

Verify External User

Endpoint

URL: `/:projectId/auth/verify-external-user`

Method: `POST`

Authentication Required: No

Description

Verifies a user identity using a signed JWT from an external project. If the user exists, it updates the profile. If not, it creates the user. Returns an access token, refresh token, and user data.

Request

Body Parameters

Field	Type	Required	Description
<code>userJwt</code>	<code>string</code>	Yes	Signed JWT issued by the external project.

Example Request

```
{
  "userJwt": "<EXTERNAL_USER_JWT>"
}
```

Response

Success Response (200 OK)

```
{
  "success": true,
  "accessToken": "<ACCESS_TOKEN>",
  "refreshToken": "<REFRESH_TOKEN>",
  "user": {
    "id": "user_123",
    "email": "jane@example.com",
    "username": "janedoe",
    "name": "Jane Doe",
    "avatar": "https://example.com/avatar.jpg",
    "bio": "Tech enthusiast",
    "location": {
      "type": "Point",
      "coordinates": [-73.935242, 40.73061]
    },
    "birthdate": "1995-01-01T00:00:00.000Z",
    "metadata": { "office": "boston" },
    "suspensions": [],
    "reputation": 0,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
}
```

Error Responses

Missing JWT (400 Bad Request)

```
{
  "error": "Missing userJwt",
  "code": "auth/missing-jwt"
}
```

Missing Keys (403 Forbidden)

```
{
  "error": "Missing JWT keys",
  "code": "auth/missing-keys"
}
```

Invalid Token (403 Forbidden)

```
{
  "error": "Invalid token",
  "code": "auth/invalid-token"
}
```

Project Mismatch (403 Forbidden)

```
{
  "error": "Project ID mismatch",
  "code": "auth/project-mismatch"
}
```

Unexpected Missing User (500 Internal Server Error)

```
{
  "error": "Unexpected error fetching user after login",
  "code": "auth/missing-user"
}
```

Server Error (500 Internal Server Error)

```
{
  "error": "Internal server error",
  "code": "auth/server-error",
  "details": "<Error message>"
}
```

Notes

- The JWT is verified using the current or previous public key associated with the project.

- On success, a secure HttpOnly cookie (`replyke-refresh-jwt`) is set.
- The user is updated or created based on `foreignId` and optionally `email` .
- Response includes tokens and user profile, including suspension info.

Last updated on May 11, 2025