/api/patron/overdue?cardnumber=XXX	Get overdue for one patron

Full example URL: `{app.baseUrl}api/patron/view` → e.g. `http://14.139.92.12/dlrl_koha_api/api/patron/view`

3. Production Configuration (.env)

File to change for production: .env only.
Do not hard-code credentials or DB settings in PHP files. All runtime settings are loaded from .env.

3.1 What you must edit

File	When to change
.env	Always — base URL, Basic Auth, database, branch code, attribute codes, environment
public/.htaccess → RewriteBase	Only if the public URL folder name changes (must match the path part of app.baseURL)

If .env is missing on the server, copy from the template file env and rename/save as .env.

3.2 Complete .env keys

```
# ENVIRONMENT
CI_ENVIRONMENT = production

# APP
app.baseURL = 'http://14.139.92.12/dlrl_koha_api/'
app.indexPage = ''

# DATABASE (Koha)
database.default.hostname = 14.139.92.12
database.default.database = koha_library
database.default.username = cqr
database.default.password = *****
database.default.DBDriver = MySQLi
database.default.port = 3306
database.default.DBDebug = false

# API BASIC AUTH
api.basic.username = DLRL
api.basic.password = DLRL@2026

# PATRON API OPTIONS
api.defaultBranchCode = DLRL
api.attributeCodes = DESIG,DIV,RT
```

3.3 How to change each setting

Need to change	.env key	Example
Base URL / host	app.baseURL	app.baseURL = 'http://NEW_HOST/dlrl_koha_api/'
API username	api.basic.username	api.basic.username = DLRL
API password	api.basic.password	api.basic.password = DLRL@2026
DB host	database.default.hostname	database.default.hostname = 14.139.92.12
DB name	database.default.database	database.default.database = koha_library
DB user	database.default.username	database.default.username = cqr
DB password	database.default.password	database.default.password = ****
Default branch	api.defaultBranchCode	api.defaultBranchCode = DLRL
Attribute codes	api.attributeCodes	api.attributeCodes = DESIG,DIV,RT
App mode	CI_ENVIRONMENT	CI_ENVIRONMENT = production

After changing Basic Auth password, update Postman / client credentials to match.
Always keep a trailing slash / on app.baseURL.

4. Authentication (Required for All APIs)

```
Authorization: Basic <base64(username:password)>
Content-Type: application/json
```

Default (from .env):

```
Username: DLRL
Password: DLRL@2026
Header: Authorization: Basic RExSTDpETFJMQDIwMjY=

base64("DLRL:DLRL@2026") → RExSTDpETFJMQDIwMjY=
```

4.1 Auth Failure — Missing header

HTTP 401

```
{
  "status": false,
  "message": "Authorization header is required for Basic Authentication. Example: Authorization: Basic base64(username:password)"
}
```

4.2 Auth Failure — Wrong format

HTTP 400

```
{
  "status": false,
  "message": "Invalid Authorization header format. Expected: Basic <base64(username:password)>."
}
```

4.3 Auth Failure — Wrong username/password

HTTP 403

```
{
  "status": false,
  "message": "Invalid username or password."
}
```

5. Image File Format (imagefile / mimetype)

Images are optional on upsert. When sent, use a **Base64 Data URI**:

```
data:image/<type>;base64,<BASE64_DATA>
```

Part	Meaning	Example
data:image/	Fixed prefix	data:image/
jpeg / png / ...	Image type	jpeg
;base64,	Fixed separator	;base64,
Base64 string	Encoded binary	/9j/4AAQSkZJRg...

```
"imagefile": "...",
"mimetype": "image/jpeg"
```

Valid: data:image/jpeg;base64,... Or data:image/png;base64,...
Raw base64 without prefix is accepted; default mime is image/jpeg unless mimetype is given.

Wrong input	Result
-------------	--------

imagefile omitted / empty	No image change; other fields still save
Invalid / corrupted base64	Error: <i>imagefile is not valid base64 data.</i>
File path or URL string	Invalid base64 → same error
Multipart file upload	Not supported — must be JSON string

```
{
  "status": false,
  "message": "[CQR40] cardnumber \"CQR40\": imagefile is not valid base64 data.",
  "success_count": 0,
  "failed_count": 1,
  "results": [
    {
      "success": false,
      "index": 0,
      "cardnumber": "CQR40",
      "action": "insert",
      "message": "cardnumber \"CQR40\": imagefile is not valid base64 data."
    }
  ]
}
```

5A. Attribute Codes & Category Codes

Attribute codes (borrower_attributes)

These keys can be sent in the upsert JSON. They are stored in `borrower_attributes` as `code` + `attribute` (value).

Code (JSON key)	Description	Example value
DESIG	Designation	SCIENTIS-E
DIV	DIVISION	MV-3
RT	RETIRED_TRANSFERED	No / Yes

```
"DESIG": "SCIENTIS-E",  
"DIV": "MV-3",  
"RT": "No"
```

Configured in `.env` as `api.attributeCodes = DESIG,DIV,RT`.

Category codes (categories)

Use one of these values in the mandatory `categorycode` field. The code must already exist in Koha.

Category code	Description
OFC	Officers
STAFF	Staff

```
"categorycode": "OFC"
```

Sending a `categorycode` that is not in Koha (for example an unknown code) will fail with a clear validation error.

6. API 1 — Upsert Patron

Item	Value
Method	POST
URL	{app.baseURL}api/patron/upsert
Auth	Basic Auth required
Body	JSON

6.1 Behaviour

Situation	Action
cardnumber does not exist	INSERT
cardnumber already exists	UPDATE only sent keys
Multiple objects in data	Each patron processed independently

6.2 Mandatory fields

Mode	Mandatory	Empty value rule
Insert	cardnumber, surname, categorycode	Must be present and non-empty
Update	cardnumber	If surname / categorycode / cardnumber is sent, value must not be empty

Optional fields: firstname, title, othernames, initials, streetnumber, address, address2, city, state, zipcode, country, email, phone, dateofbirth, dateenrolled, dateexpiry, sex, userid, password, branchcode (default from api.defaultBranchCode), imagefile, mimetype

See **section 5A** for attribute codes (DESIG / DIV / RT) and category codes (OFC / STAFF).

categorycode must be a valid Koha category (e.g. OFC OR STAFF).
branchcode must exist in Koha branches.
JSON must be valid — no trailing comma.

6.3 Success — Single insert request

```
{
  "data": [
    {
      "cardnumber": "CQR40",
      "surname": "VVVV",
      "firstname": "K",
      "title": "CQR40",
      "othernames": "VK",
      "initials": "K",
      "streetnumber": "123",
      "address": "Coimbatore",
      "address2": "",
      "city": "Mettupalayam",
      "state": "TamilNadu",
      "zipcode": "641301",
      "country": "India",
      "email": "vk123@gmail.com",
      "phone": "1234567890",
      "dateofbirth": "2000-01-01",
      "categorycode": "OFC",
      "dateenrolled": "2024-09-19",
      "dateexpiry": "2040-09-19",
      "sex": "M",
      "userid": "CQR40",
      "password": "CQR40",
      "imagefile": "...",
      "mimetype": "image/jpeg",
      "DESIG": "SCIENTIS-E",
      "DIV": "MV-3",
      "RT": "No"
    }
  ]
}
```

6.4 Success — Single insert response

HTTP 200

```
{
  "status": true,
  "message": "All patrons processed successfully.",
  "success_count": 1,
  "failed_count": 0,
  "results": [
    {
      "success": true,
      "index": 0,
      "cardnumber": "CQR40",
      "borrowernumber": 101,
      "action": "insert",
      "message": "Patron inserted successfully."
    }
  ]
}
```

6.5 Success — Multiple patrons request

```
{
  "data": [
    {
      "cardnumber": "CQR40",
      "surname": "VVVV",
      "firstname": "K",
      "categorycode": "OFC",
      "email": "vk123@gmail.com",
      "phone": "1234567890",
      "userid": "CQR40",
      "password": "CQR40",
      "DESIG": "SCIENTIS-E",
      "DIV": "MV-3",
      "RT": "No"
    },
    {
      "cardnumber": "CQR41",
      "surname": "Kumar",
      "firstname": "Ravi",
      "categorycode": "OFC",
      "email": "ravi@example.com",
      "phone": "9876543210",
      "userid": "CQR41",
      "password": "CQR41",
      "DESIG": "SCIENTIST",
      "DIV": "MV-1",
      "RT": "Yes"
    },
    {
      "cardnumber": "CQR42",
      "email": "updated@example.com",
      "phone": "9000000001",
      "DESIG": "OFFICER"
    }
  ]
}
```

6.6 Success — Multiple patrons response

HTTP 200

```
{
  "status": true,
  "message": "All patrons processed successfully.",
  "success_count": 3,
  "failed_count": 0,
  "results": [
    {
      "success": true,
      "index": 0,
      "cardnumber": "CQR40",
      "borrowernumber": 101,
      "action": "insert",
      "message": "Patron inserted successfully."
    },
    {
      "success": true,
      "index": 1,

```

```

    "cardnumber": "CQR41",
    "borrowernumber": 102,
    "action": "insert",
    "message": "Patron inserted successfully."
  },
  {
    "success": true,
    "index": 2,
    "cardnumber": "CQR42",
    "borrowernumber": 103,
    "action": "update",
    "message": "Patron updated successfully."
  }
]
}

```

6.7 Success — Partial update request

```

{
  "data": [
    {
      "cardnumber": "CQR40",
      "email": "new.email@gmail.com",
      "phone": "9876543210",
      "DESIG": "SCIENTIST"
    }
  ]
}

```

6.8 Success — Update response

HTTP 200

```

{
  "status": true,
  "message": "All patrons processed successfully.",
  "success_count": 1,
  "failed_count": 0,
  "results": [
    {
      "success": true,
      "index": 0,
      "cardnumber": "CQR40",
      "borrowernumber": 101,
      "action": "update",
      "message": "Patron updated successfully."
    }
  ]
}

```

6.9 Failure — Missing mandatory field

HTTP 400

```

{
  "status": false,
  "message": "[CQR40] cardnumber \"CQR40\": Insert failed. Missing mandatory field(s): surname. Required for insert: cardnumber, surname, categorycode.",
  "success_count": 0,
  "failed_count": 1,
  "results": [
    {
      "success": false,
      "index": 0,
      "cardnumber": "CQR40",
      "action": "insert",
      "message": "cardnumber \"CQR40\": Insert failed. Missing mandatory field(s): surname. Required for insert: cardnumber, surname, categorycode."
    }
  ]
}

```

6.10 Failure — Empty mandatory value on update

```

{ "data": [ { "cardnumber": "CQR40", "surname": "" } ] }

```

HTTP 400

```
{
  "status": false,
  "message": "[CQR40] cardnumber \"CQR40\": Mandatory field \"surname\" cannot be empty. Provide a valid value or omit the key.",
  "success_count": 0,
  "failed_count": 1,
  "results": [
    {
      "success": false,
      "index": 0,
      "cardnumber": "CQR40",
      "action": "update",
      "message": "cardnumber \"CQR40\": Mandatory field \"surname\" cannot be empty. Provide a valid value or omit the key."
    }
  ]
}
```

6.11 Failure — Invalid categorycode

HTTP 400

```
{
  "status": false,
  "message": "[CQR40] cardnumber \"CQR40\": Invalid categorycode \"XYZ\". This value does not exist in the categories table. Please use a valid categorycode.",
  "success_count": 0,
  "failed_count": 1,
  "results": [
    {
      "success": false,
      "index": 0,
      "cardnumber": "CQR40",
      "action": "insert",
      "message": "cardnumber \"CQR40\": Invalid categorycode \"XYZ\". This value does not exist in the categories table. Please use a valid categorycode."
    }
  ]
}
```

6.12 Failure — Invalid JSON

HTTP 400

```
{
  "status": false,
  "message": "Invalid JSON body: Syntax error"
}
```

7. API 2 — View Patron

Item	Value
Method	GET
All	{app.baseUrl}api/patron/view
One	{app.baseUrl}api/patron/view?cardnumber=CQR40
Auth	Basic Auth required

7.1 Success

HTTP 200

```
{
  "status": true,
  "message": "Patron details fetched successfully.",
  "count": 1,
  "data": [
    {
      "borrowernumber": 101,
      "cardnumber": "CQR40",
      "surname": "VVVV",
      "firstname": "K",
      "email": "vk123@gmail.com",
      "categorycode": "OFC",
      "branchcode": "DLRL",
      "imagefile": "/9j/4AAQ...",
      "mimetype": "image/jpeg",
      "DESIG": "SCIENTIS-E",
      "DIV": "MV-3",
      "RT": "No",
      "codes_attributes": [
        { "code": "DESIG", "attribute": "SCIENTIS-E" },
        { "code": "DIV", "attribute": "MV-3" },
        { "code": "RT", "attribute": "No" }
      ]
    }
  ]
}
```

Password is never returned.

7.2 Failure — Not found

HTTP 404

```
{
  "status": false,
  "message": "No patron found for cardnumber: CQR999",
  "data": []
}
```

8. API 3 — Overdue List

Item	Value
Method	GET
All	{app.baseUrl}api/patron/overdue
One	{app.baseUrl}api/patron/overdue?cardnumber=CQR40
Auth	Basic Auth required

Designation comes from attribute code DESIG.

8.1 Success — Records found

HTTP 200

```
{
  "status": true,
  "message": "Overdue records fetched successfully.",
  "count": 1,
  "data": [
    {
      "S_No": 1,
      "ID_No": "CQR40",
      "Name": "VVVV",
      "Designation": "SCIENTIS-E",
      "Acc_No": "B00123",
      "Title": "Introduction to Algorithms",
      "Author": "Cormen",
      "Issue_Date": "01/01/2026",
      "Due_Date": "15/01/2026",
      "Overdue_Days": 22
    }
  ]
}
```

8.2 Success — No records

HTTP 200

```
{
  "status": true,
  "message": "No overdue records found.",
  "count": 0,
  "data": []
}
```

9. Common Error Messages

Message	What to do
Authorization header is required...	Add Basic Auth
Invalid username or password.	Check api.basic.username / api.basic.password in .env
Invalid JSON body...	Fix JSON (no trailing commas)
Missing mandatory field(s)...	Send cardnumber, surname, categorycode for insert
Mandatory field "X" cannot be empty...	Provide value or omit the key on update
Invalid categorycode...	Use a valid category such as oFC (Officers) or STAFF (Staff)
Invalid branchcode...	Use a valid Koha branch / check api.defaultBranchCode
imagefile is not valid base64 data.	Use data URI format (section 5)
No patron found...	Check cardnumber
Database connection failed...	Check DB keys in .env

10. Checklist

1. Ensure `.env` exists and is filled for production
2. Set `CI_ENVIRONMENT = production`
3. Set correct `app.baseUrl` (trailing slash)
4. Set DB host / name / user / password
5. Set Basic Auth username / password
6. Confirm `api.defaultBranchCode` exists in Koha branches
7. Client uses Basic Auth + `Content-Type: application/json`
8. If URL folder name changes, update `RewriteBase` in `public/.htaccess` to match

Quick Postman test order

1. POST upsert — single patron
2. POST upsert — multiple patrons
3. GET `view?cardnumber=CQR40`
4. POST upsert — partial update
5. GET overdue

DLRL Koha Patron API — Complete User Manual

Production configuration: edit `.env` only | Auth defaults: DLRL / DLRL@2026 | Attribute codes: DESIG (Designation), DIV (DIVISION), RT (RETIRED_TRANSFERRED) |

Category codes: OFC (Officers), STAFF (Staff)