

Clash Intelligence API (1.0.0)

Download OpenAPI specification:

[Download](#)

A set of apis to compare your favourite clash royale players!

Main capabilities:

- Player ingestion from the official API
- Battle log analytics
- Deck intelligence and progression tracking
- Playstyle classification
- Player comparison engine

Typical workflow:

1. Authenticate
2. Sync cards
3. Ingest player
4. Ingest battles
5. Access analytics endpoints

Note on Authentication:

All `/api/player` and `/api/cards` routes require a JWT authentication.

1. Register → POST `/api/auth/register`
2. Login → POST `/api/auth/login`
3. Copy the returned JWT token
4. Click ****Authorize**** in Swagger
5. Enter: Bearer `<token>`

Auth

Authentication and user management

Register a new user

AUTHORIZATIONS: > *bearerAuth*

REQUEST BODY SCHEMA: `application/json`

email required	string
-------------------	--------

password required	string
----------------------	--------

Responses

- 201 User successfully registered
- 400 Validation error
- 409 User already exists
- 500 Server error

POST /api/auth/register

Request samples

Payload

Content type
application/json

Copy

```
{
  "email": "test@example.com",
  "password": "strongPassword123"
}
```

Response samples

201

Content type
application/json

Copy

```
{
  "id": 2,
  "email": "test@example.com",
  "createdAt": "2026-02-21T16:41:07.843Z"
}
```

Login and obtain JWT token

AUTHORIZATIONS: > *bearerAuth*

REQUEST BODY SCHEMA: application/json

email required	string
password required	string

Responses

- 200 Login successful
- 401 Invalid credentials
- 500 Server error

POST /api/auth/login

Request samples

Payload

Content type
application/json

Copy

```
{
  "email": "test@example.com",
```

```
"password": "strongPassword123"
}
```

Response samples

200

Content type

application/json

Copy

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Get current authenticated user

AUTHORIZATIONS: > *bearerAuth*

Responses

— **200** Authenticated user profile

— **401** Unauthorized – JWT missing or invalid

— **500** Server error

GET /api/auth/me

Response samples

200

Content type

application/json

```
{
  "id": 1,
  "email": "test@example.com",
  "createdAt": "2026-01-01T12:00:00.000Z"
}
```

Cards

Card metadata synchronisation

Synchronise Clash Royale card metadata

Fetches all card data from the Clash Royale API and stores it in the local database. This is required before ingesting players to maintain referential integrity.

AUTHORIZATIONS: > *bearerAuth*

Responses

— **200** Card metadata successfully synchronised

— **401** Unauthorized – JWT missing or invalid

— **500** Server error

POST /api/cards/sync

Response samples

200

Content type
application/json

Copy

```
{
  "status": "success",
  "totalFromAPI": 128,
  "insertedIntoDB": 128
}
```

Players

Player analytics and ingestion endpoints

Compare two players

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

tag1 required	string Example: tag1=%23PLAYER1
------------------	------------------------------------

tag2 required	string Example: tag2=%23PLAYER2
------------------	------------------------------------

Responses

- > 200 Comparison result
- > 400 Bad request - tag1 and tag2 are required
- 401 Unauthorized - JWT missing or invalid
- 404 Player not found

— 500 Server error

GET /api/player/compare

Response samples

200

400

Content type
application/json

Copy Expand all Collapse all

```
{
  "status": "success",
  "data": {
    "player1": { ... },
    "player2": { ... },
    "comparison": { ... }
  }
}
```

Get player performance overview

AUTHORIZATIONS: > bearerAuth

PATH PARAMETERS

tag	string
required	Example: %23ABC123

Responses

> 200 Player overview metrics

— 401 Unauthorized - JWT missing or invalid

— 404 Player not found

— 500 Server error

GET /api/player/{tag}/overview

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
  "status": "success",
  "data": {
    "trophies": 7500,
    "bestTrophies": 8000,
    "winRate": 0.64
  }
}
```

Get player playstyle analytics

AUTHORIZATIONS: > bearerAuth

PATH PARAMETERS

tag	string
required	Example: %23ABC123

Responses

> 200 Playstyle metrics

- > 400 Not enough battles
- 401 Unauthorized - JWT missing or invalid
- 404 Player not found
- 500 Server error

GET /api/player/{tag}/playstyle

Response samples

200

400

Content type
application/json

Copy Expand all Collapse all

```
{
  "status": "success",
  "data": {
    "aggressionScore": 2.1,
    "playstyle": "AGGR0",
    "totalBattles": 90,
    "consistencyScore": 0.5,
    "favouriteGameMode": "Ladder"
  }
}
```

Get card and deck intelligence

AUTHORIZATIONS: > bearerAuth

PATH PARAMETERS

tag	string
required	Example: %23ABC123

Responses

- > 200 Card intelligence
- > 400 No deck data available
- 401 Unauthorized - JWT missing or invalid
- 404 Player not found
- 500 Server error

GET /api/player/{tag}/cardIntelligence

Response samples

200

400

Content type
application/json

Copy Expand all Collapse all

```
{
  "status": "success",
  "data": {
    "deck": { },
    "averageElixir": 4,
    "cycleCardCount": 0,
    "rarityDistribution": { },
    "additionalProperties": { ... },
    "type": "BEATDOWN",
    "cards": [ ... ],
    "progression": { ... }
  }
}
```

Fetches player data from the external API and stores it in the local database. Existing records are updated.

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

tag	string
required	Example: <code>%2320RGGRCJJ9</code>
	Clash Royale player tag (URL encoded)

Responses

> **200** Player successfully ingested

— **401** Unauthorized – JWT missing or invalid

— **404** Player not found in external API

— **500** Server error

POST /api/player/ingest/{tag}

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{  
  "status": "success",
```

```
- "data": {
  "id": 2,
  "tag": "#20RGGRCJJ9",
  "name": "OsamaBinSaggin",
  "trophies": 5707,
  "bestTrophies": 5859,
  "wins": 757,
  "losses": 593,
  "arena": "Executioner's Kitchen",
  "createdAt": "2026-02-18T23:59:54.773Z",
  "favouriteCardName": "Witch"
}
```

Ingest player battle log

Fetches the latest battle log for a player and stores new battles in the database.

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

tag	string
required	
	Example: %2320RGGRCJJ9
	Clash Royale player tag (URL encoded)

Responses

> **200** Battles successfully ingested

— **401** Unauthorized – JWT missing or invalid

— **404** Player not found

— **500** Server error

POST /api/player/ingest/battles/{tag}

Response samples

200

Content type

application/json

Copy

```
{
  "status": "success",
  "count": 35
}
```

Get stored player profile

AUTHORIZATIONS: > *bearerAuth*

Responses

> 200 Player profile

— 401 Unauthorized - JWT missing or invalid

— 404 Player not found

— 500 Server error

GET /api/player/{tag}

Response samples

200

Content type

application/json

Copy

Expand all

Collapse all

```
{
  "status": "success",
  - "player": {
    "id": 1,
    "tag": "#ABC123",
    "name": "Arjun",
    "trophies": 7500,
    "bestTrophies": 8000,
    "wins": 700,
    "losses": 500,
    "arena": "Executioner's Kitchen",
    "createdAt": "2026-02-18T23:59:54.773Z",
    "favouriteCard": "Witch",
    + "battles": [ ... ],
    + "currentDeck": [ ... ]
  }
}
```

Health check endpoint

Checks database connectivity

AUTHORIZATIONS: > *bearerAuth*

Responses

— 200 Database is healthy

— 500 Database connection failed

GET /health

Response samples

200

Content type
text/plain

Copy

DB OK