Capstone Project Part-1 DESIGN THE REFUNDS API SPECIFICATION (OpenAPI 3.0)

This section includes:

OpenAPI 3.0 specification

Versioning (/api/v1/refunds)

Pagination (page, size)

Standard error response model

Examples for all endpoints

Paths for:

- /refunds
- /refunds/{id}
- /refunds/status

---

1. Full OpenAPI 3.0 Specification (Swagger YAML)

```
openapi: 3.0.3
info:
  title: Refunds API
  description: API for creating, retrieving and checking status of refunds.
  version: "1.0.0"


servers:
  - url: http://localhost:8080/api/v1
    description: Local Development Server
```

```
paths:

  /refunds:
    get:
      summary: Get all refunds with pagination
      tags:
        - Refunds
      parameters:
        - name: page
          in: query
          description: Page number (starting from 0)
          required: false
          schema:
            type: integer
            default: 0
        - name: size
          in: query
          description: Number of items per page
          required: false
          schema:
            type: integer
            default: 10
      responses:
        '200':
          description: List of refunds
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Refund'
              example:
```

```yaml
            - id: "f32c1e33"
              amount: 200
              currency: "USD"
              reason: "Order Cancellation"
              status: "PENDING"

post:
  summary: Create a refund request
  tags:
    - Refunds
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/RefundRequest'
        example:
          amount: 350
          currency: "USD"
          reason: "Product Damaged"
  responses:
    '201':
      description: Refund created
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Refund'
    '400':
      description: Invalid request
      content:
        application/json:
          schema:
              $ref: '#/components/schemas/ErrorResponse'

/refunds/{id}:
  get:
    summary: Get refund by ID
    tags:
      - Refunds
    parameters:
      - name: id
        in: path
        required: true
        description: Refund ID
        schema:
          type: string
    responses:
      '200':
        description: Refund details
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Refund'
      '404':
        description: Refund not found
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'

/refunds/status:
  post:
    summary: Get status of a refund
    tags:
```

```
    - Refunds
   requestBody:
    required: true
    content:
     application/json:
      schema:
       $ref: '#/components/schemas/RefundStatusRequest'
      example:
       refundId: "abc123"
   responses:
    '200':
     description: Refund status
     content:
      application/json:
       schema:
        $ref: '#/components/schemas/RefundStatusResponse'
    '404':
     description: Refund not found
     content:
      application/json:
       schema:
        $ref: '#/components/schemas/ErrorResponse'


components:

 schemas:

  Refund:
   type: object
   properties:
    id:
      type: string
    amount:
      type: number
    currency:
      type: string
    reason:
      type: string
    status:
      type: string
      enum: [PENDING, APPROVED, REJECTED]

RefundRequest:
  type: object
  required: [amount, currency, reason]
  properties:
    amount:
      type: number
    currency:
      type: string
    reason:
      type: string

RefundStatusRequest:
  type: object
  properties:
    refundId:
      type: string

RefundStatusResponse:
  type: object
  properties:
    refundId:
      type: string
    status:
      type: string

ErrorResponse:
  type: object
  properties:
    timestamp:
      type: string
    path:
      type: string
    message:
      type: string
    code:
      type: integer
```

2. Versioning Applied

Base URL includes version:

```
/api/v1/refunds
```

This is correct REST API version practice.

3. Pagination Applied

GET /refunds supports:

```
?page=0&size=10
```

This allows:

- scalable responses
- frontend compatibility

- supports infinite scroll

---

4. Error Handling Applied

All error responses follow:

```
ErrorResponse:
  type: object
  properties:
    timestamp: string
    path: string
    message: string
    code: integer
```

Matches proper API design.

---

5. Swagger Usage (Step-by-Step)

Step 1 — Open https://editor.swagger.io

Step 2 — Paste the entire YAML

Step 3 — Validate the schema

Step 4 — Export as:

- JSON
- YAML
- Server Stub (Spring Boot)
- Client SDK

Step 5 — Save file as:

refunds-api.yaml