



Server Endpoints - Complete Visual Map



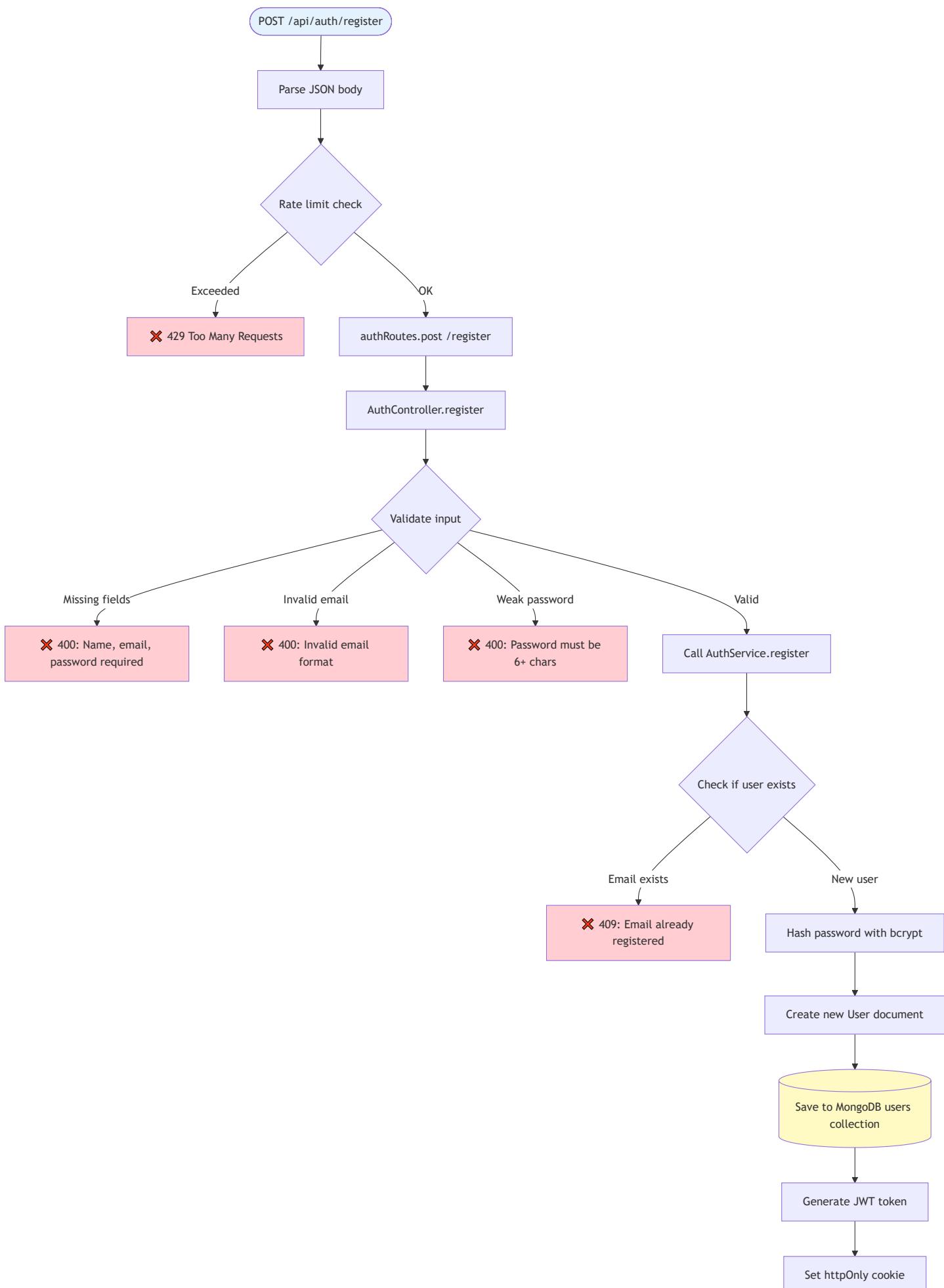
Table of Contents

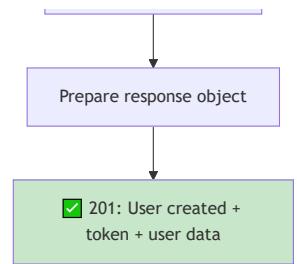
-  Authentication Endpoints
 - POST /api/auth/register
 - POST /api/auth/login
 - POST /api/auth/logout
 - GET /api/auth/verify
 - GET /api/auth/profile
 - PUT /api/auth/profile
 - PUT /api/auth/password
 - DELETE /api/auth/account
 - GET /api/auth/stats
-  Cart Endpoints
 - GET /api/cart
 - POST /api/cart/add
 - PUT /api/cart/update
 - DELETE /api/cart/remove
 - DELETE /api/cart/clear
 - GET /api/cart/count
 - POST /api/cart/merge
-  Product Endpoints
 - GET /api/products
 - GET /api/products/:id
-  Order Endpoints
 - POST /api/orders
 - GET /api/orders
 - GET /api/orders/:id
 - POST /api/orders/:id/cancel
 - PUT /api/orders/:id/status
 - GET /api/orders/stats
-  Health Endpoints

- [GET /api/health](#)
- [GET /api/health/ping](#)

Authentication Endpoints

POST /api/auth/register





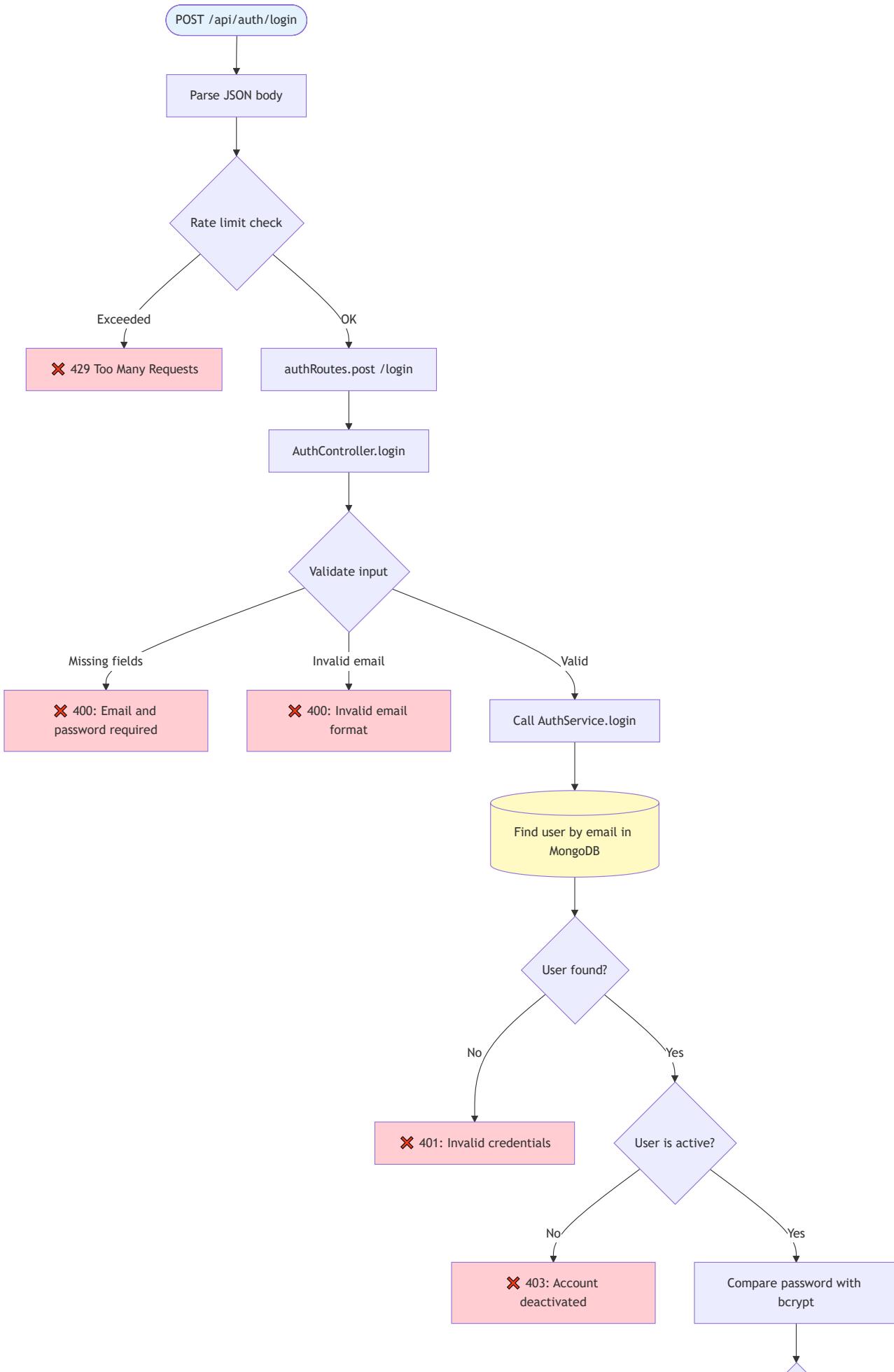
Request:

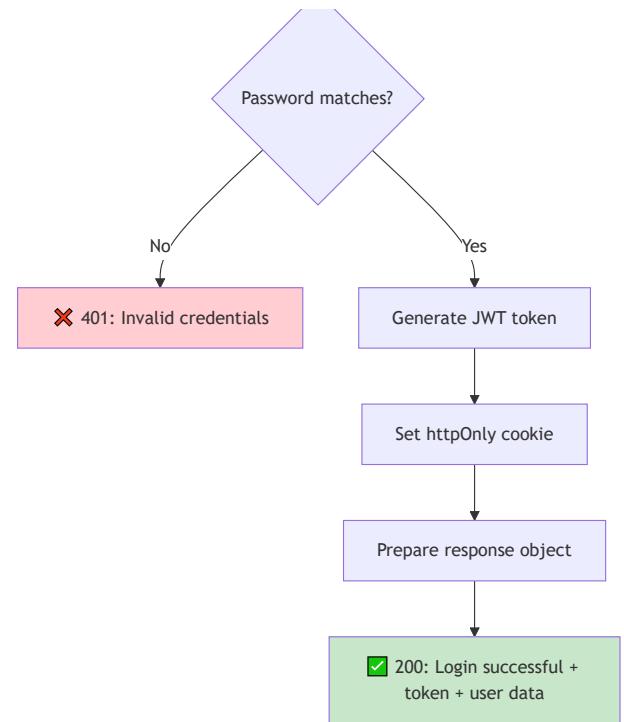
```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "password": "securepass123"  
}
```

Response (201):

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

POST /api/auth/login





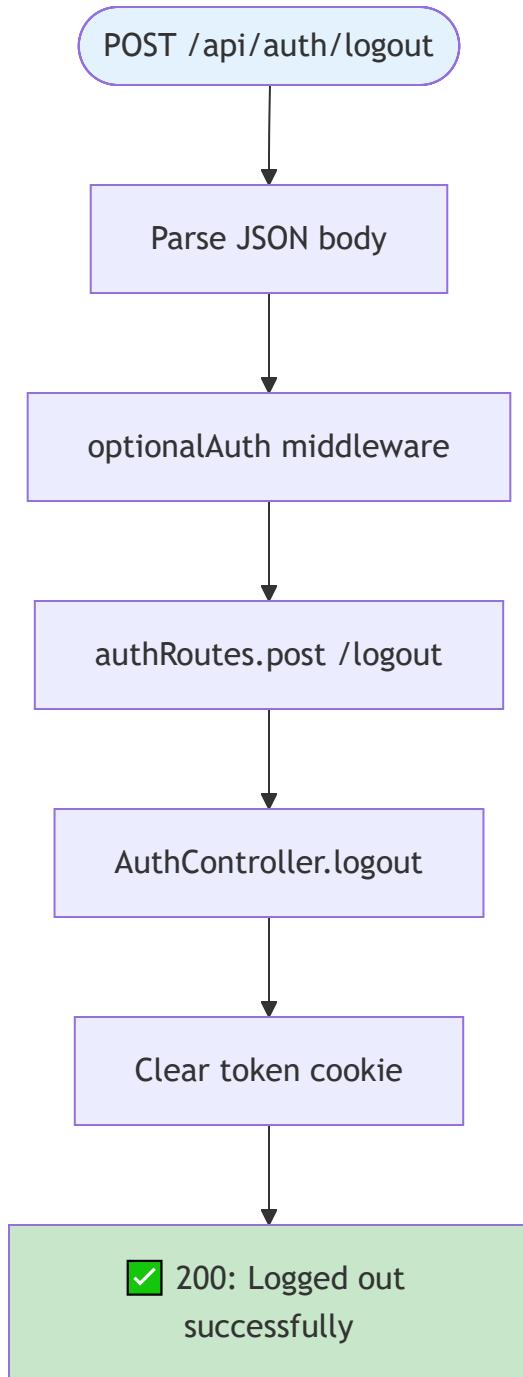
Request:

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

Response (200):

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

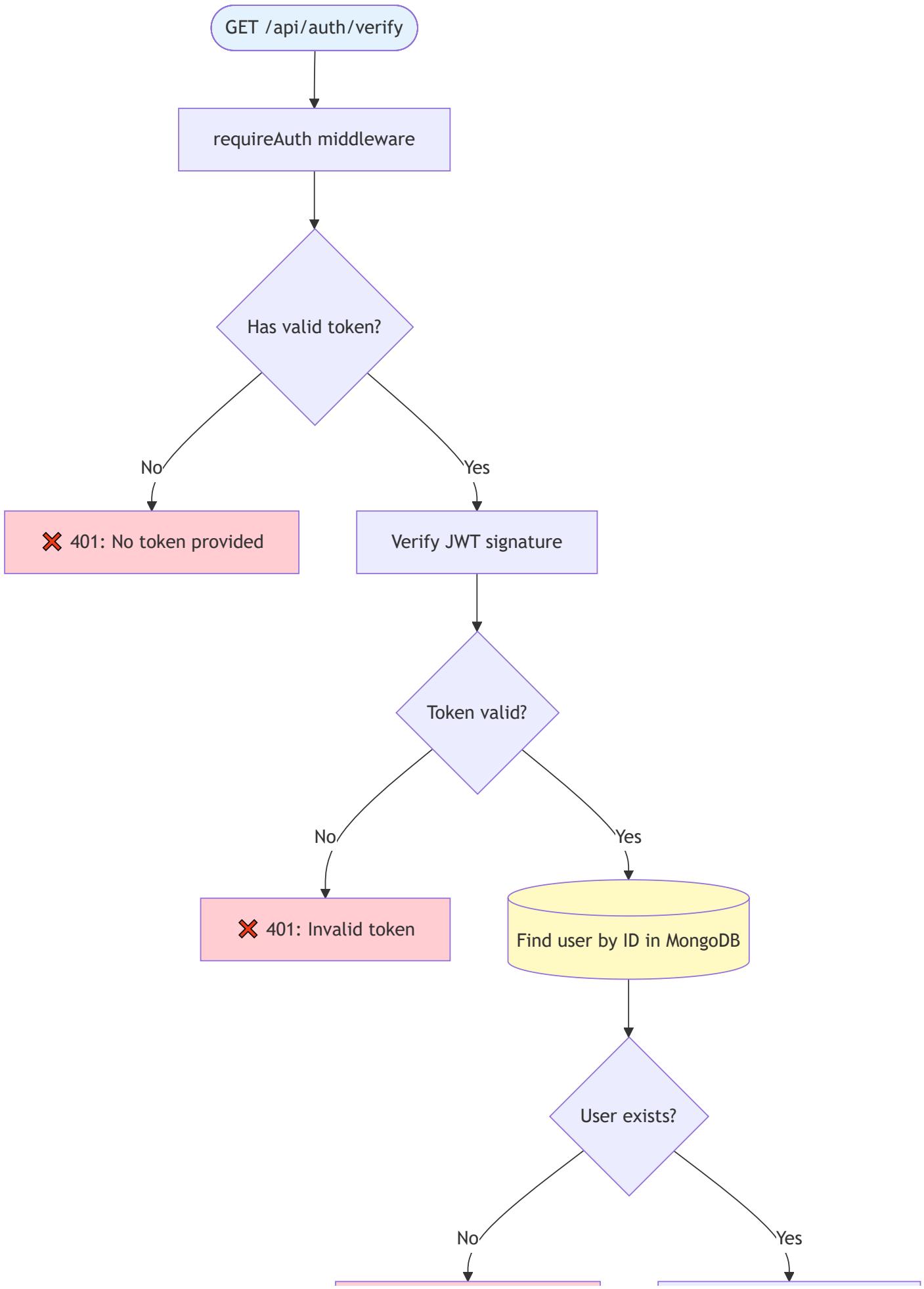
POST /api/auth/logout

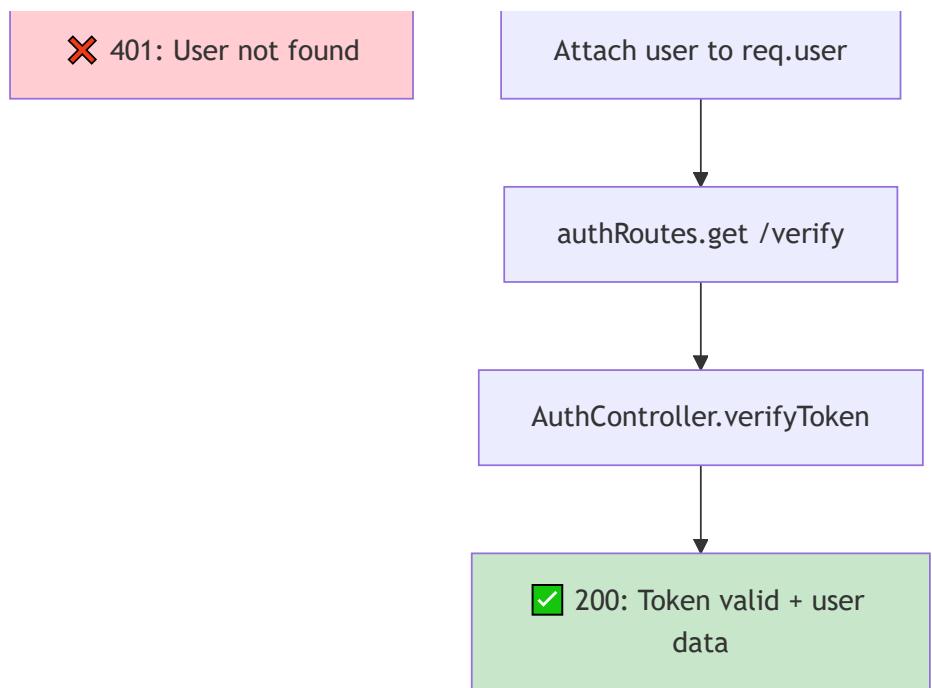


Response (200):

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

GET /api/auth/verify

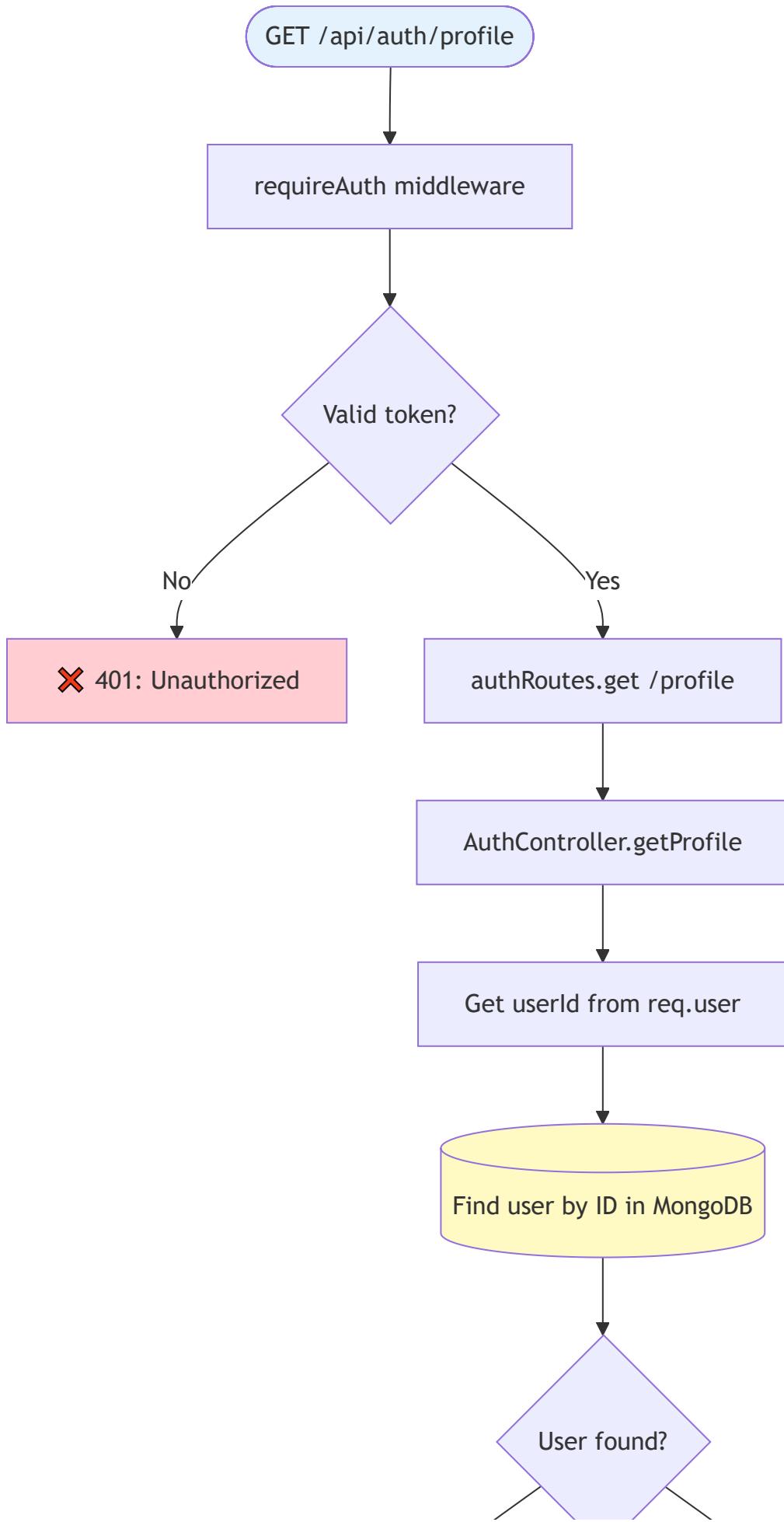


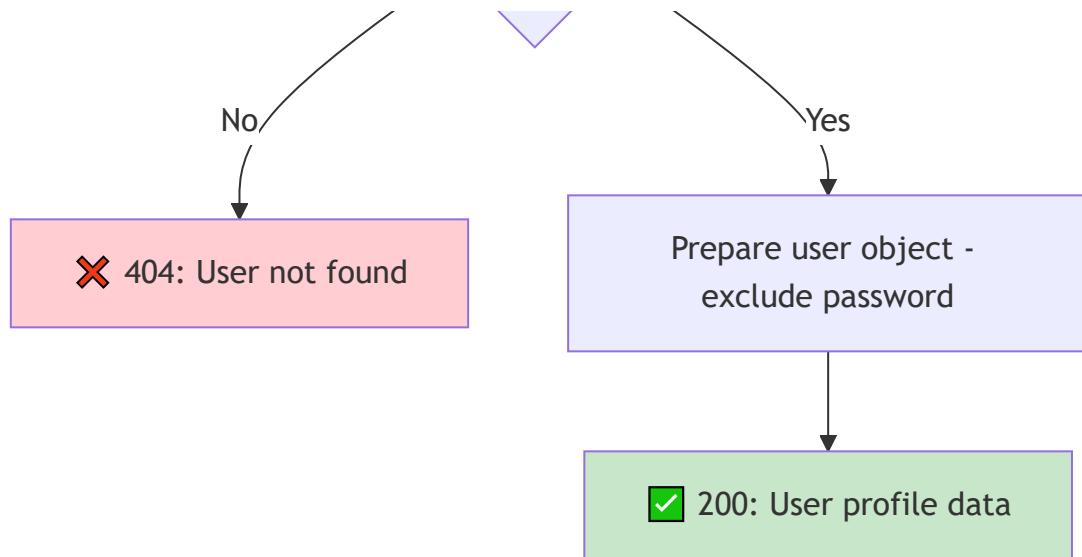


Response (200):

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

GET /api/auth/profile

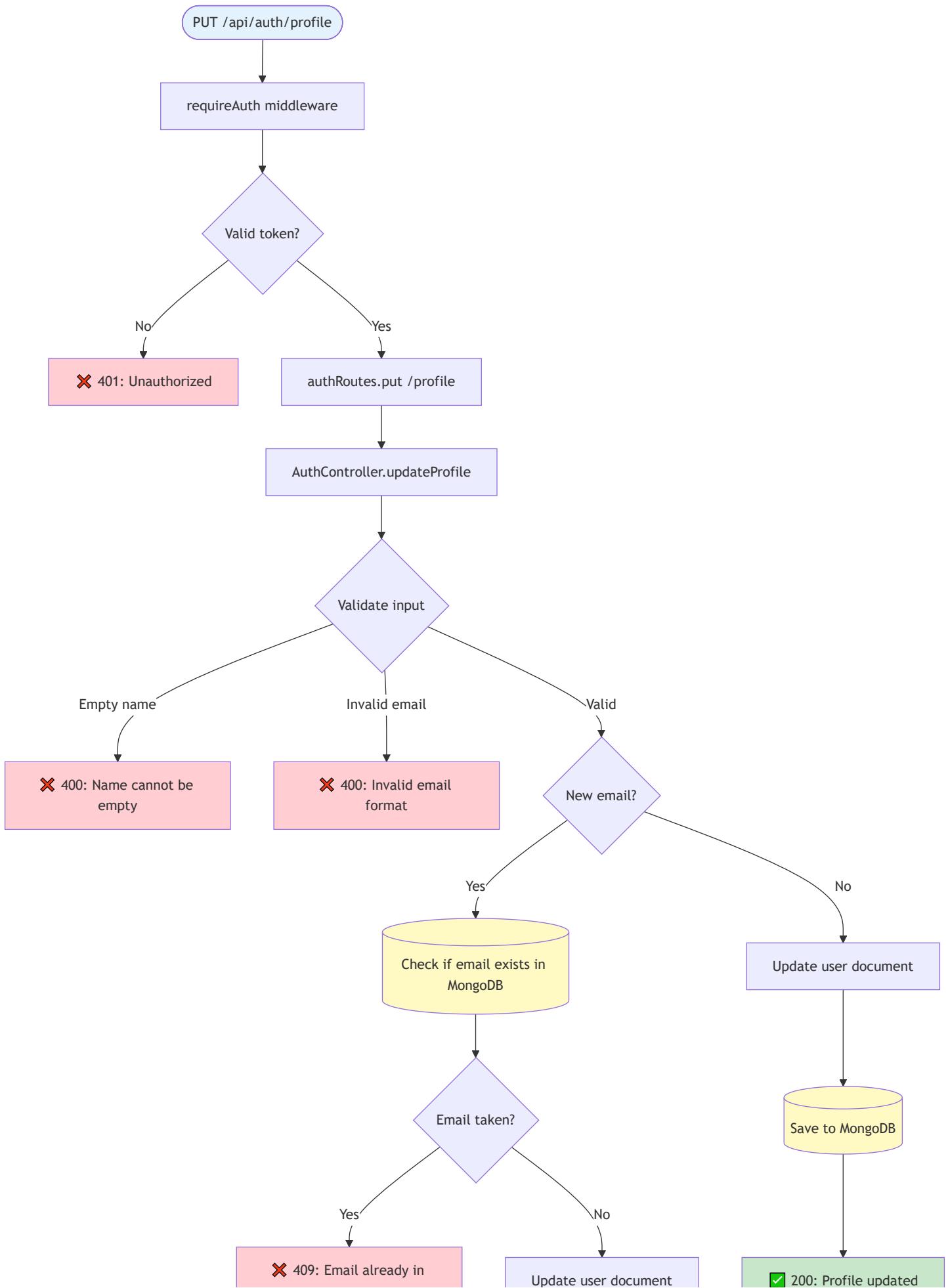


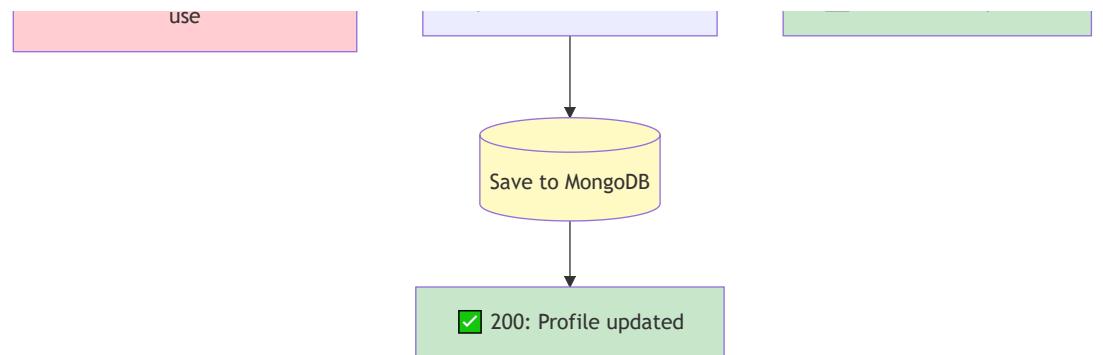


Response (200):

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

PUT /api/auth/profile





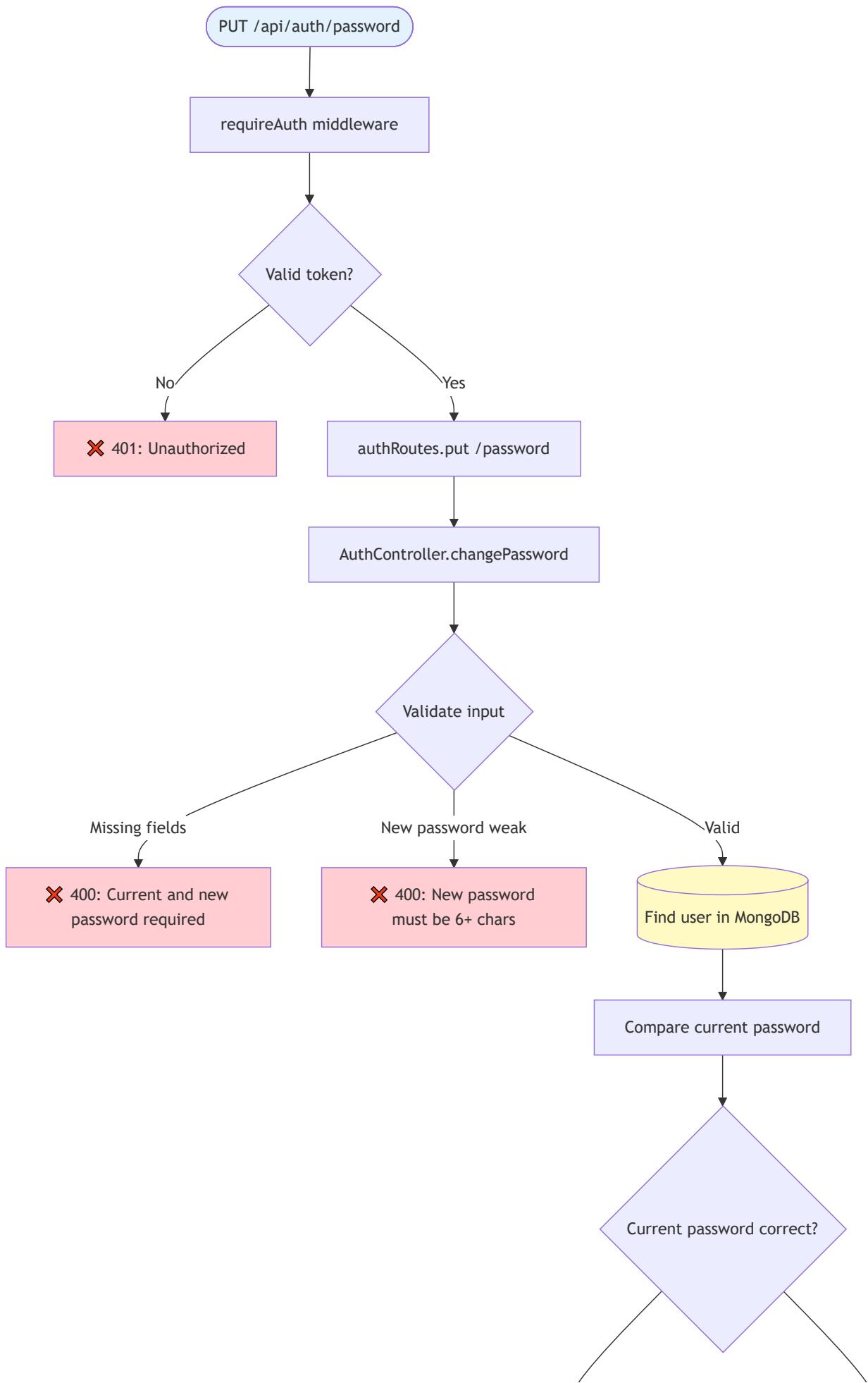
Request:

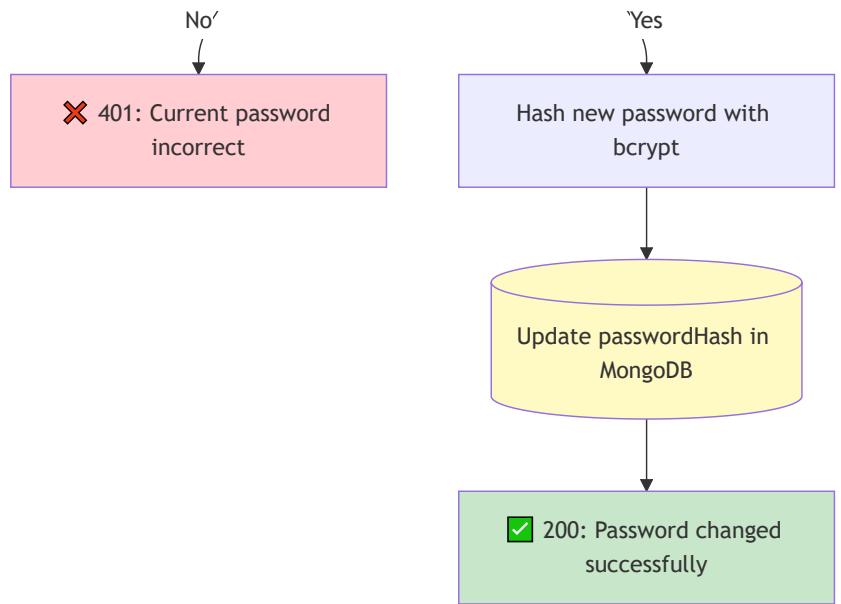
```
{  
  "name": "John Updated",  
  "email": "john.new@example.com"  
}
```

Response (200):

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

PUT /api/auth/password





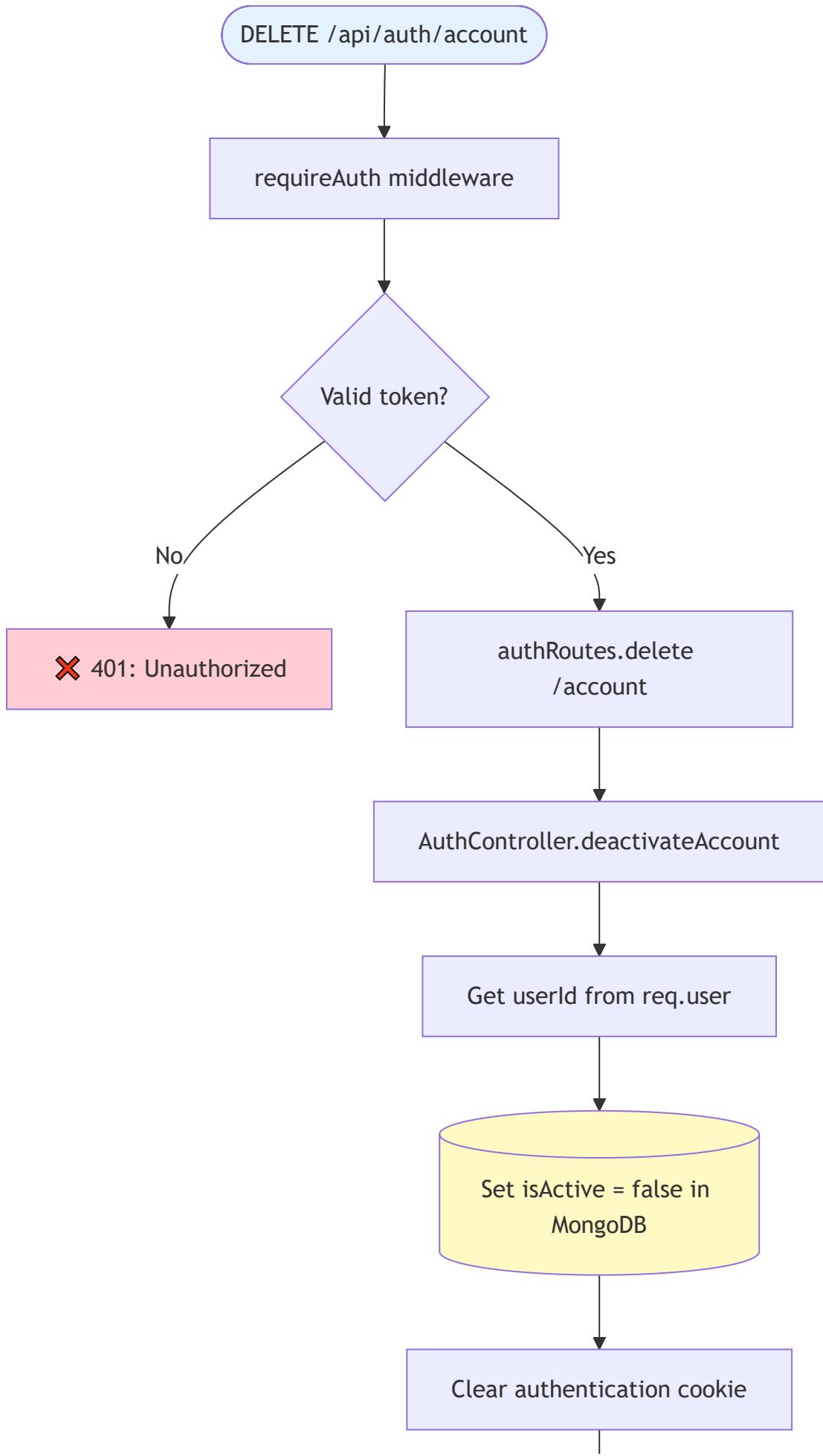
Request:

```
{  
  "currentPassword": "oldpass123",  
  "newPassword": "newpass456"  
}
```

Response (200):

```
{  
  "status": "success",  
  "message": "Password changed successfully"  
}
```

DELETE /api/auth/account

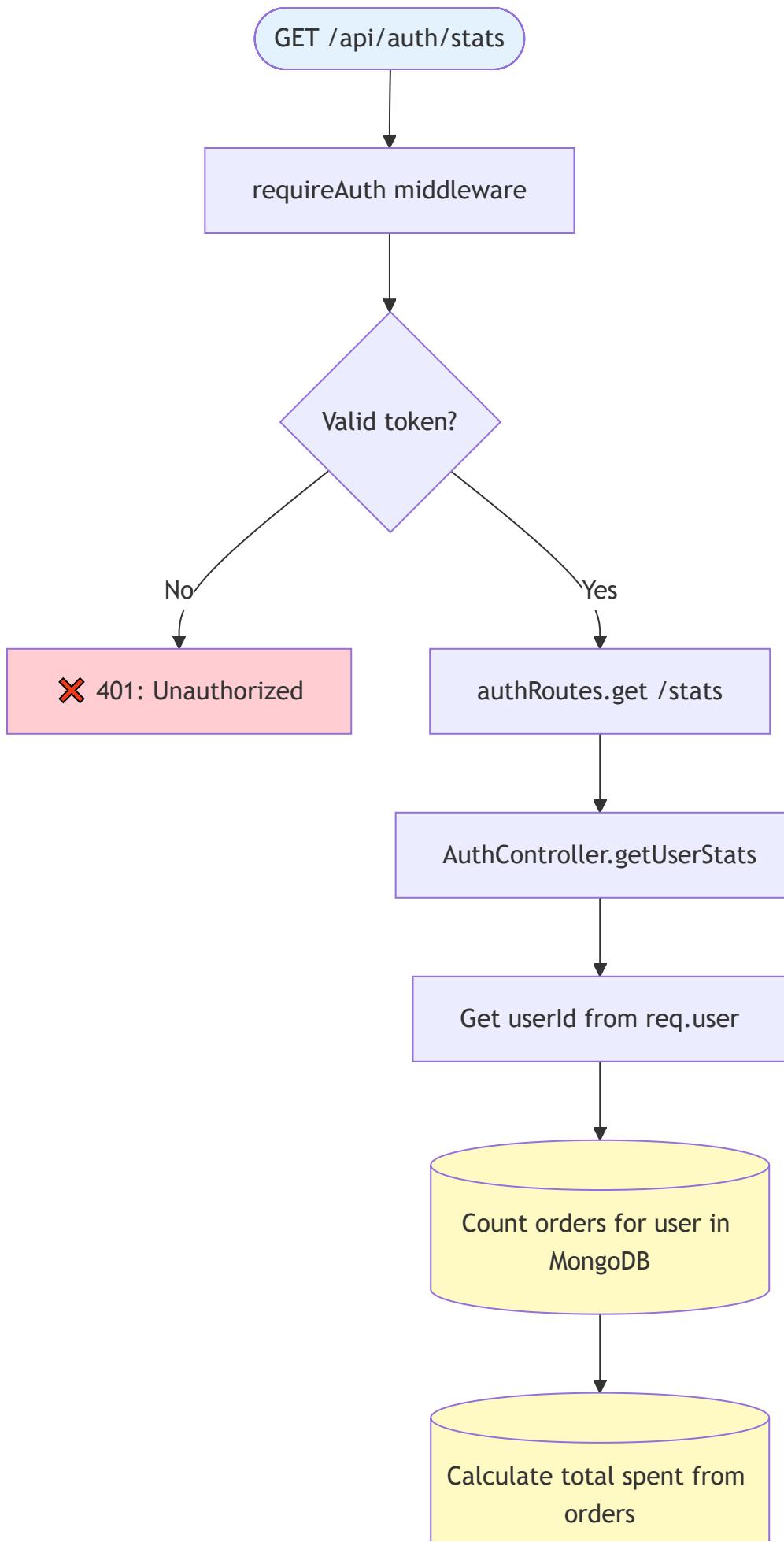


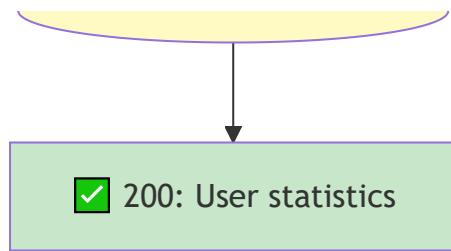
 200: Account deactivated

Response (200):

```
{  
  "status": "success",  
  "message": "Account deactivated successfully"  
}
```

GET /api/auth/stats



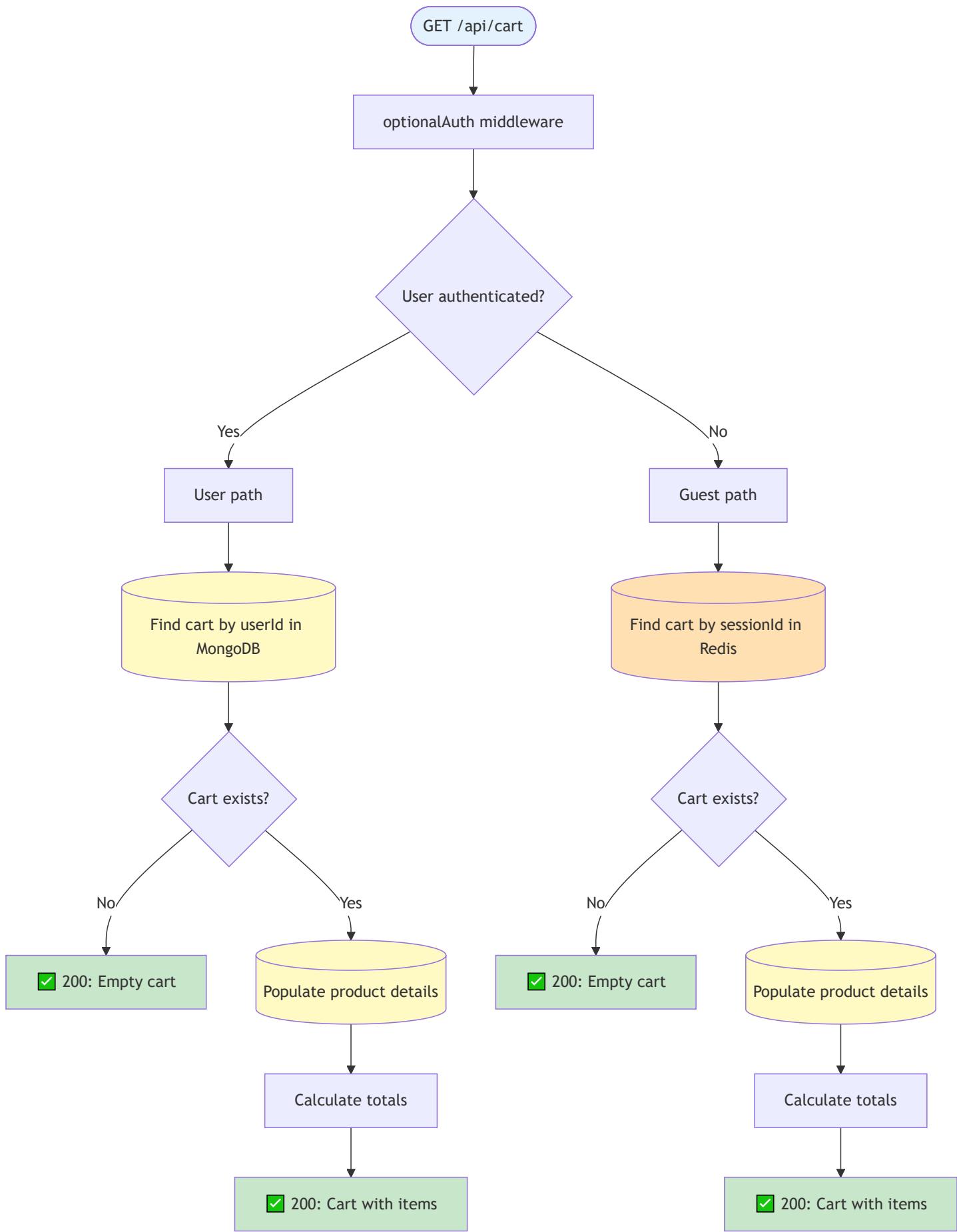


Response (200):

```
{  
  "status": "success",  
  "data": {  
    "stats": {  
      "totalOrders": 5,  
      "totalSpent": 299.95  
    }  
  }  
}
```

Cart Endpoints

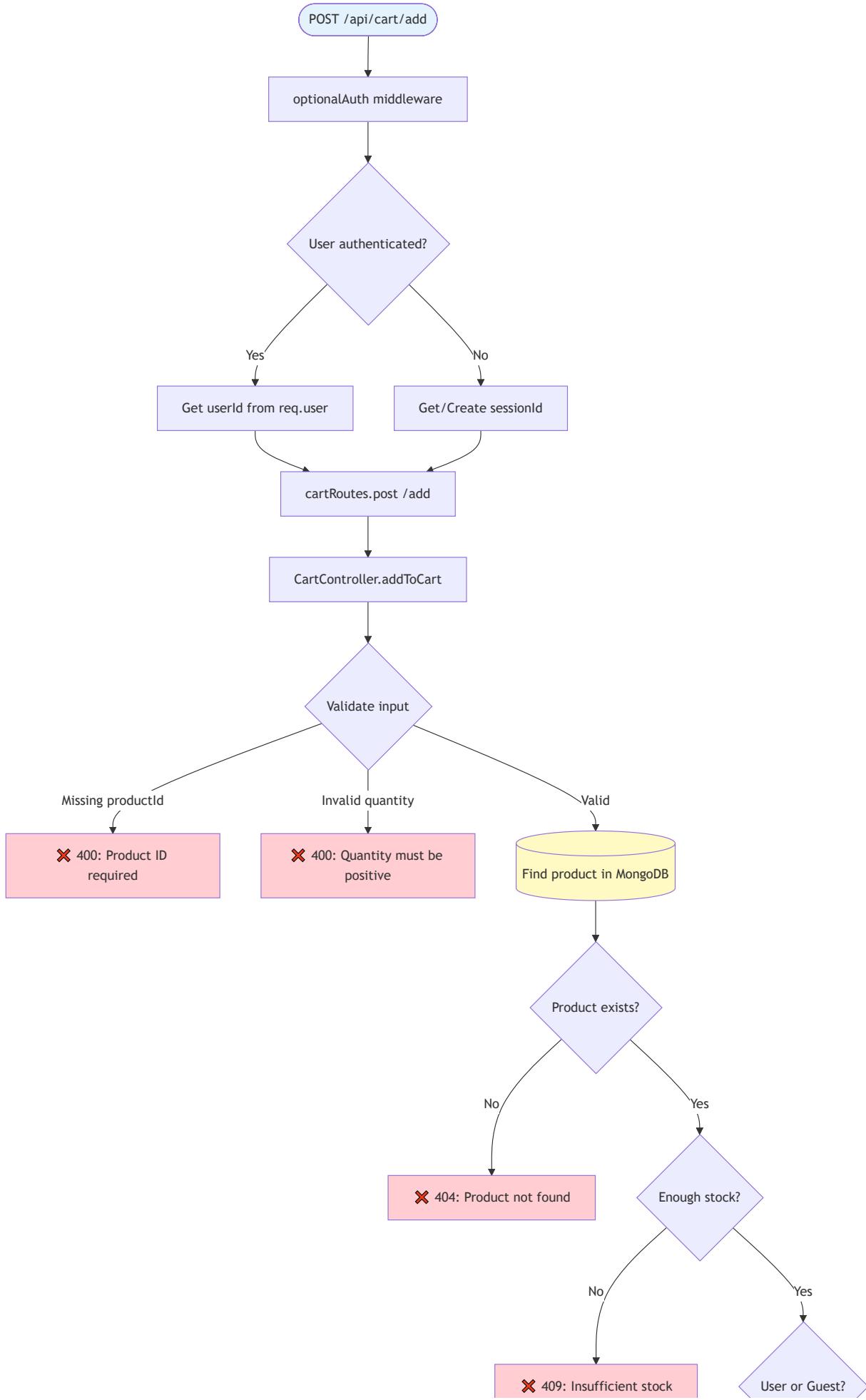
GET /api/cart

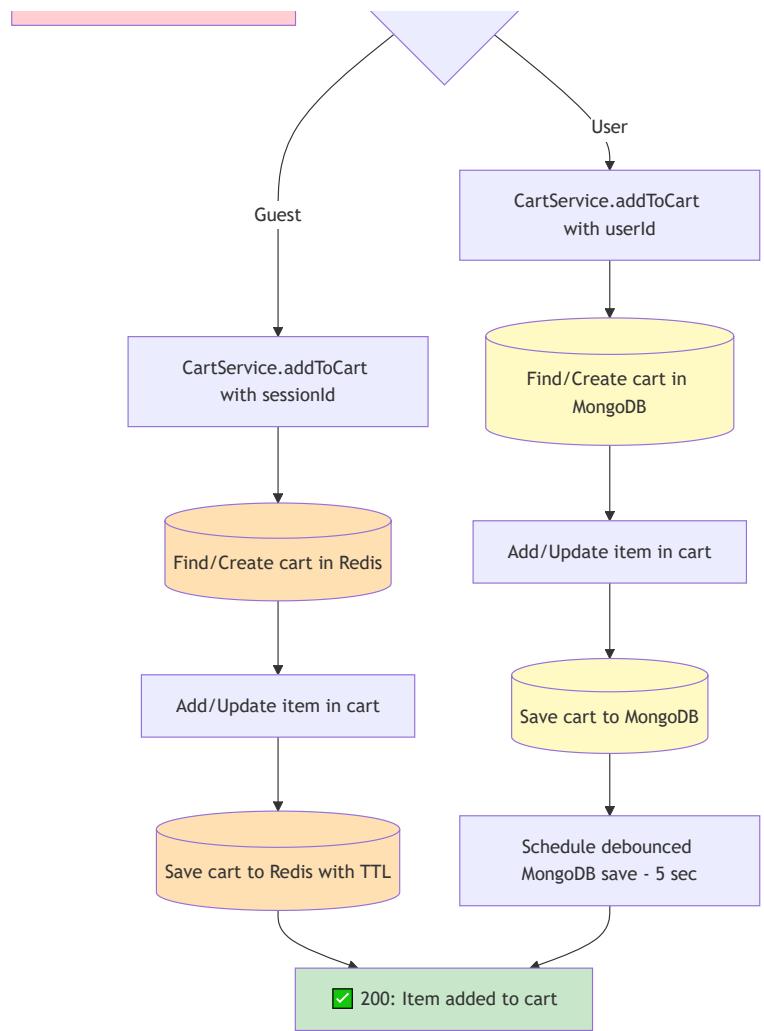


Response (200):

```
{
  "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





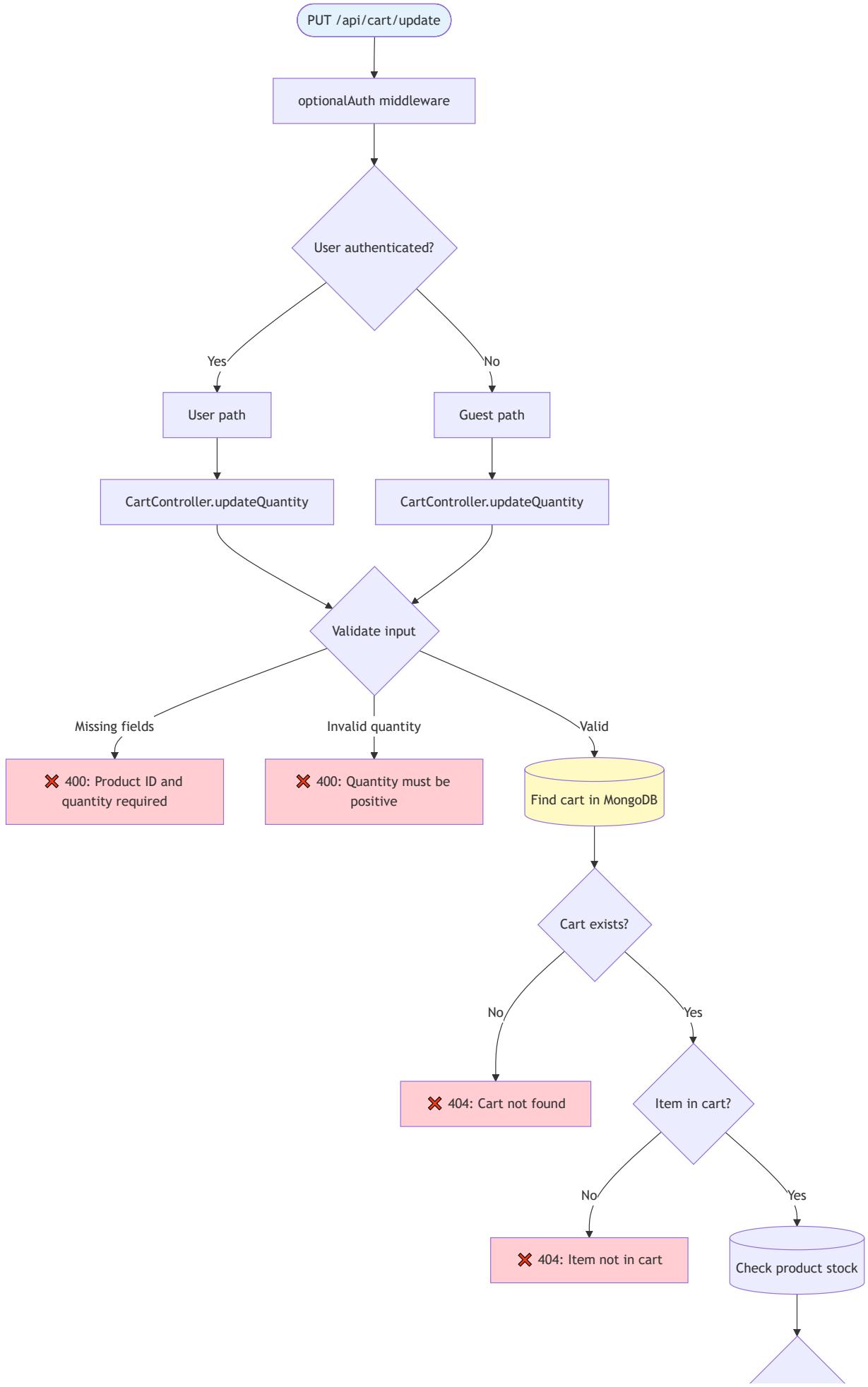
Request:

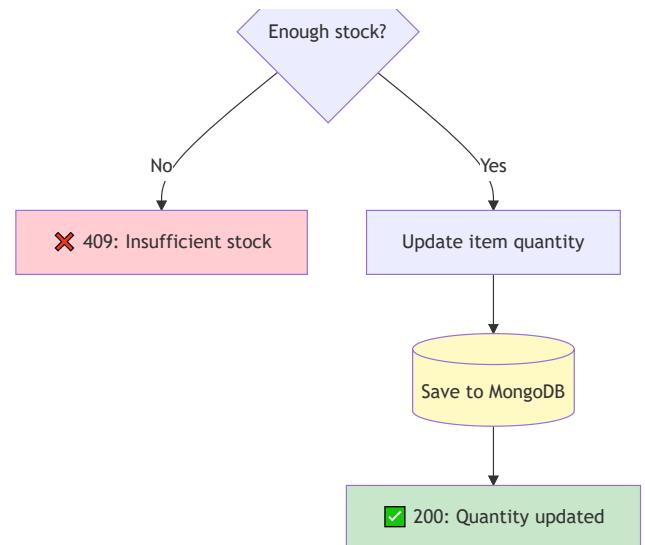
```
{
  "productId": "507f1f77bcf86cd799439011",
  "quantity": 2
}
```

Response (200):

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

PUT /api/cart/update





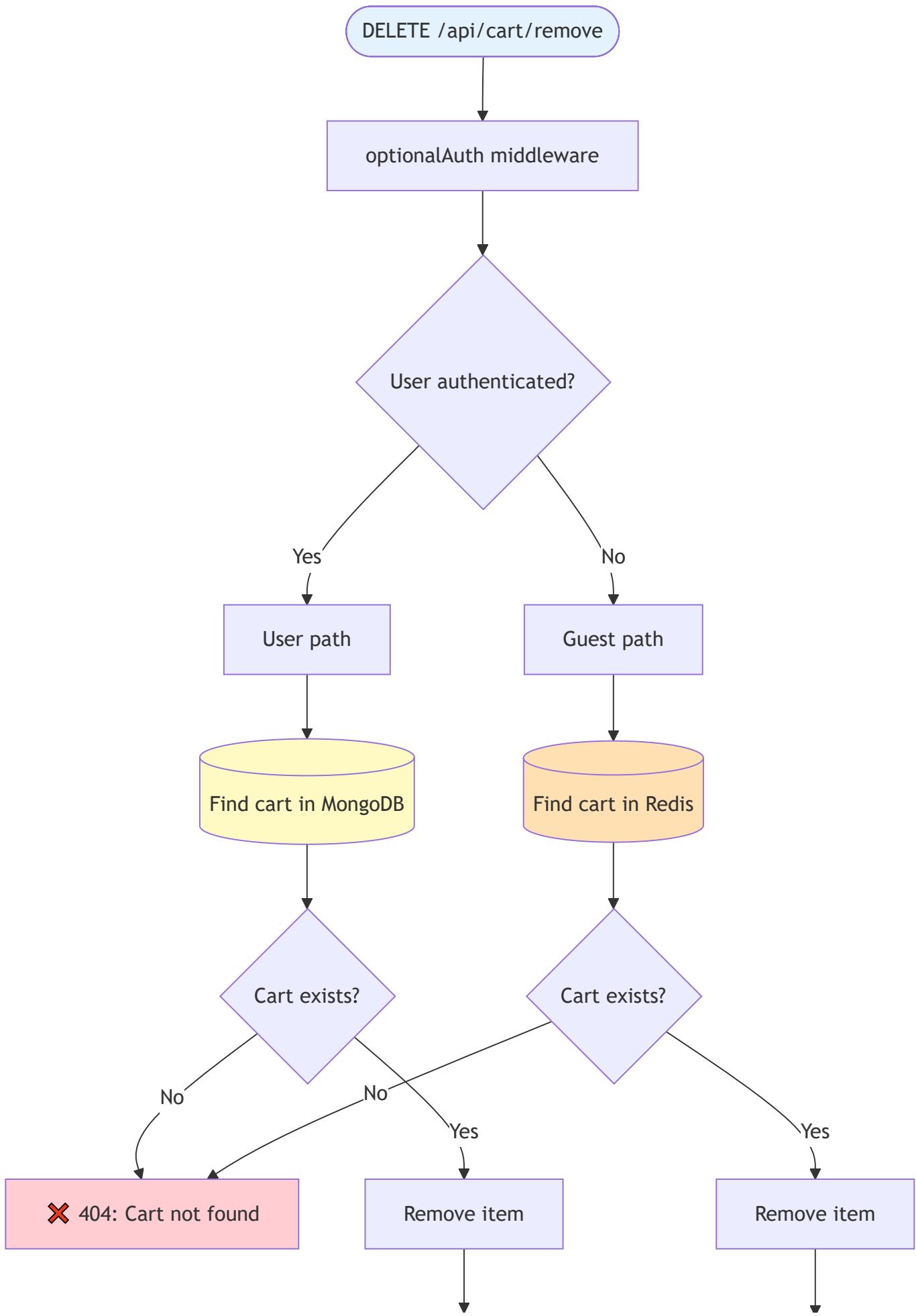
Request:

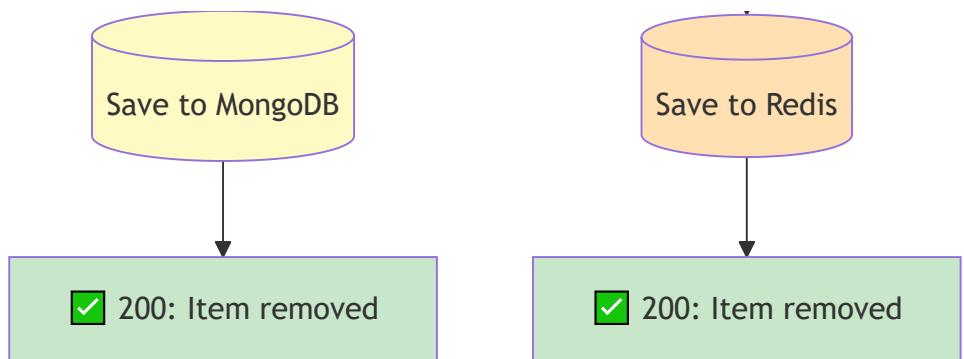
```
{
  "productId": "507f1f77bcf86cd799439011",
  "quantity": 3
}
```

Response (200):

```
{
  "status": "success",
  "data": {
    "cart": {
      "items": [
        {
          "productId": "507f1f77bcf86cd799439011",
          "quantity": 3,
          "price": 29.99
        }
      ],
      "totalItems": 3,
      "totalPrice": 89.97
    }
  }
}
```

DELETE /api/cart/remove





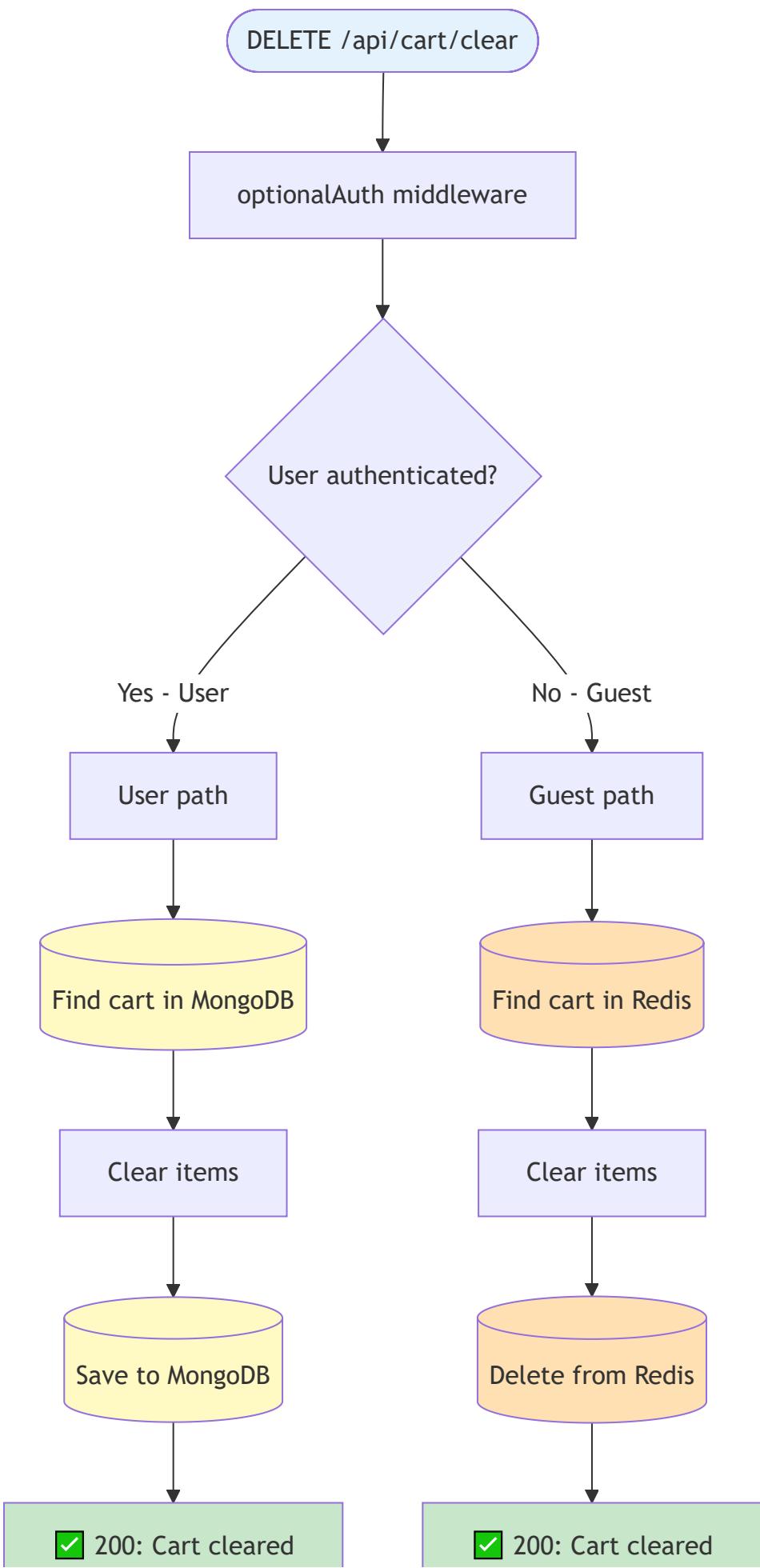
Request:

```
{  
  "productId": "507f1f77bcf86cd799439011"  
}
```

Response (200):

```
{  
  "status": "success",  
  "message": "Item removed from cart"  
}
```

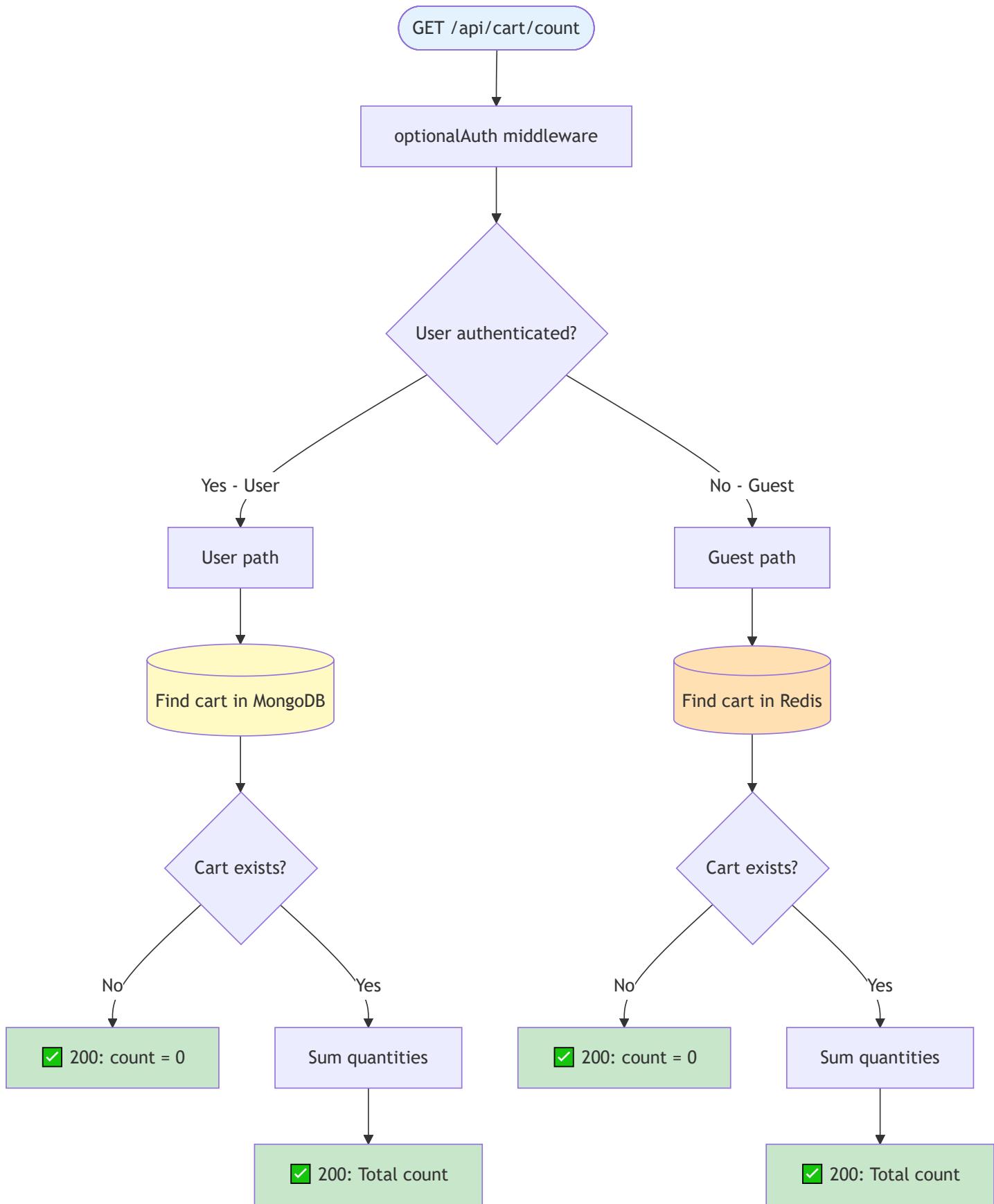
DELETE /api/cart/clear



Response (200):

```
{  
  "status": "success",  
  "message": "Cart cleared successfully"  
}
```

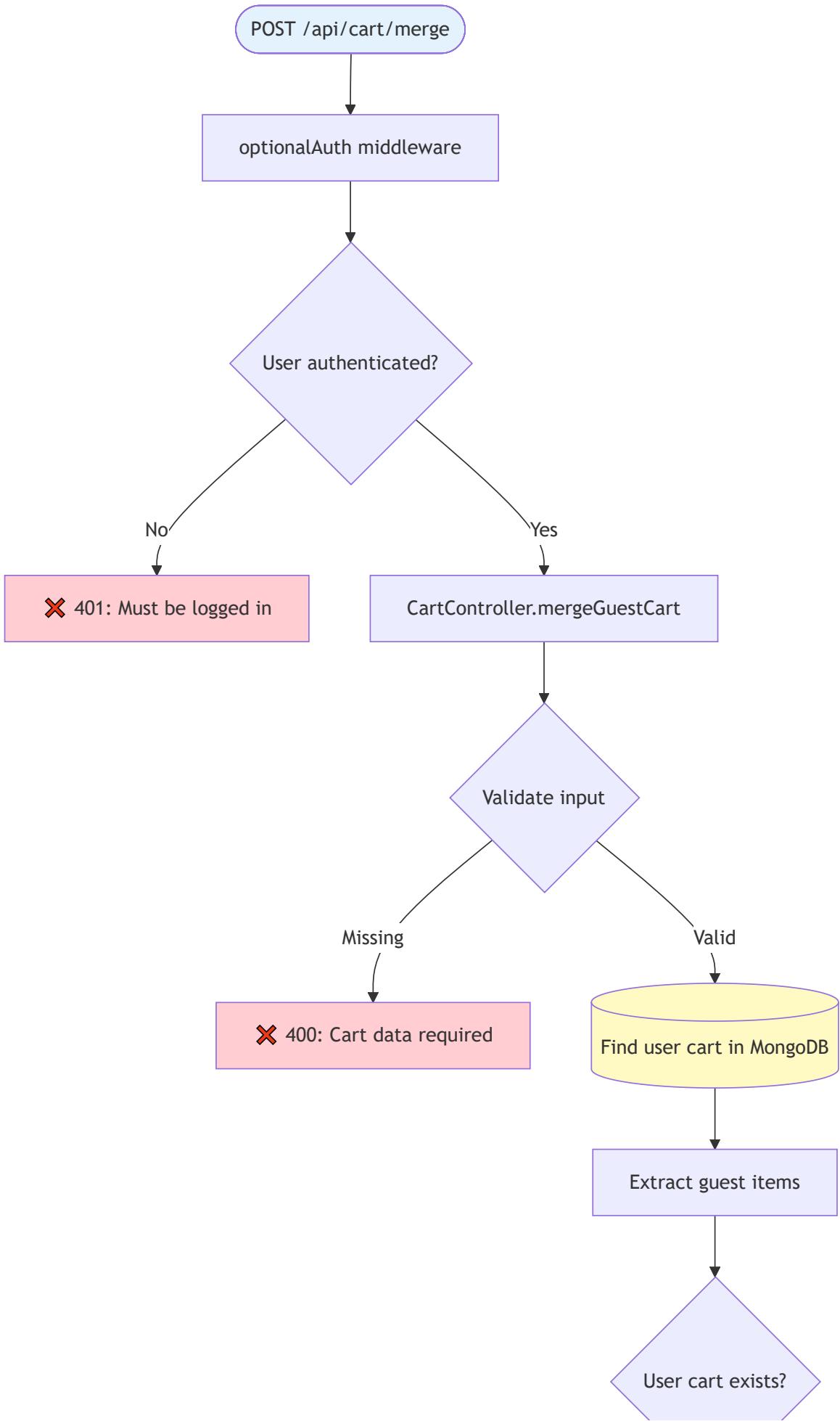
GET /api/cart/count

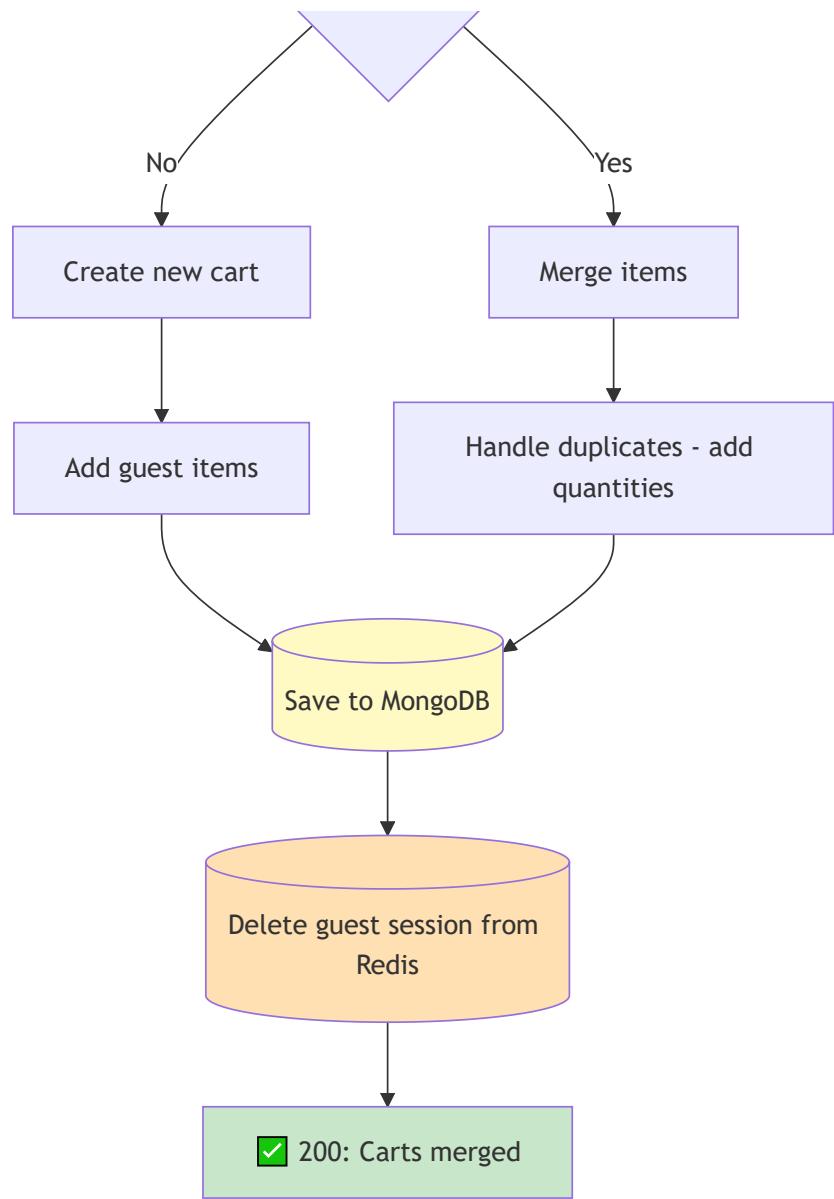


Response (200):

```
{  
  "status": "success",  
  "data": {  
    "count": 5  
  }  
}
```

POST /api/cart/merge





Request:

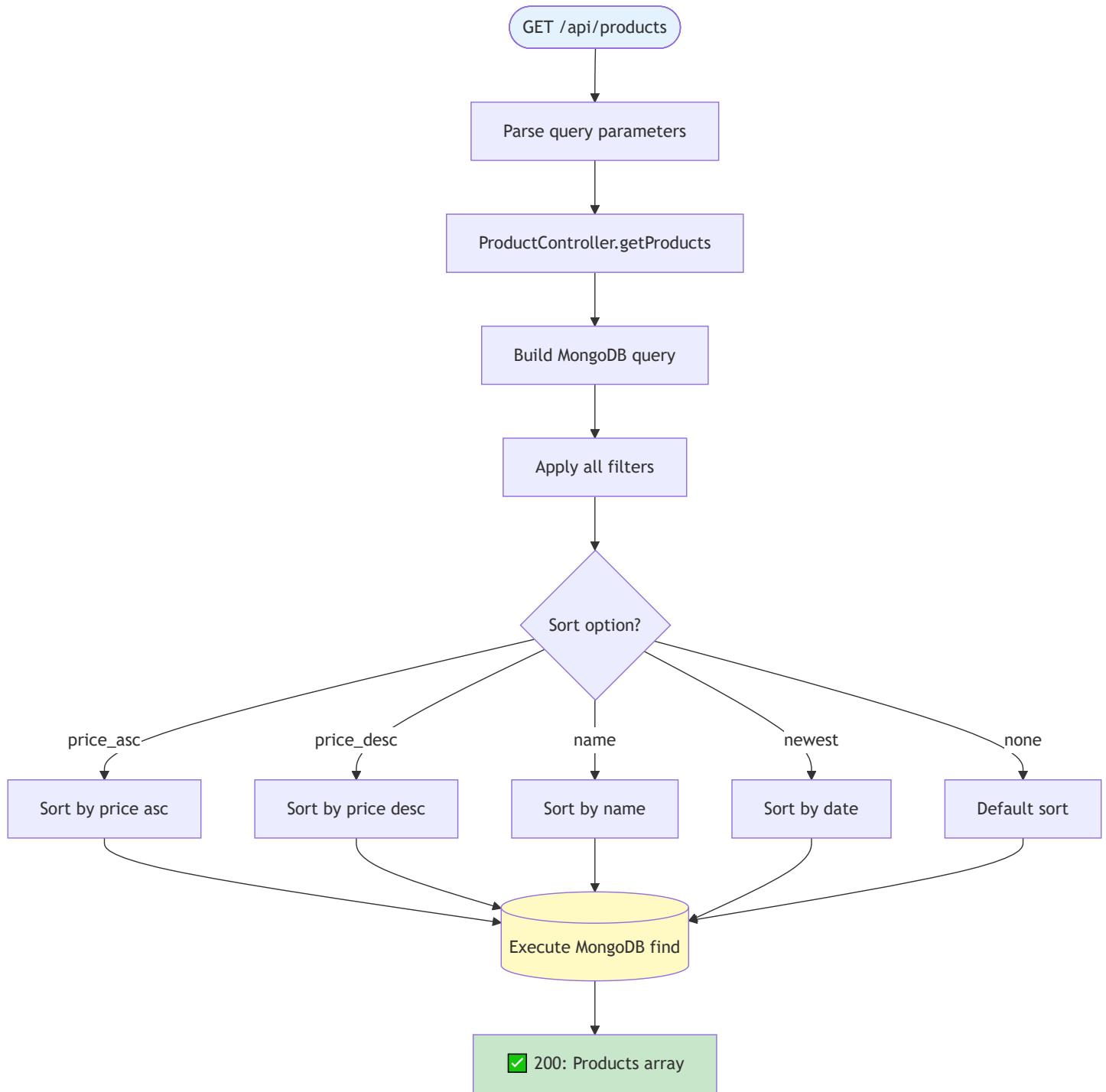
```
{
  "guestCart": {
    "items": [
      {
        "productId": "507f1f77bcf86cd799439011",
        "quantity": 2
      }
    ]
  }
}
```

Response (200):

```
{  
  "status": "success",  
  "data": {  
    "cart": {  
      "items": [  
        {  
          "productId": "507f1f77bcf86cd799439011",  
          "quantity": 4  
        }  
      ],  
      "totalItems": 4  
    }  
  }  
}
```

Product Endpoints

GET /api/products



Query Parameters:

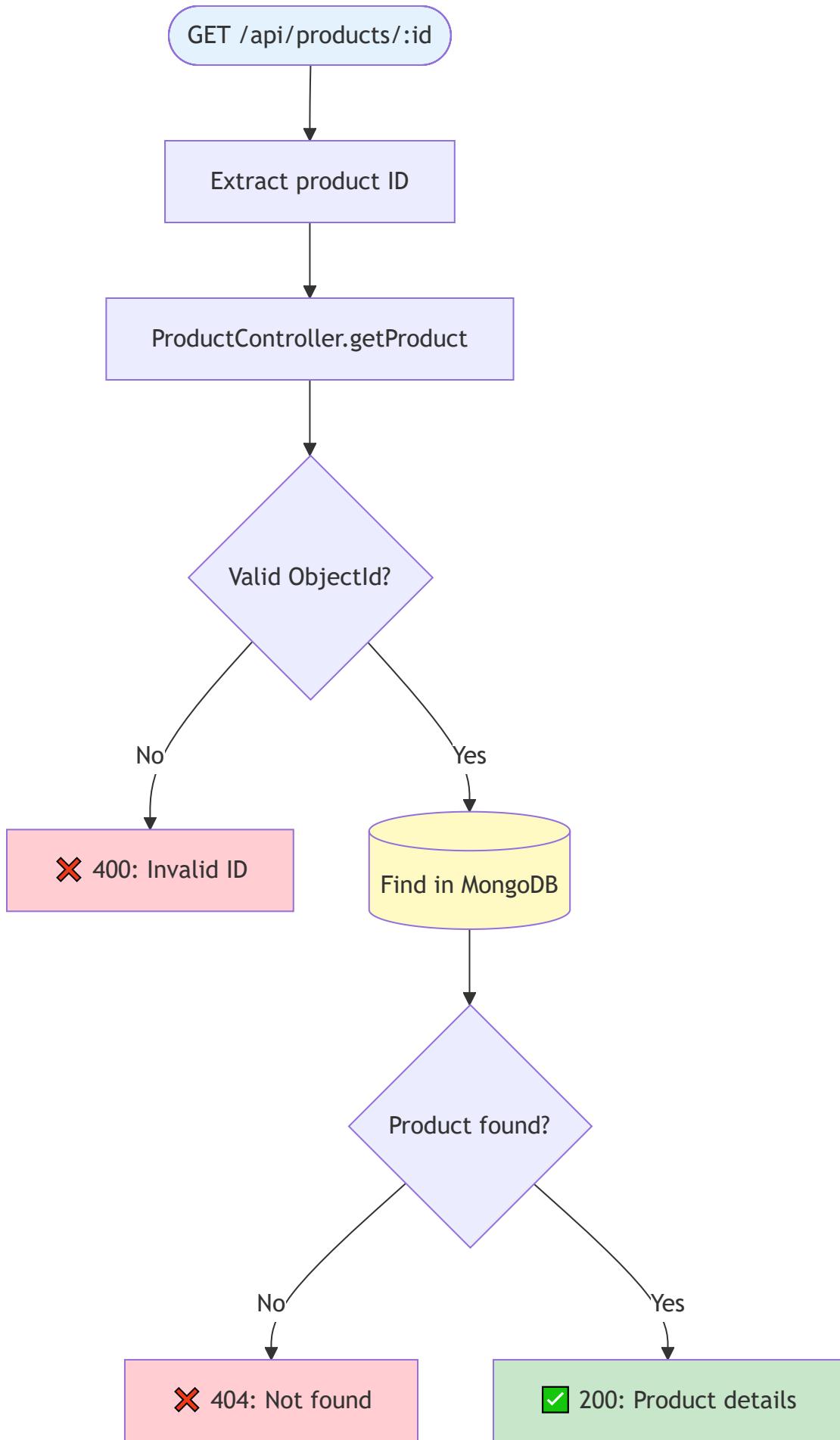
- `search` - Text search on name/description
- `category` - Filter by category
- `minPrice` - Minimum price

- `maxPrice` - Maximum price
- `sort` - Sort order (price_asc, price_desc, name, newest)

Response (200):

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

GET /api/products/:id



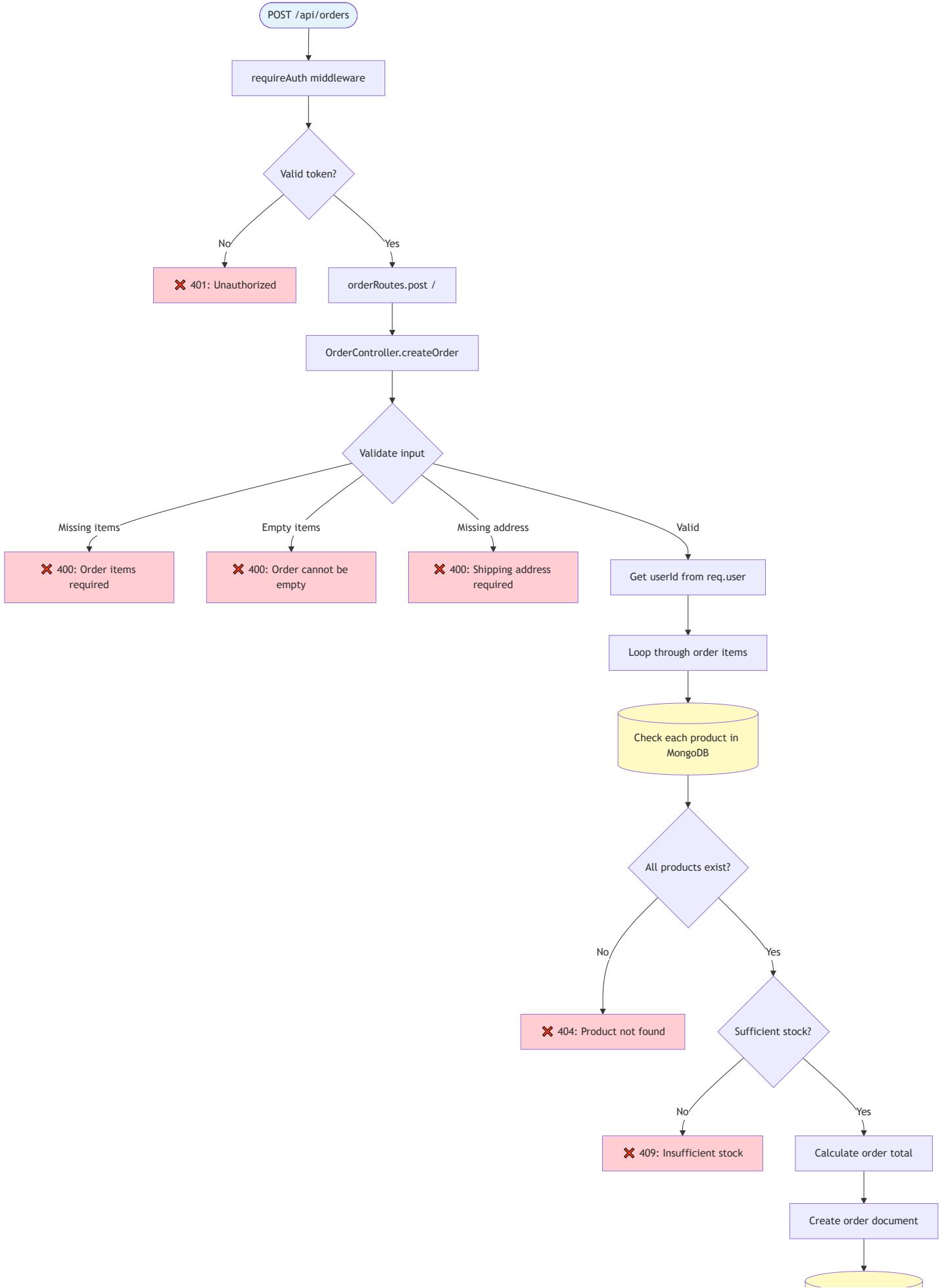
Response (200):

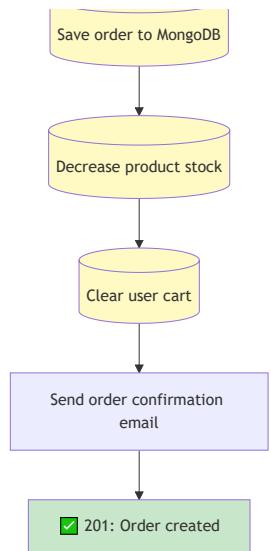
```
{  
  "status": "success",  
  "data": {  
    "product": {  
      "_id": "507f1f77bcf86cd799439011",  
      "name": "Product Name",  
      "description": "Detailed product description",  
      "price": 29.99,  
      "stock": 100,  
      "category": "electronics",  
      "imageUrl": "https://example.com/image.jpg",  
      "createdAt": "2024-01-15T10:30:00.000Z"  
    }  
  }  
}
```



Order Endpoints

POST /api/orders





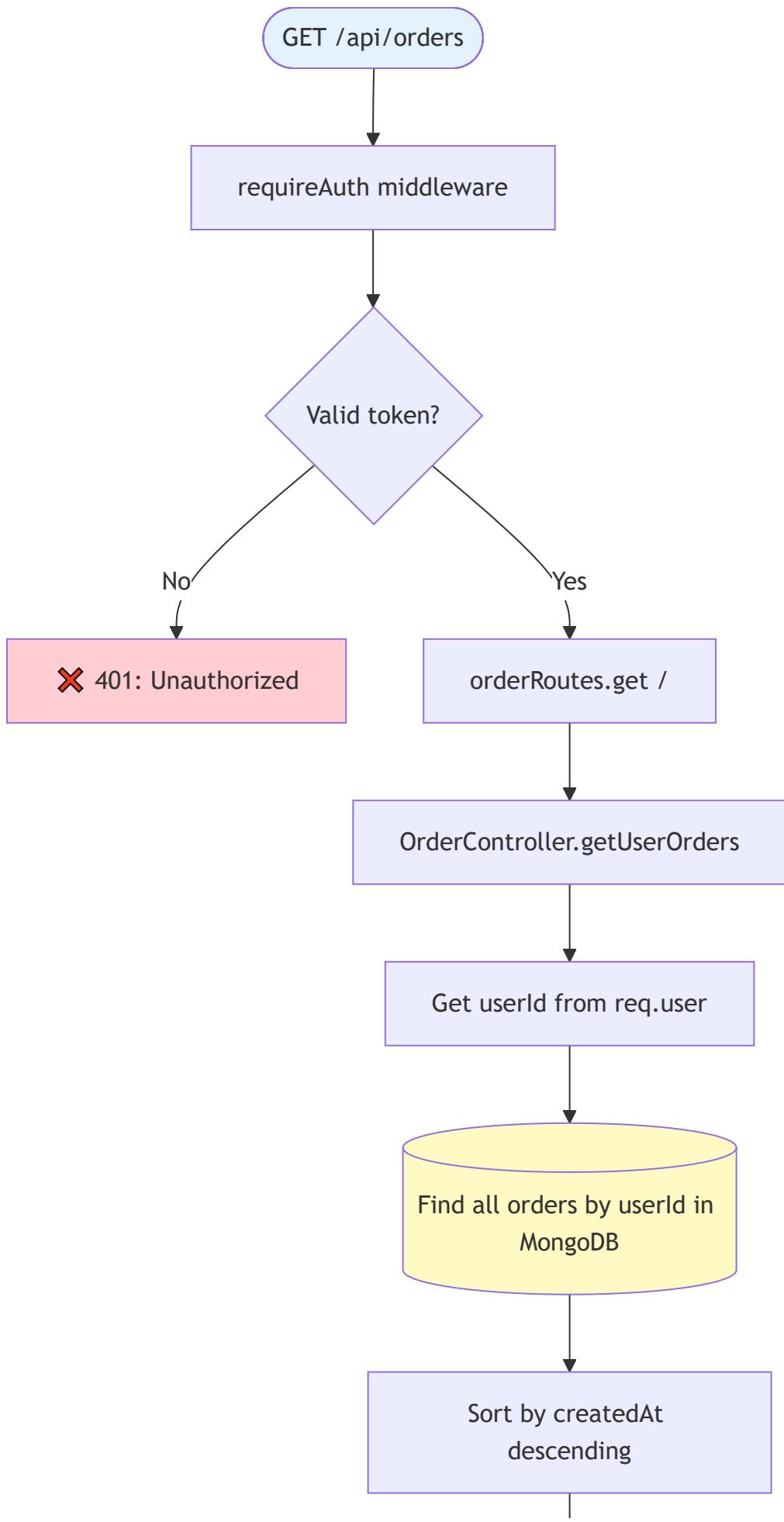
Request:

```
{
  "items": [
    {
      "productId": "507f1f77bcf86cd799439011",
      "quantity": 2
    }
  ],
  "shippingAddress": {
    "street": "123 Main St",
    "city": "New York",
    "zipCode": "10001",
    "country": "USA"
  }
}
```

Response (201):

```
{
  "status": "success",
  "data": {
    "order": {
      "_id": "507f1f77bcf86cd799439012",
      "userId": "507f1f77bcf86cd799439011",
      "items": [
        {
          "productId": "507f1f77bcf86cd799439011",
          "productName": "Product Name",
          "price": 29.99,
          "quantity": 2,
          "subtotal": 59.98
        }
      ],
      "totalAmount": 59.98,
      "status": "pending",
      "shippingAddress": {...},
      "createdAt": "2024-01-15T10:30:00.000Z"
    }
  }
}
```

GET /api/orders



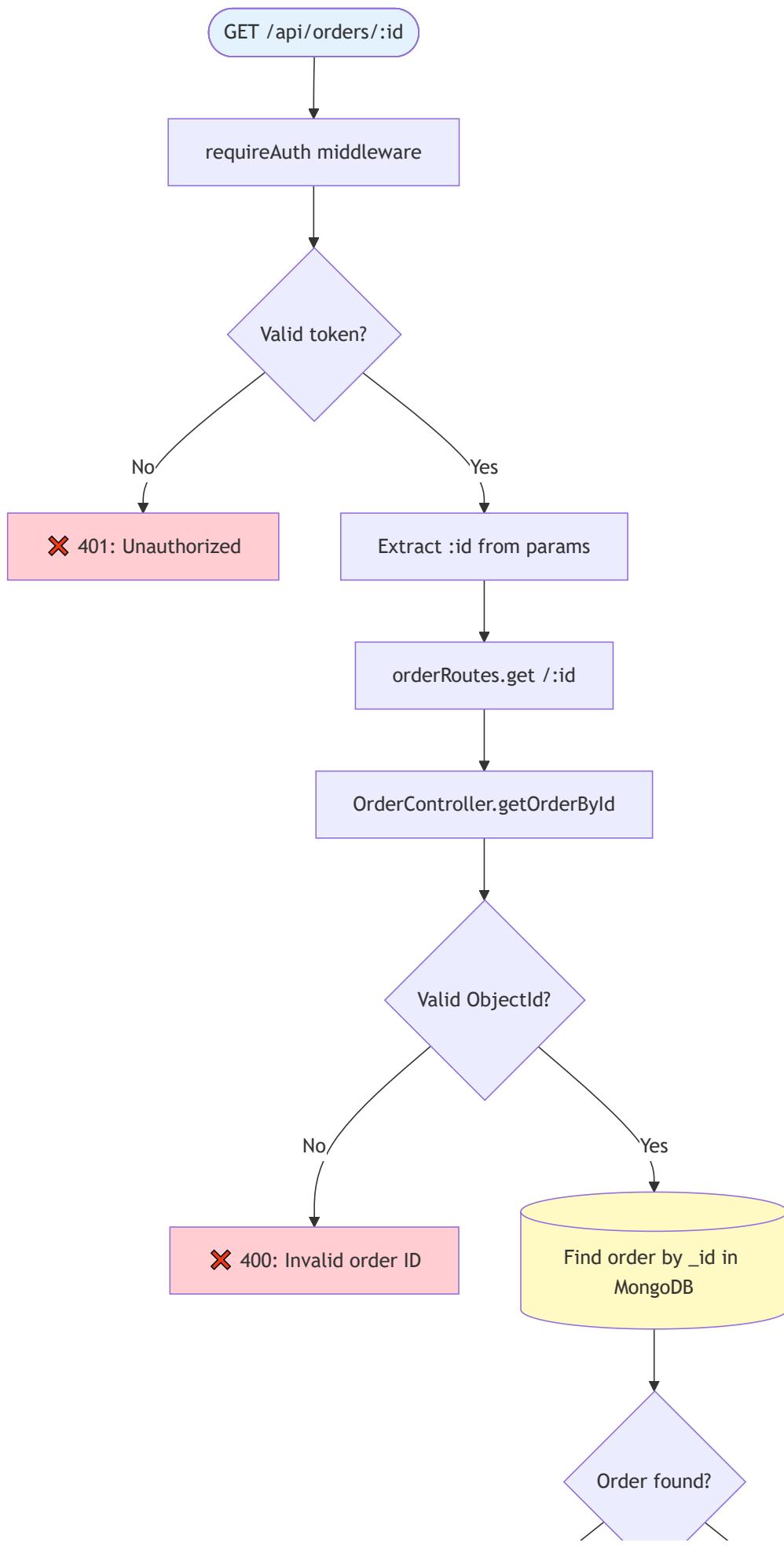


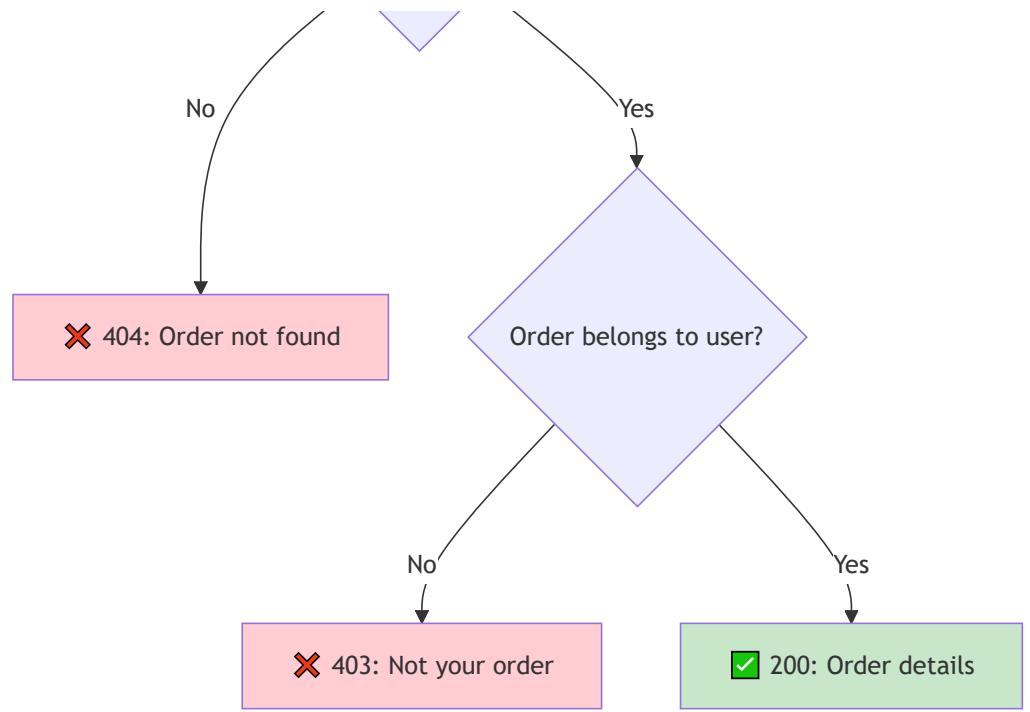
✓ 200: Orders array

Response (200):

```
{  
  "status": "success",  
  "data": {  
    "orders": [  
      {  
        "_id": "507f1f77bcf86cd799439012",  
        "userId": "507f1f77bcf86cd799439011",  
        "items": [...],  
        "totalAmount": 59.98,  
        "status": "pending",  
        "createdAt": "2024-01-15T10:30:00.000Z"  
      }  
    ]  
  }  
}
```

GET /api/orders/:id

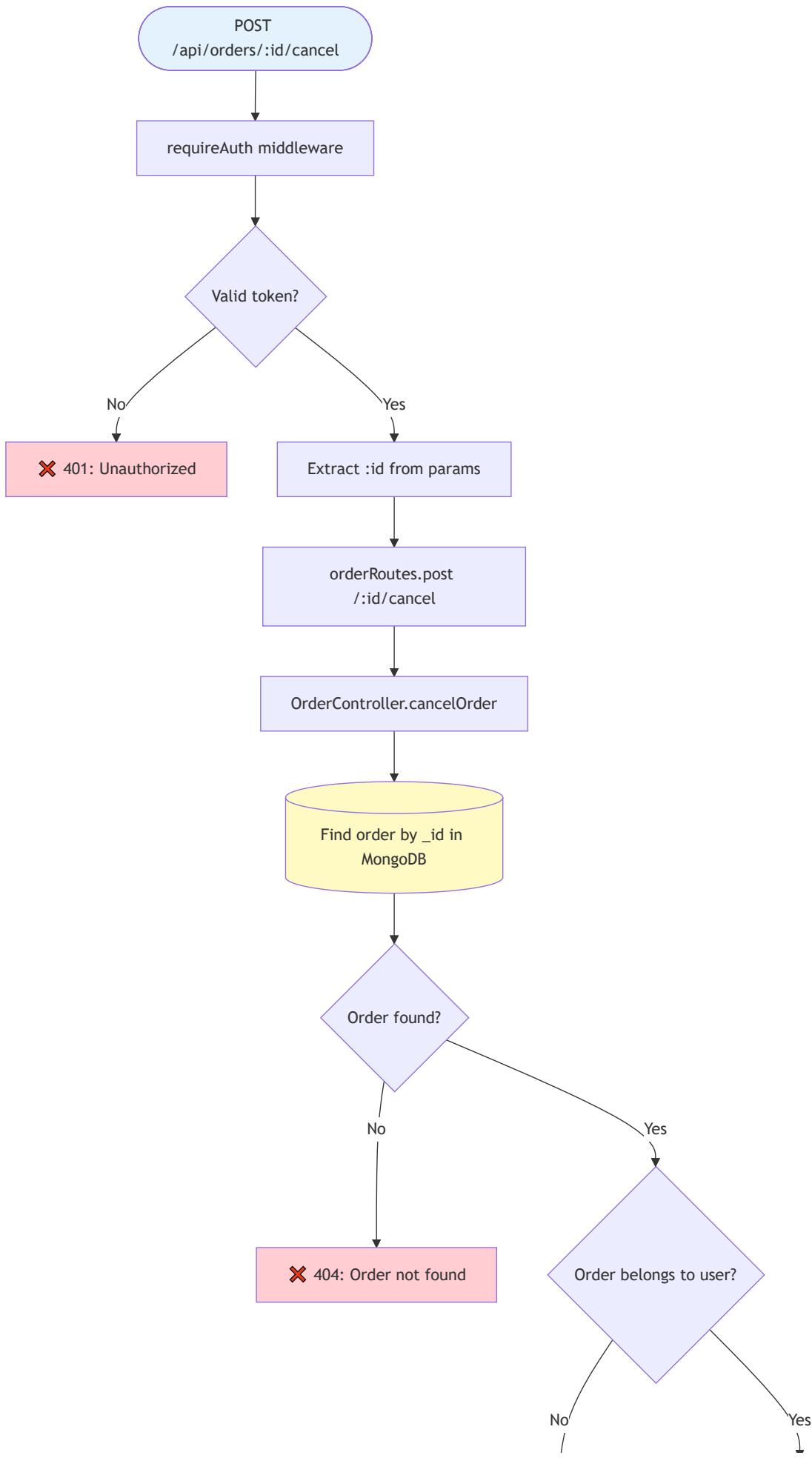


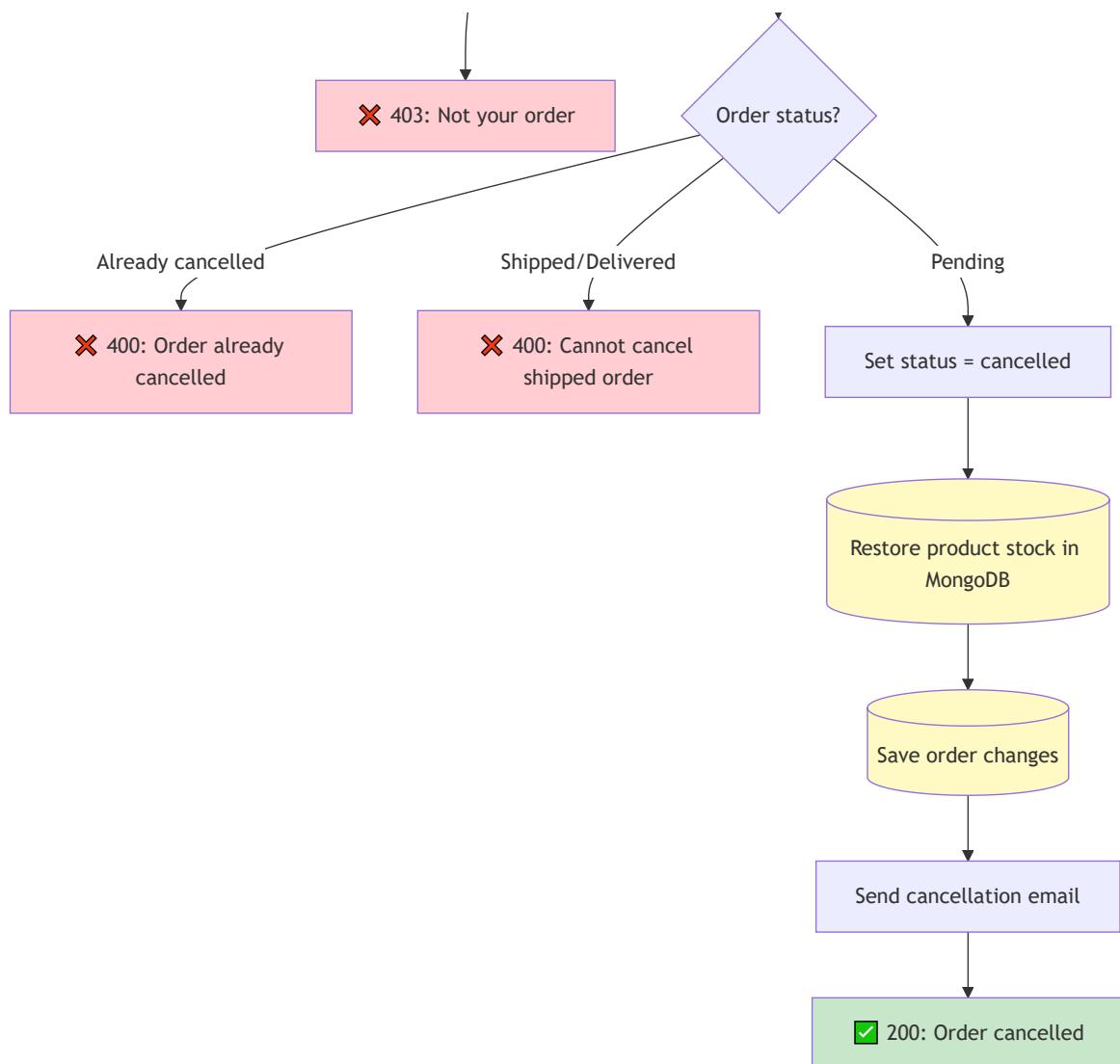


Response (200):

```
{
  "status": "success",
  "data": {
    "order": {
      "_id": "507f1f77bcf86cd799439012",
      "userId": "507f1f77bcf86cd799439011",
      "items": [...],
      "totalAmount": 59.98,
      "status": "pending",
      "shippingAddress": {...},
      "createdAt": "2024-01-15T10:30:00.000Z"
    }
  }
}
```

POST /api/orders/:id/cancel

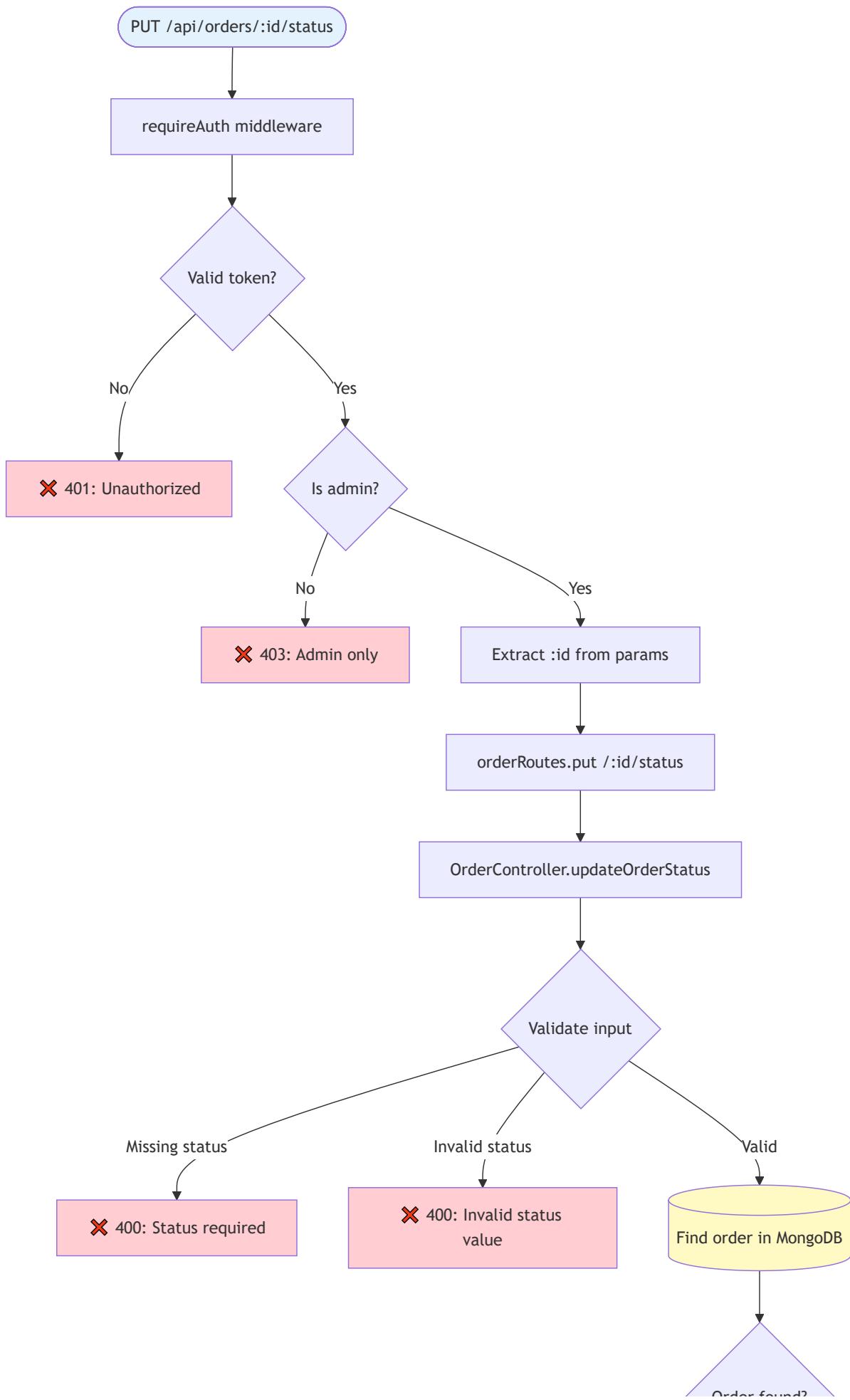


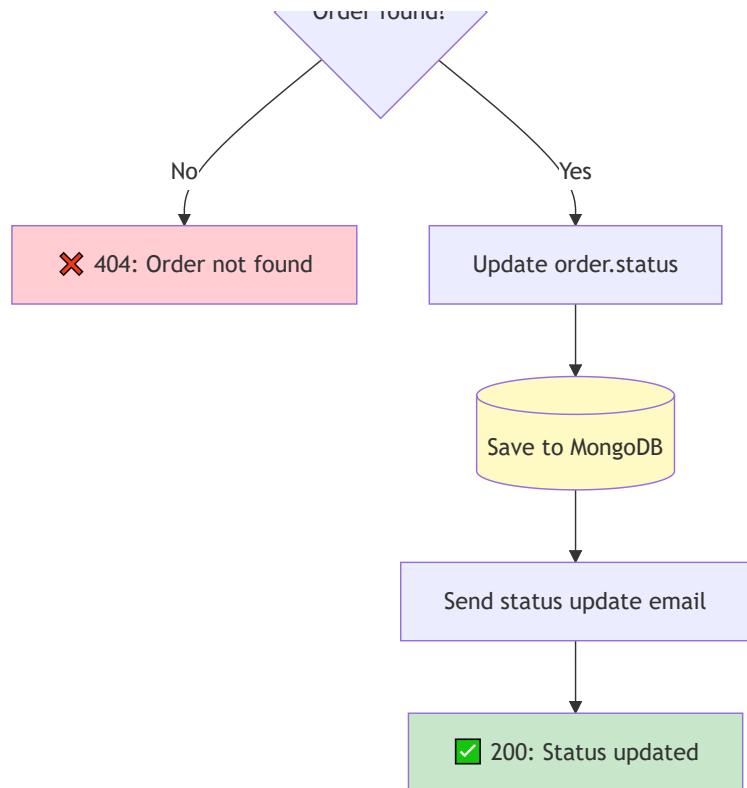


Response (200):

```
{
  "status": "success",
  "message": "Order cancelled successfully",
  "data": {
    "order": {
      "_id": "507f1f77bcf86cd799439012",
      "status": "cancelled"
    }
  }
}
```

PUT /api/orders/:id/status





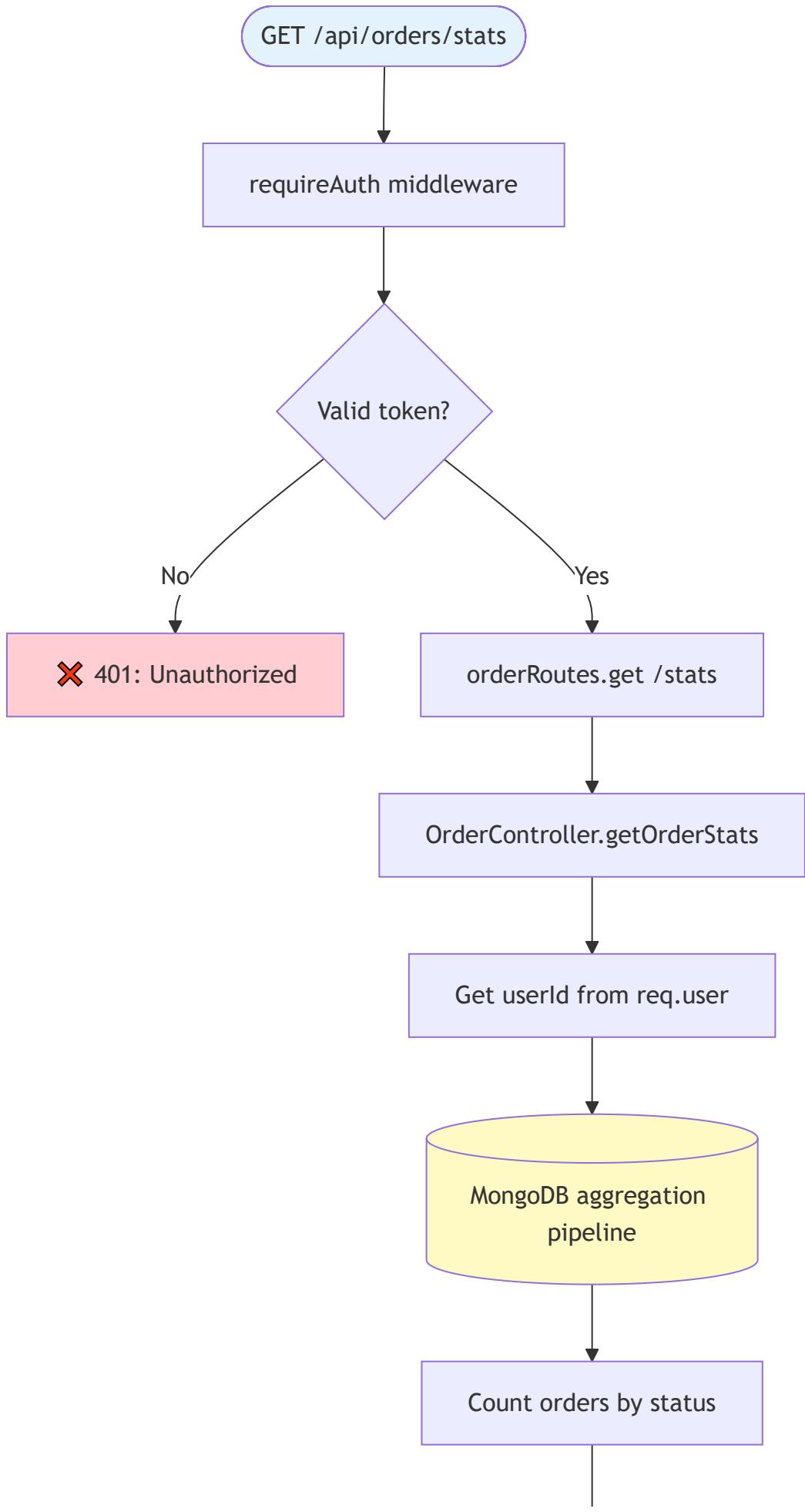
Request:

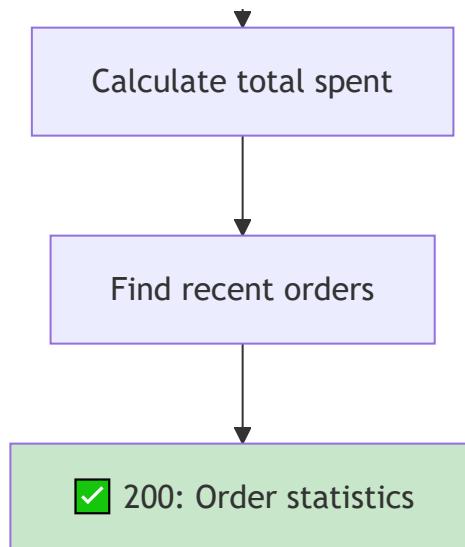
```
{
  "status": "shipped"
}
```

Response (200):

```
{
  "status": "success",
  "message": "Order status updated",
  "data": {
    "order": {
      "_id": "507f1f77bcf86cd799439012",
      "status": "shipped"
    }
  }
}
```

GET /api/orders/stats



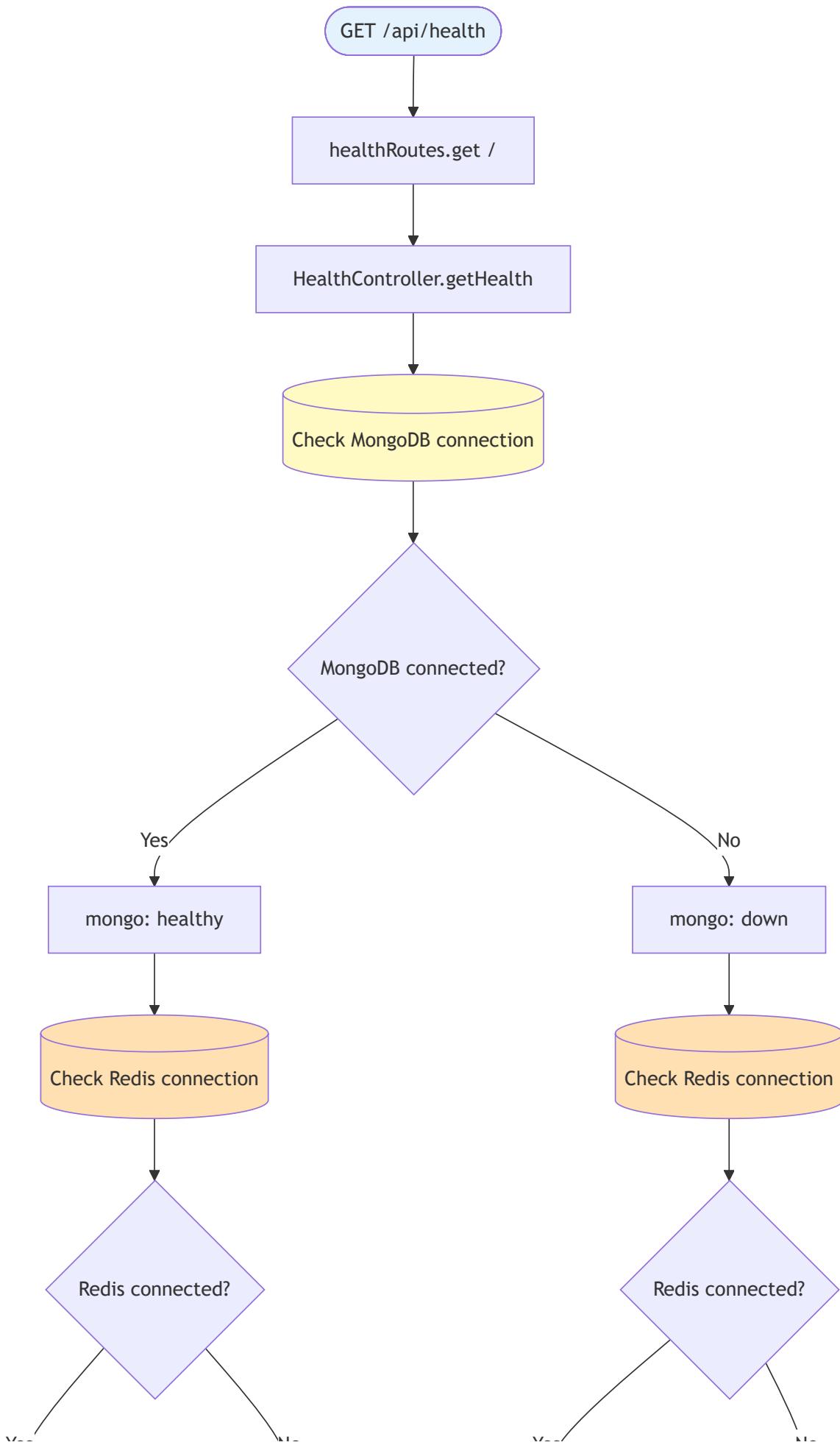


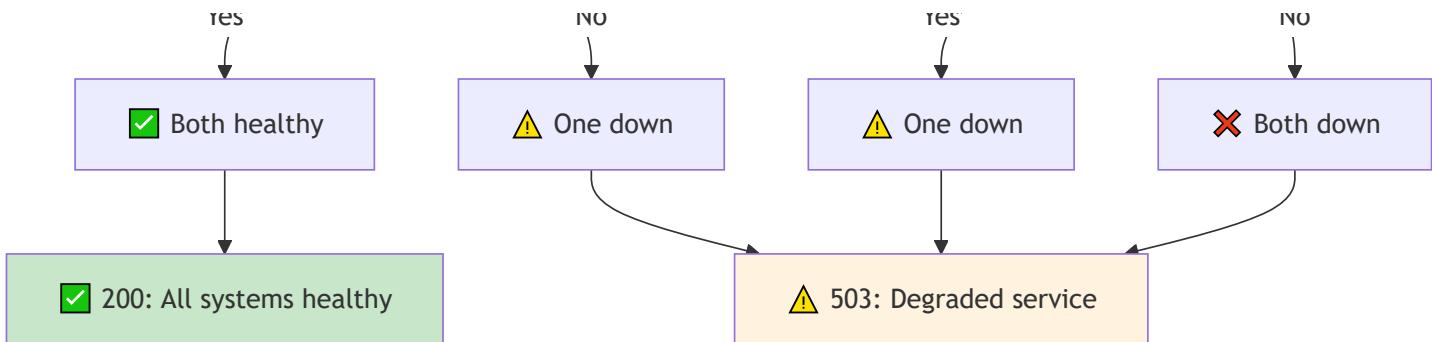
Response (200):

```
{  
  "status": "success",  
  "data": {  
    "stats": {  
      "totalOrders": 5,  
      "totalSpent": 299.95,  
      "ordersByStatus": {  
        "pending": 1,  
        "shipped": 2,  
        "delivered": 2,  
        "cancelled": 0  
      }  
    }  
  }  
}
```

❤️ Health Endpoints

GET /api/health

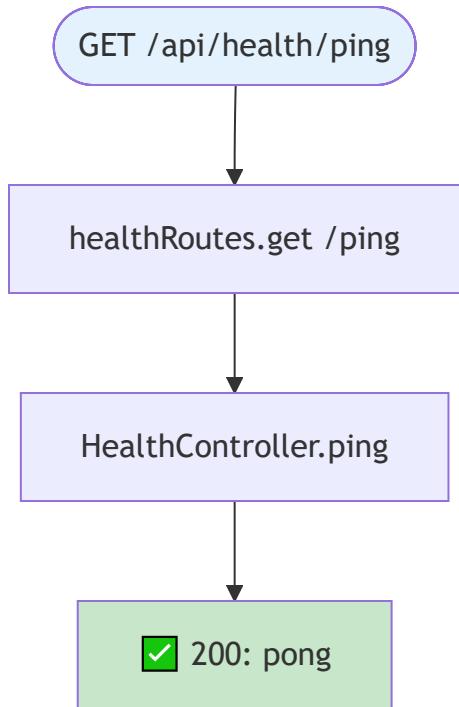




Response (200):

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

GET /api/health/ping



Response (200):

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

🎯 Legend

Color Coding:

- ● **Blue** - Request entry point
- ● **Green** - Successful response
- ● **Red** - Error response
- ● **Yellow (MongoDB)** - MongoDB operations
- ● **Orange (Redis)** - Redis operations

Common HTTP Status Codes:

- **200 OK** - Success

- **201 Created** - Resource created
- **400 Bad Request** - Invalid input
- **401 Unauthorized** - Authentication required
- **403 Forbidden** - Permission denied
- **404 Not Found** - Resource not found
- **409 Conflict** - Resource conflict (e.g., duplicate, stock issue)
- **429 Too Many Requests** - Rate limit exceeded
- **503 Service Unavailable** - Service degraded

Middleware Flow:

1. **Rate Limit** - Check request rate
2. **CORS** - Validate origin
3. **Parse Body** - Parse JSON
4. **Auth Middleware** - optionalAuth or requireAuth
5. **Route Handler** - Match route
6. **Controller** - Business logic
7. **Service** - Data operations
8. **Database** - MongoDB/Redis
9. **Response** - Send result

Summary

This document provides **complete visual maps** for all server endpoints including:

- All 28 endpoints** mapped in detail
- Every layer** - Middleware → Routes → Controllers → Services → Database
- All conditions** - Success and error paths
- Authentication flows** - optionalAuth vs requireAuth
- Database operations** - MongoDB and Redis interactions
- Request/Response examples** - Real JSON payloads
- Color-coded diagrams** - Easy visual parsing

Perfect for:

-  Understanding exact endpoint behavior
-  Debugging API issues

-  API documentation reference
-  Writing tests
-  Team onboarding