

# מדריך חזותי מלא - Server Endpoints

של השירות עם דיאגרמות חזותיות מפורטות endpoints-מדריך זה מציג את כל ה  של Request/Response כולל: זרימות נתוניות, תנאים, שגיאות, ודוגמאות endpoint

## קישורים מהירים

- בדיקה שהשירות עובד - [Health Check](#)
- [Authentication](#): [Auth Endpoints](#) הרשמה והתחברות
- [Cart System](#) - עגלת קניות
- [Products](#) - מוצרים
- [Orders](#) - ניהול הזמנות
- [Best Practices](#): [למטה ↓](#) - המלצות ודוגמאות

## איך להשתמש במדריך?

### לפי תפקיד:

#### Frontend Developer:

1. Request/Response examples - ראה את ה
2. Error tables (מה לטפל במקרה) - שים לב ל
3. Authentication requirements - בדיקת

#### Backend Developer:

1. accountable for the application - stage responsible for
2. Side Effects (stock, cart, emails) - שים לב ל
3. MongoDB vs Redis - בבין הבדל בין

#### QA/Tester:

1. Error tables - השמש ב
2. test cases (expected vs actual) - בפיה
3. Check all requirements in the specification - בדוק את כל התנאים

#### מתכנת מתחילה:

## 1. התחל ב-Common Workflows

קרא את הדיאגרמות משמאל לימין.

עקבות אחרי הצבעים (כחול → ירוק=הצלחה)



## מקרה צבעים מהיר

צבע	משמעות	דוגמה
כחול	- נקודת כניסה Request	POST /api/auth/login
ירוק	- Response 200/201 הצלחה	200: Success
אדום	- Errors 400/401/404/409 שגיאה	401: Unauthorized
צהוב	MongoDB מסד נתונים ראשי -	Find user in MongoDB
כתום	Redis - Cache מהיר	Get cart from Redis

## Authentication Endpoints

**מטרה:** ניהול משתמשים - הרשמה, התחברות, ניהול פרופיל

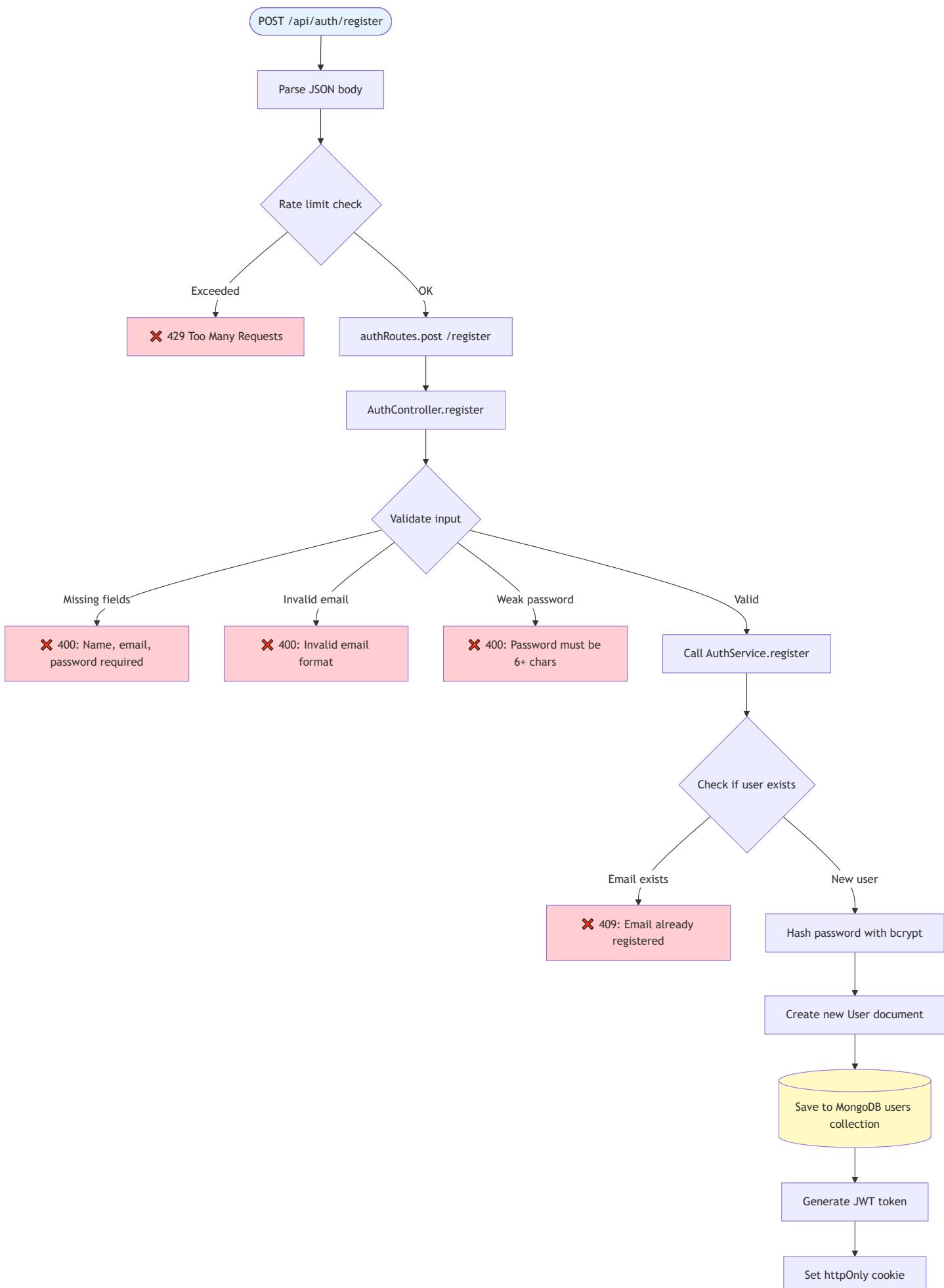
**Authentication:** JWT Token-ב-`httpOnly cookie`

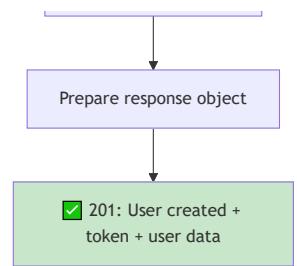
**Rate Limiting:** מוגבל ל-5 ניסיונות לדקה

## POST /api/auth/register

**תיאור:** ייצור משתמש חדש במערכת

**Security:** Password-`bcrypt`-מוגצפן ב-`httpOnly cookie`





## 📍 Request Example:

```
{
  "name": "John Doe",
  "email": "john@example.com",
  "password": "securepass123"
}
```

## ✓ Success Response (201):

```
{
  "status": "success",
  "data": {
    "user": {
      "_id": "507f1f77bcf86cd799439011",
      "name": "John Doe",
      "email": "john@example.com"
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  }
}
```

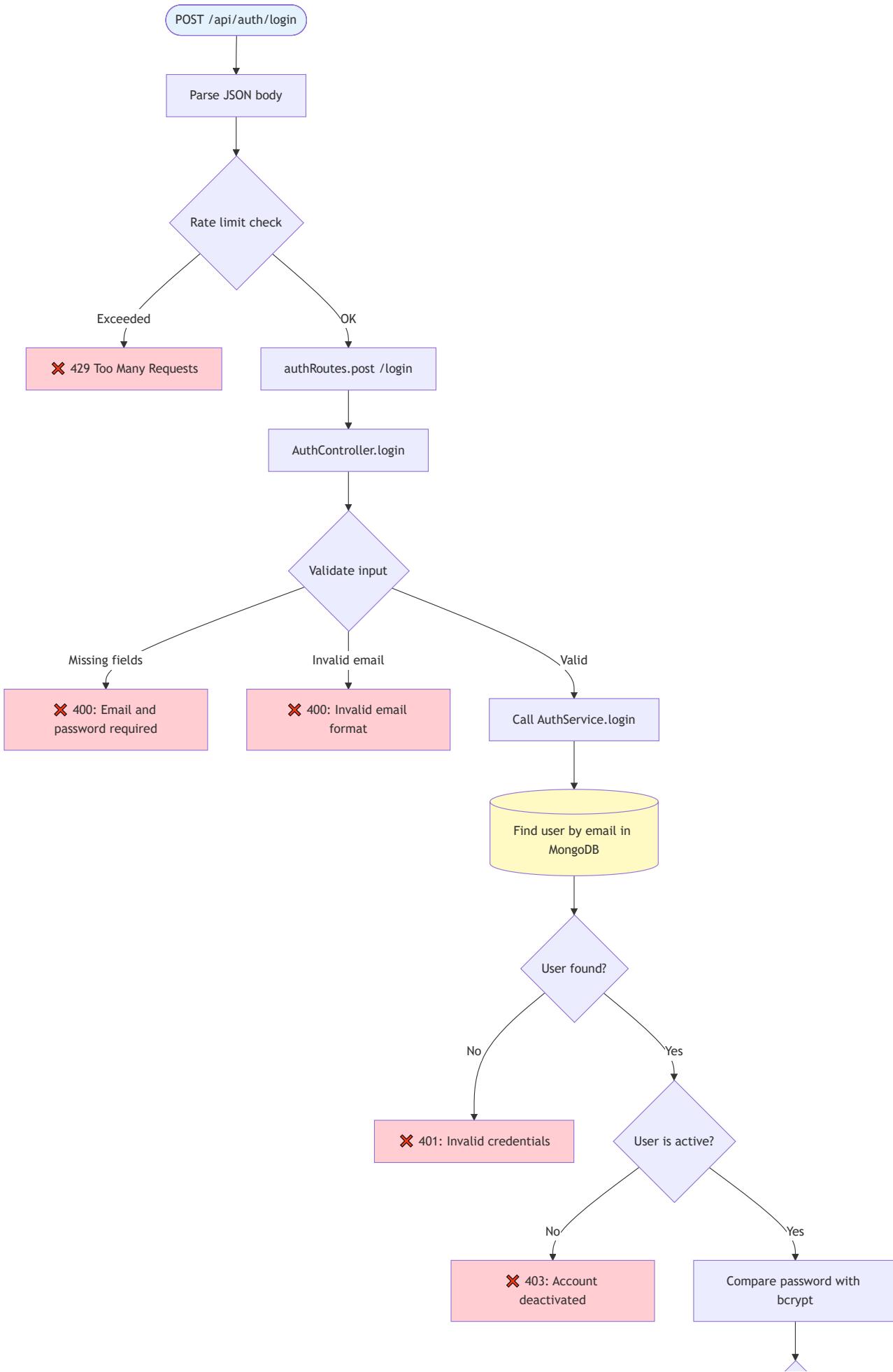
## ✗ Possible Errors:

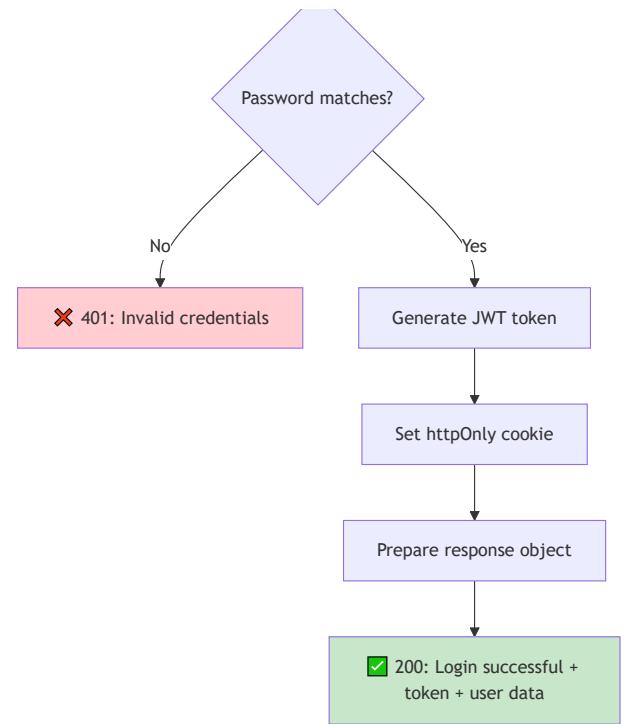
Status	Message	Cause
400	Name, email, password required	חסרים שדות חובה
400	Invalid email format	לא תקין email פורמט
400	Password must be 6+ chars	סיסמה קצרה מדי
409	Email already registered	כבר קיימ במערכת Email
429	Too Many Requests	יותר מדי ניסיונות

## POST /api/auth/login

 **תיאור:** התחברות למערכת עם 이메יל וסיסמה

 **Security:** bcrypt password comparison, JWT token generation





## 🌐 Request Example:

```
{
  "email": "john@example.com",
  "password": "securepass123"
}
```

## ✓ Success Response (200):

```
{
  "status": "success",
  "data": {
    "user": {
      "_id": "507f1f77bcf86cd799439011",
      "name": "John Doe",
      "email": "john@example.com"
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  }
}
```

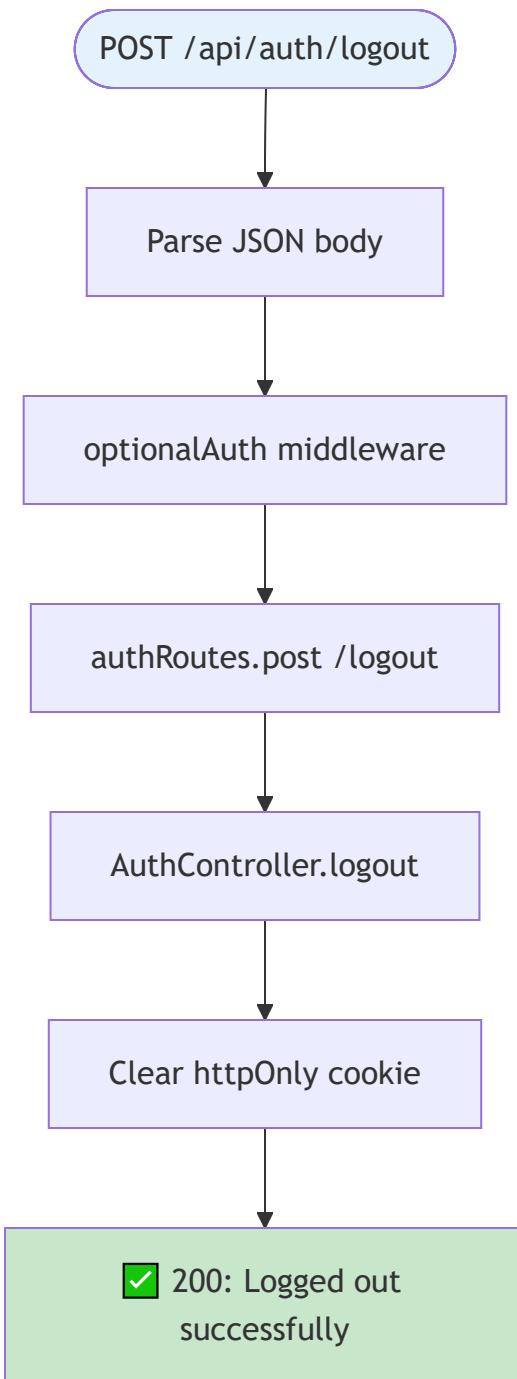
## ✗ Possible Errors:

Status	Message	Cause
400	Email and password required	חסרים שדות חובה
400	Invalid email format	לא תקין 형식 email
401	Invalid credentials	או סיסמה שגויים Email
403	Account deactivated	החשבון מושבת
429	Too Many Requests	יותר מדי ניסיונותהתחברות

## POST /api/auth/logout

 **תיאור:** התנתקות מהמערכת - מחיקת session cookie

 **Security:** מהדפן httpOnly cookie מחיקת



#### ✓ Success Response (200):

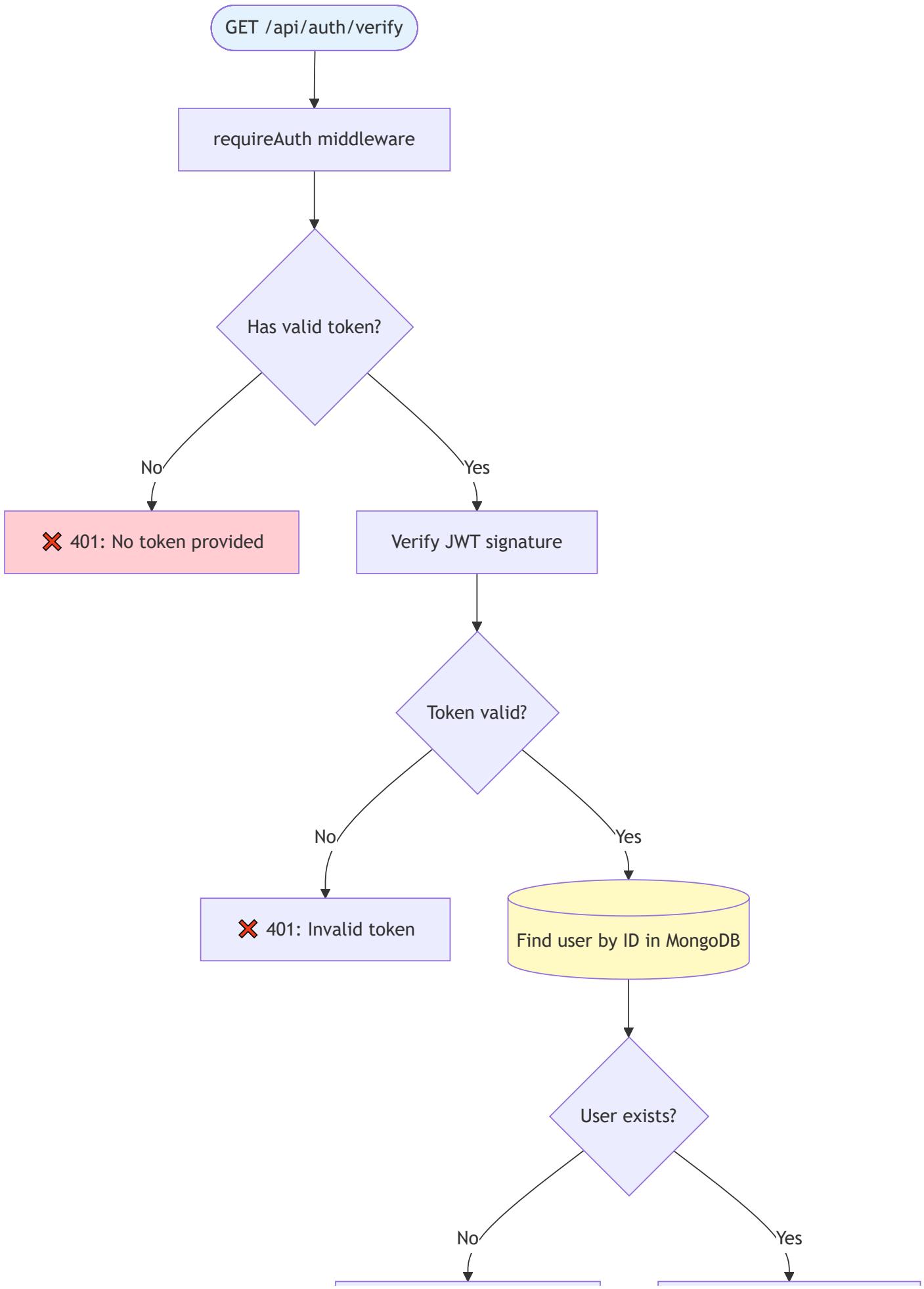
```
{  
  "status": "success",  
  "message": "Logged out successfully"  
}
```

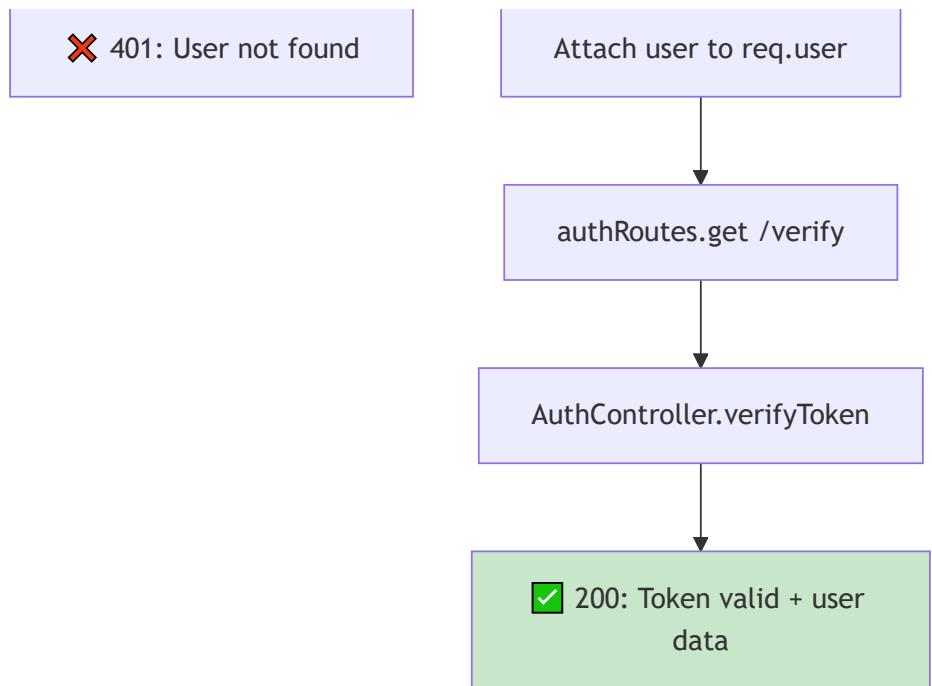
💡 **Note:** ה-logout עובד גם עבור משתמשים לא מחוברים (optionalAuth)

## GET /api/auth/verify

 מודא שהמשתמש מחובר - Token **טיור:** בדיקת תקינות

 **Security:** תקף JWT token מחייב





### ✓ Success Response (200):

```
{
  "status": "success",
  "data": {
    "user": {
      "_id": "507f1f77bcf86cd799439011",
      "name": "John Doe",
      "email": "john@example.com"
    }
  }
}
```

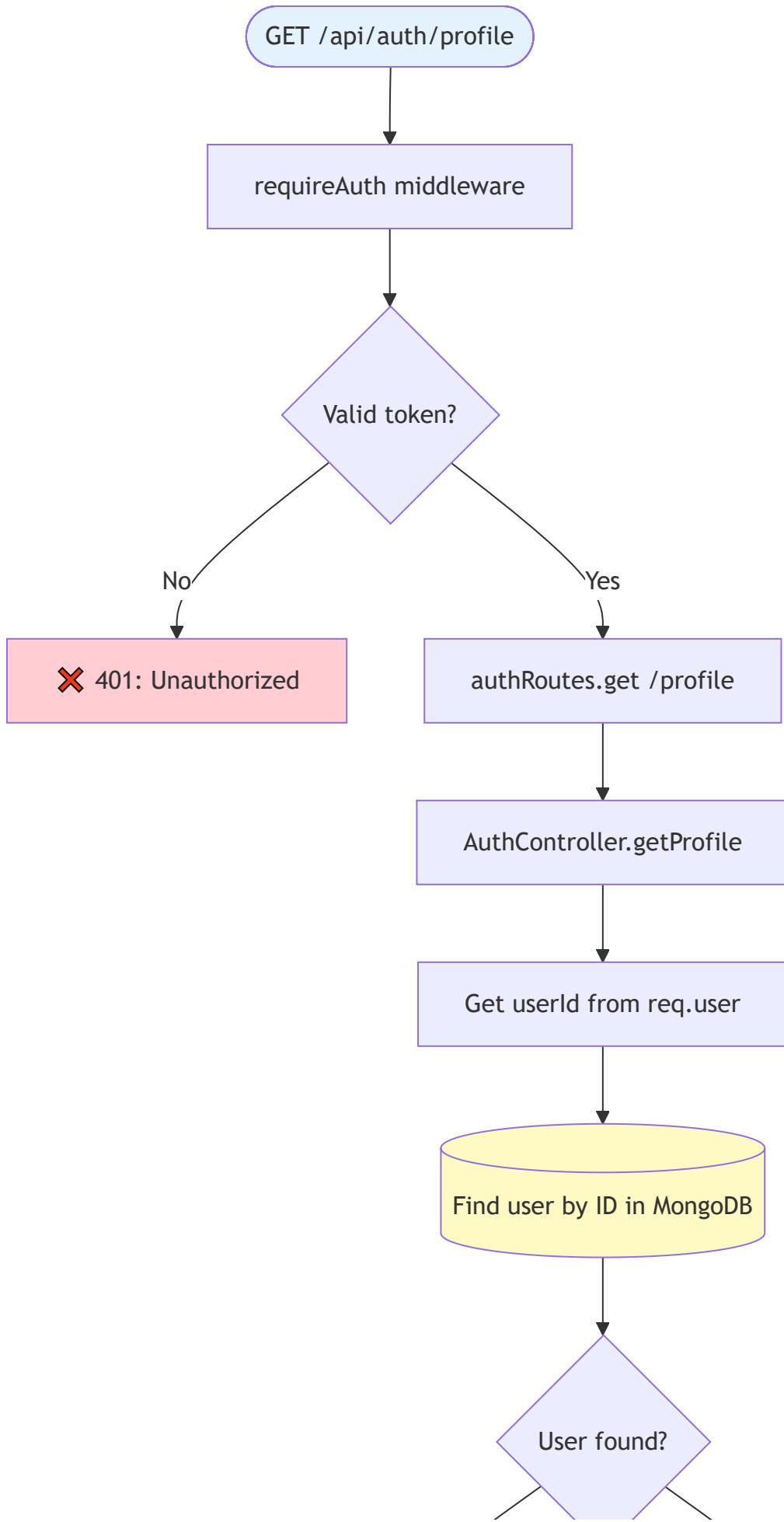
### ✗ Possible Errors:

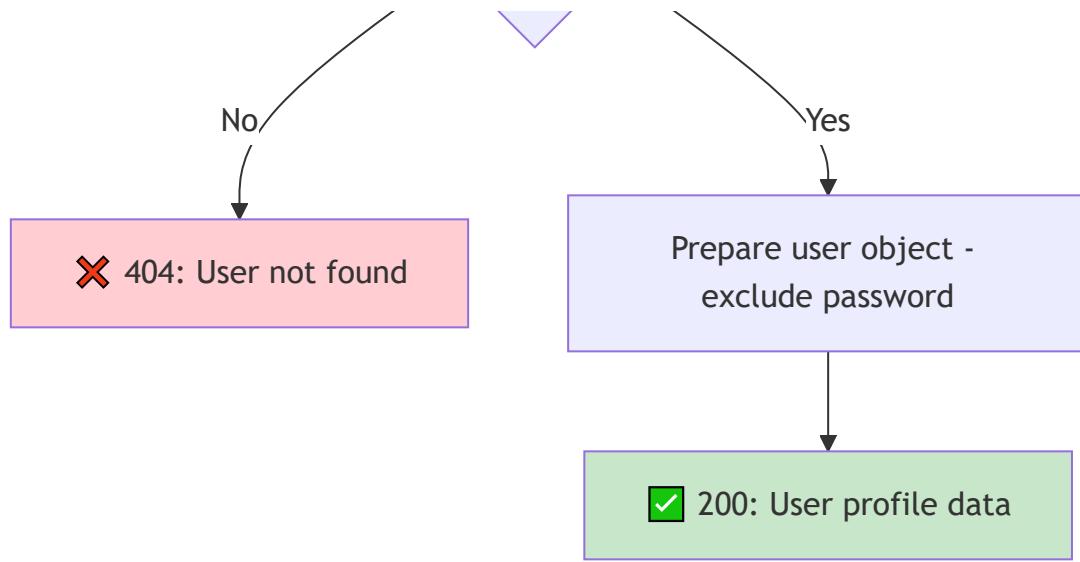
Status	Message	Cause
401	No token provided	בקשה אינטלקטואלית בבקשה cookie/token אין
401	Invalid token	תוקן לא תקין או פג תוקן
401	User not found	המשתמש נמחק מהמערכת

## GET /api/auth/profile

 **תיאור:** קבלת פרטי המשתמש המחבר

 **Security:** מחייב התחברות (requireAuth)





### ✓ Success Response (200):

```
{
  "status": "success",
  "data": {
    "user": {
      "_id": "507f1f77bcf86cd799439011",
      "name": "John Doe",
      "email": "john@example.com",
      "createdAt": "2024-01-15T10:30:00.000Z"
    }
  }
}
```

### ✗ Possible Errors:

Status	Message	Cause
401	Unauthorized	לא מחובר
404	User not found	המשתמש לא נמצא

## 🛒 Cart Endpoints

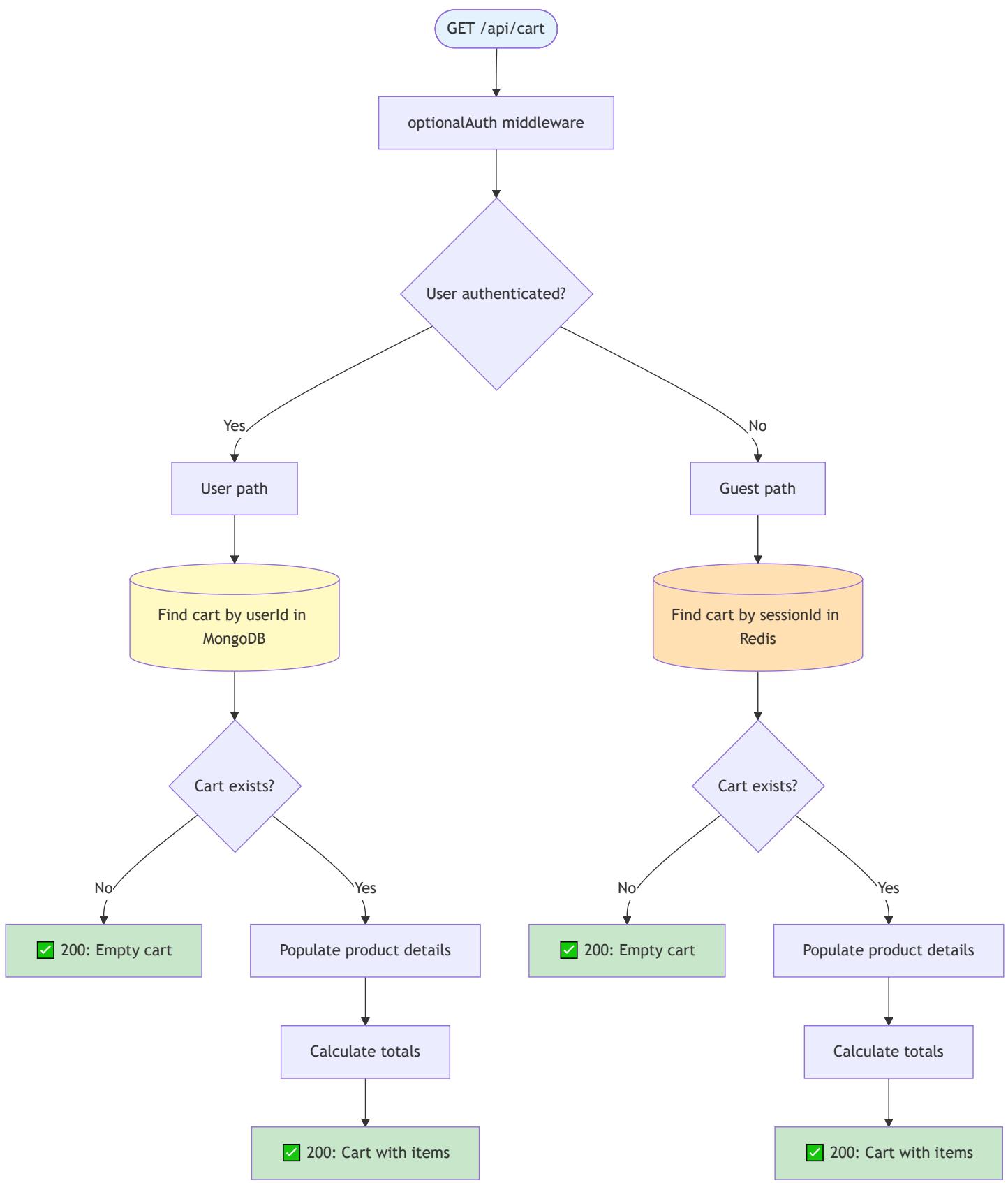
**מטרה:** ניהול עגלת קניות - תמיכה במשתמשים מחוברים ו אורחים

**Storage:**

- קבוע) → משתמשים מחוברים
- Redis (זמן, 24 שעות) → משתמשים אורחים

## GET /api/cart

 **תיאור:** קבלת עגלת הקניות - עובד גם למשתמשים מחוברים וגם לאורחים

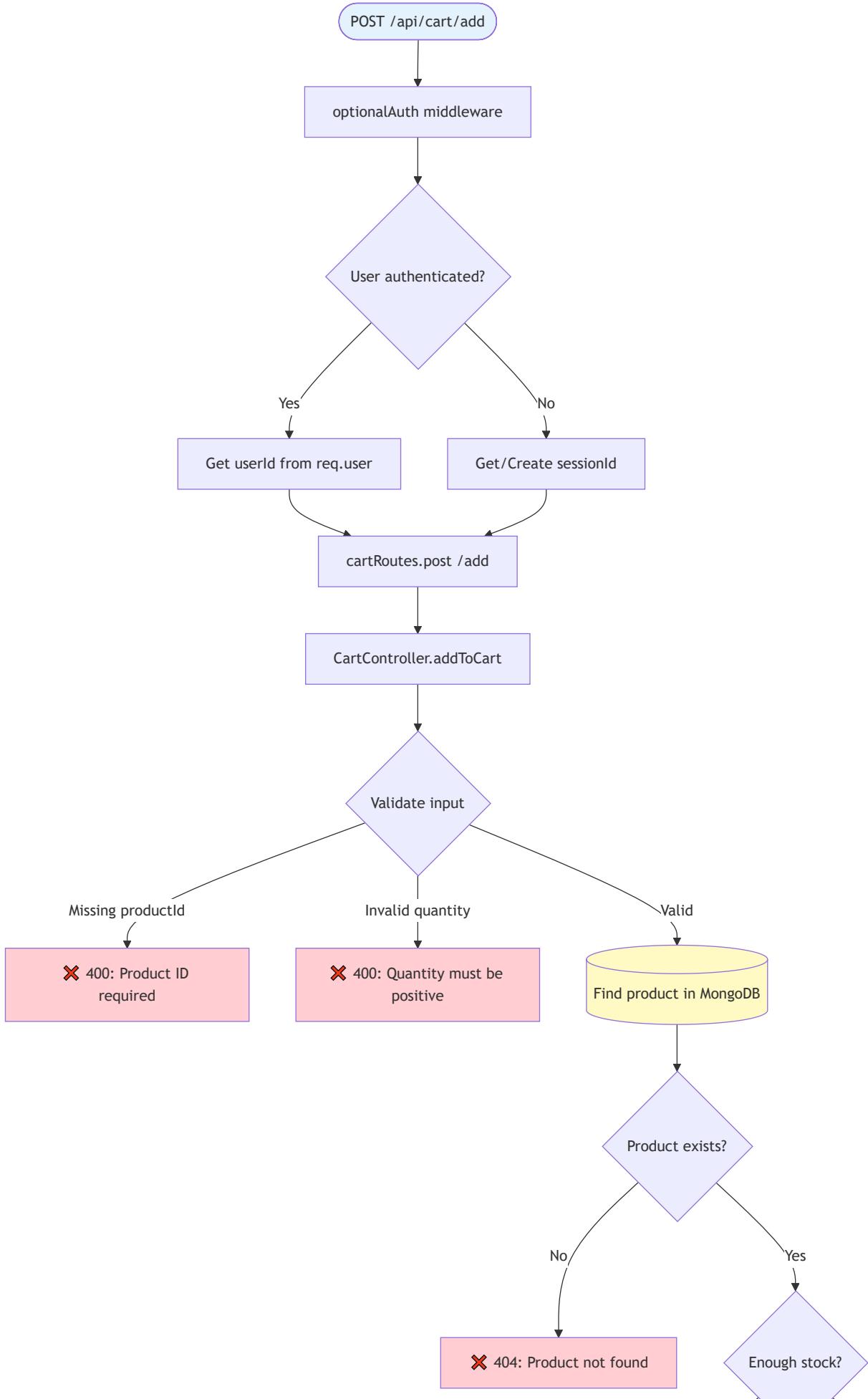


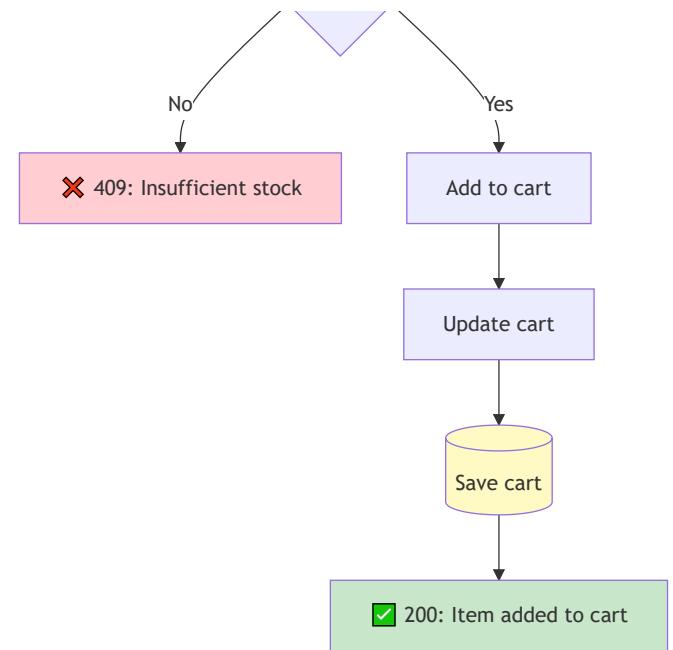
**Success Response (200):**

```
{  
  "status": "success",  
  "data": {  
    "cart": {  
      "items": [  
        {  
          "productId": "507f1f77bcf86cd799439011",  
          "name": "Product Name",  
          "price": 29.99,  
          "quantity": 2,  
          "subtotal": 59.98  
        }  
      ],  
      "totalItems": 2,  
      "totalPrice": 59.98  
    }  
  }  
}
```

## POST /api/cart/add

 **תיאור:** הוספה מוצר לעגלת קניות





## Request Example:

```
{
  "productId": "507f1f77bcf86cd799439011",
  "quantity": 2
}
```

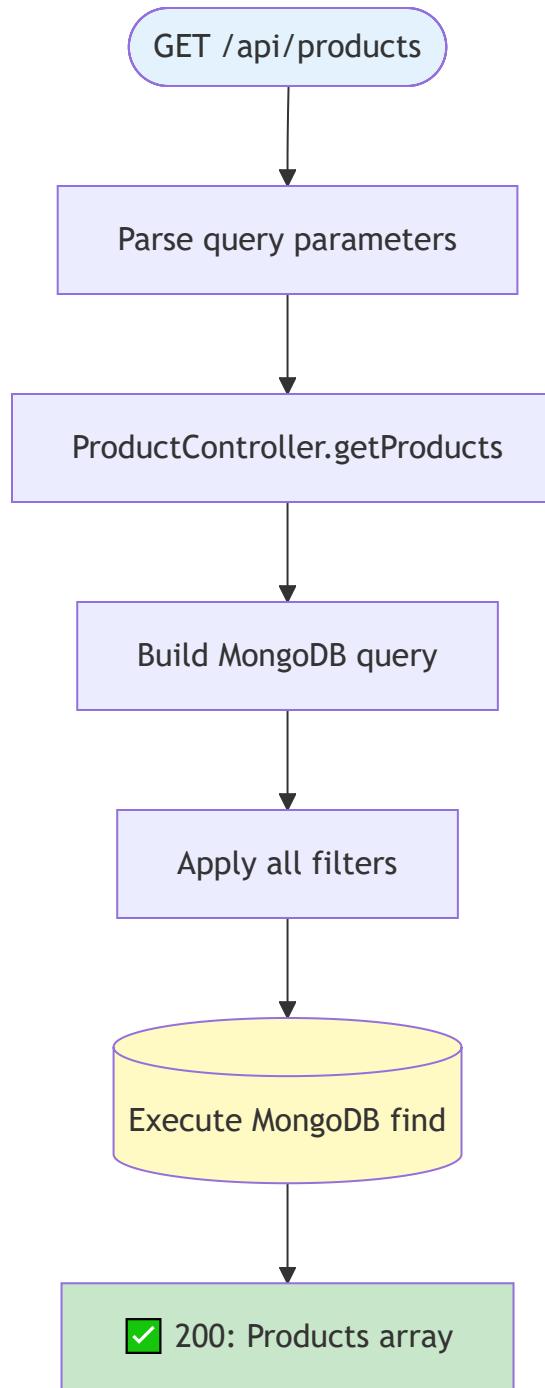
## Success Response (200):

```
{
  "status": "success",
  "data": {
    "cart": {
      "items": [
        {
          "productId": "507f1f77bcf86cd799439011",
          "quantity": 2,
          "price": 29.99
        }
      ],
      "totalItems": 2,
      "totalPrice": 59.98
    }
  }
}
```

# Product Endpoints

## GET /api/products

 **תיאור:** קבלת רשימה מוצרים עם אפשרות סינון וחיפוש

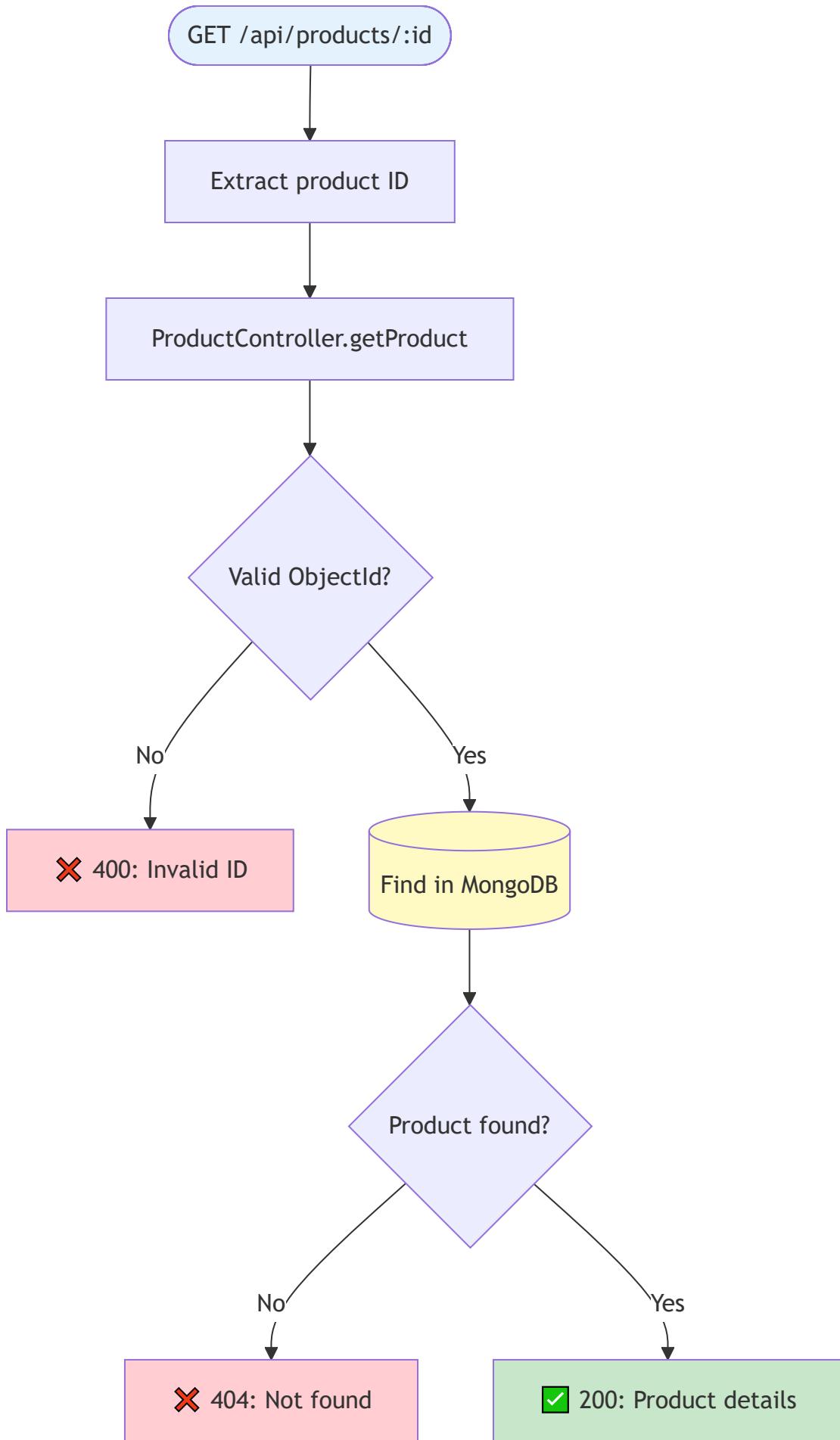


 **Success Response (200):**

```
{  
  "status": "success",  
  "data": {  
    "products": [  
      {  
        "_id": "507f1f77bcf86cd799439011",  
        "name": "Product Name",  
        "description": "Product description",  
        "price": 29.99,  
        "stock": 100,  
        "category": "electronics"  
      }  
    ]  
  }  
}
```

## GET /api/products/:id

 **תיאור:** קבלת פרט אחד בלבד



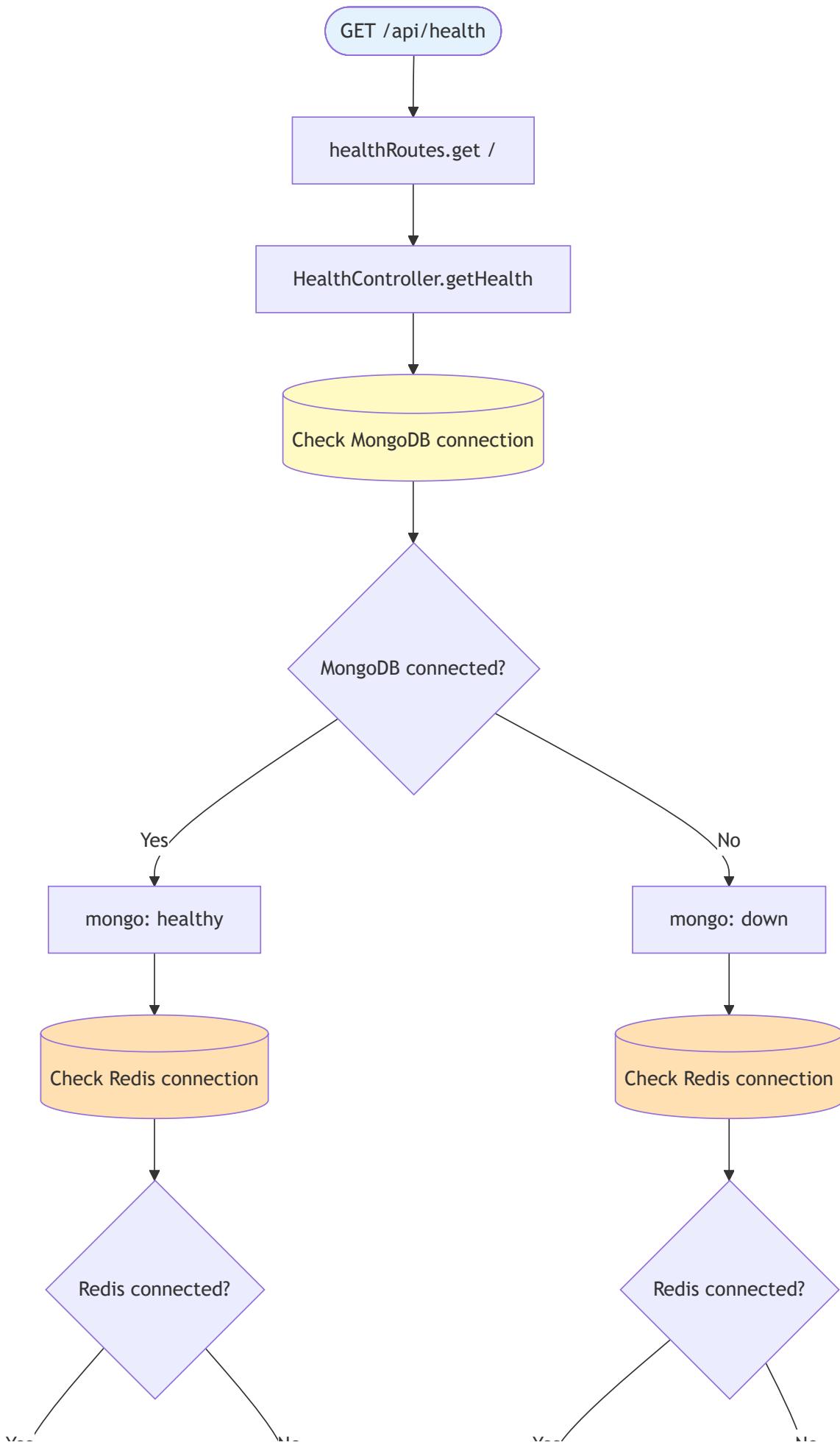
## Success Response (200):

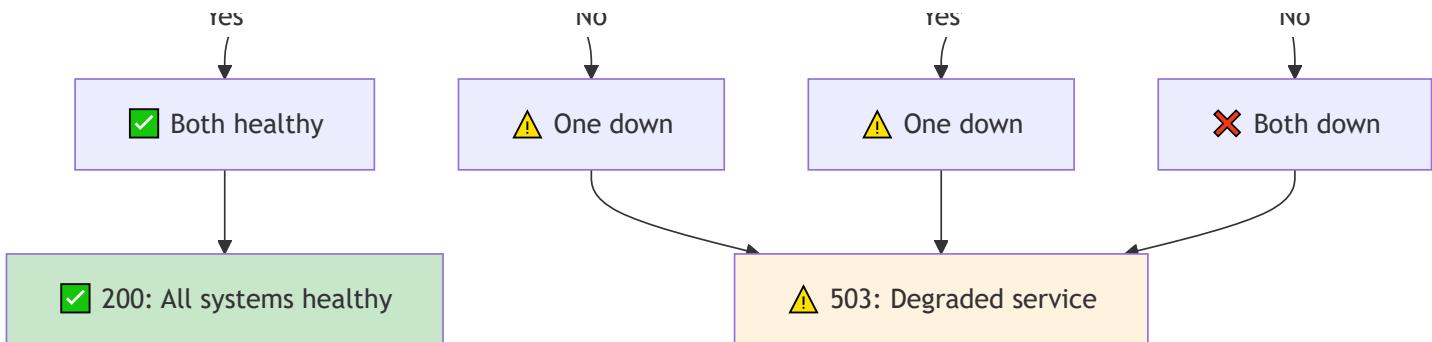
```
{  
  "status": "success",  
  "data": {  
    "product": {  
      "_id": "507f1f77bcf86cd799439011",  
      "name": "Product Name",  
      "description": "Detailed product description",  
      "price": 29.99,  
      "stock": 100,  
      "category": "electronics",  
      "imageUrl": "https://example.com/image.jpg"  
    }  
  }  
}
```

## Health Endpoints

### GET /api/health

 זמן הפעולה - **תיאור:** בדיקת סטטוס השרת MongoDB, Redis



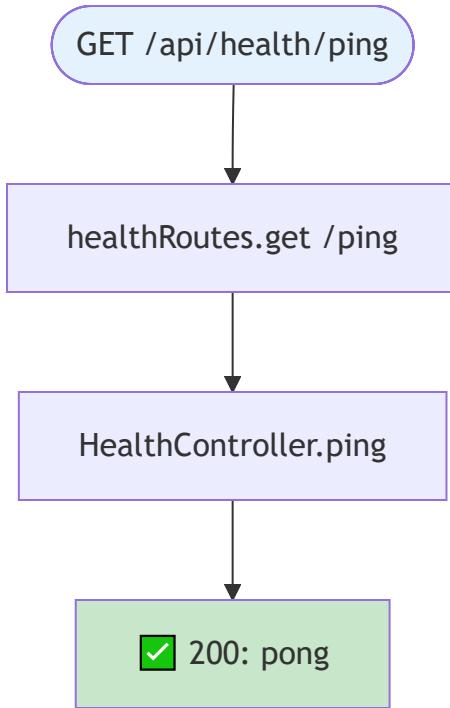


### ✓ Success Response (200):

```
{
  "status": "success",
  "data": {
    "health": {
      "status": "healthy",
      "mongodb": "connected",
      "redis": "connected",
      "uptime": 12345,
      "timestamp": "2024-01-15T10:30:00.000Z"
    }
  }
}
```

## GET /api/health/ping

**תיאור:** בדיקה פשוטה - האם השירותעובד



### ✓ Success Response (200):

```
{
  "status": "success",
  "message": "pong"
}
```

## 💡 Best Practices

### 🔒 Security

1. **Always verify JWT** - requireAuth כל הפעולות הרגישות דורשות - מהיר פי 10 מיליאון פעולות
2. **Rate Limiting** - Auth endpoints מוגבלים ל-5 ניסיונות/דקה
3. **httpOnly Cookies** - Tokens-JavaScript לא נגישים ל-JavaScript
4. **Password Hashing** - bcrypt עם salt
5. **Input Validation** - לפני שימוש בעוברים validation-כל ה-

### ⚡ Performance

1. **Redis for Guest Carts** - מהיר פי 10 MongoDB
2. **Debounced MongoDB Saves** - Cart saves מתבצעים כל 5 שניות

3. **Product Population** - Lazy loading של פרטיו מוצרים
4. **Index fields על** - **חשיבותם** email, userId, sessionId

## Error Handling

1. **Specific Error Messages** - כל שגיאה עם הסבר ברור -
2. **HTTP Status Codes** - שימוש-status codes
3. **Validation Errors** - 400 עם פירוט השודות החסרים
4. **Not Found** - למשאים שלא קיימים 404
5. **Unauthorized** - 401 כשיין token, כשיין הרשאה

### Perfect for:

-  מדיקת endpoints הבנת התנהגות
-  א-יתור באגים ב-API
-  למפתחים API תיעוד
-  כתיבת טסטים
-  לצוות חדש Onboarding