

REST API Documentation

- **API Endpoints**
- **HTTP Response Codes**
- **Request Body Schema**
- **Response Body Schema**

```
openapi: 3.0.3
info:
  title: Barcode Scanning API
  description: |-
    Industrial-grade barcode scanning API for processing data from Honeywell CT47 and C
    This API allows for secure recording and retrieval of barcode scan data.
  version: 1.0.0
servers:
  - url: https://api.example.com/api
    description: Production server
  - url: http://localhost:3000/api
    description: Local development server
tags:
  - name: barcode
    description: Barcode scanning operations
  - name: health
    description: API health check endpoints
paths:
  /barcode:
    post:
      tags:
        - barcode
      summary: Process a single barcode scan
      description: Record a barcode scan with associated metadata
      operationId: processScan
      security:
        - BearerAuth: []
      requestBody:
        description: Barcode scan data
        required: true
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/BarcodeData"
      responses:
        "201":
          description: Scan successfully processed
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/ApiResponseSuccess"
        "400":
          description: Invalid request data
```

```

    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"
  "401":
    description: Unauthorized
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"
  "403":
    description: Forbidden – insufficient permissions
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"
  "409":
    description: Conflict – scan ID already exists
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"
  "500":
    description: Server error
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"

```

get:

tags:

– health

summary: Check API health

description: Simple health check endpoint for the barcode scanning API

operationId: checkHealth

responses:

"200":

description: API is operational

content:

application/json:

schema:

\$ref: "#/components/schemas/ApiResponseSuccess"

/barcode/batch:

post:

```
tags:
  - barcode
summary: Process multiple barcode scans
description: Batch process multiple barcode scans in a single request
operationId: processBatchScans
security:
  - BearerAuth: []
requestBody:
  description: Batch of barcode scan data
  required: true
  content:
    application/json:
      schema:
        type: object
        required:
          - scans
        properties:
          scans:
            type: array
            items:
              $ref: "#/components/schemas/BarcodeData"
responses:
  "201":
    description: All scans processed successfully
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/BatchApiResponseSuccess"
  "207":
    description: Multi-Status - Some scans processed, some failed
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/BatchApiResponsePartial"
  "400":
    description: Invalid request data
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"
  "401":
    description: Unauthorized
    content:
```

```

    application/json:
      schema:
        $ref: "#/components/schemas/ApiResponseError"
  "500":
    description: Server error
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/ApiResponseError"
/barcode/lookup:
  get:
    tags:
      - barcode
    summary: Look up a scan by ID
    description: Retrieve details of a scan by its scan_id
    operationId: lookupScan
    security:
      - BearerAuth: []
    parameters:
      - name: scan_id
        in: query
        description: Scan ID to look up
        required: true
        schema:
          type: string
    responses:
      "200":
        description: Scan found
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/ApiResponseSuccess"
      "400":
        description: Invalid request parameters
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/ApiResponseError"
      "401":
        description: Unauthorized
        content:
          application/json:
            schema:

```

```

        $ref: "#/components/schemas/ApiResponseError"
    "404":
        description: Scan not found
        content:
            application/json:
                schema:
                    $ref: "#/components/schemas/ApiResponseError"
    "500":
        description: Server error
        content:
            application/json:
                schema:
                    $ref: "#/components/schemas/ApiResponseError"
components:
    securitySchemes:
        BearerAuth:
            type: http
            scheme: bearer
            bearerFormat: API key
    schemas:
        BarcodeData:
            type: object
            required:
                - scan_id
            properties:
                scan_id:
                    type: string
                    description: The unique barcode identifier
                    example: "ABC123456789"
                scanner_id:
                    type: string
                    description: Identifier for the scanner device
                    example: "HONEYWELL-CT47-001"
                scan_timestamp:
                    type: string
                    format: date-time
                    description: When the scan occurred
                    example: "2025-04-09T12:00:00Z"
                scan_location:
                    type: string
                    description: Where the scan occurred
                    example: "Warehouse-A"
                scan_type:

```

type: string
description: Type of scan
example: "inventory_check"
metadata:
 type: object
 description: Additional data as JSON
 example:
 operator: "John Doe"
 batch: "B12345"

StoredScanData:

allOf:
 - \$ref: "#/components/schemas/BarcodeData"
 - type: object
 required:
 - id
 - created_at
 properties:
 id:
 type: integer
 description: Database record ID
 example: 123
 created_at:
 type: string
 format: date-time
 description: When the record was created
 example: "2025-04-09T12:00:01Z"

ApiResponseSuccess:

type: object
required:
 - success
 - message
properties:
 success:
 type: boolean
 description: Whether the operation was successful
 example: true
 message:
 type: string
 description: Success message
 example: "Barcode scan processed successfully"
 data:
 type: object
 description: Response data

ApiResponseError:

type: object

required:

- success
- message

properties:

success:

type: boolean

description: Whether the operation was successful

example: `false`

message:

type: string

description: Error message

example: `"Invalid request data"`

error:

type: string

description: Error type

enum:

- VALIDATION_ERROR
- AUTHORIZATION_ERROR
- DUPLICATE_SCAN
- DATABASE_ERROR
- SERVER_ERROR

example: `"VALIDATION_ERROR"`

BatchApiResponseSuccess:

type: object

required:

- success
- message
- data

properties:

success:

type: boolean

description: Whether the operation was successful

example: `true`

message:

type: string

description: Success message

example: `"Processed 10 of 10 scans"`

data:

type: object

required:

- processed

- failed
- totalProcessed
- totalFailed
- totalSubmitted

properties:

processed:

type: array

description: Successfully processed scans

items:

\$ref: ["#/components/schemas/StoredScanData"](#)

failed:

type: array

description: Failed scans

items:

type: object

required:

- data
- error

properties:

data:

\$ref: ["#/components/schemas/BarcodeData"](#)

error:

type: string

description: Error message

example: ["Duplicate scan_id: ABC123456789"](#)

totalProcessed:

type: integer

description: Number of successfully processed scans

example: [10](#)

totalFailed:

type: integer

description: Number of failed scans

example: [0](#)

totalSubmitted:

type: integer

description: Total number of scans submitted

example: [10](#)

BatchApiResponsePartial:

allOf:

- \$ref: ["#/components/schemas/BatchApiResponseSuccess"](#)

- type: object

properties:

success:

```
    example: true
message:
  example: "Processed 8 of 10 scans"
data:
  properties:
    totalProcessed:
      example: 8
    totalFailed:
      example: 2
    totalSubmitted:
      example: 10
```