**Black Box API - URI Documentation (CRUD + Functional)**

## **CRUD URIs**

### **1. Get All Products**

* **HTTP Method:** GET
* **URI:** /api/products
* **Description:** Retrieves a list of all products.
* **Request Parameters:** None
* **JSON Response**

[

{"id": 1, "name": "Gaming Mouse", "price": 49.99},

{"id": 2, "name": "Mechanical Keyboard", "price": 89.99}

]

* **Status:** 200 OK
* **Json file:** Returns list of products

### **2. Get Product by ID**

* **HTTP Method:** GET
* **URI:** /api/products/{id}
* **Description:** Retrieves details of a specific product.
* **Request Parameters:** id of the product
* **JSON Response:**

{"id": 1, "name": "Gaming Mouse", "price": 49.99, "stock": 20}

* **Status:** 200 OK
* **Json file:** Returns product details

### **3. Create New Product**

* **HTTP Method:** POST
* **URI:** /api/products
* **Description:** Creates a new product entry.
* **Request Parameters:**

{"name": "Gaming Headset", "price": 59.99, "category": "Audio", "stock": 50}

* **JSON Response:**

{"message": "Product created successfully"}

* **Status:** 201 Created
* **Json file:** Confirmation message

### **4. Update Product**

* **HTTP Method:** PUT
* **URI:** /api/products/{id}
* **Description:** Updates product information.
* **Request Parameters:**

{"name": "Gaming Headset Pro", "price": 64.99}

* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **5. Delete Product**

* **HTTP Method:** DELETE
* **URI:** /api/products/{id}
* **Description:** Deletes a product.
* **Request Parameters:** Product ID
* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **6. Get All Categories**

* **HTTP Method:** GET
* **URI:** /api/categories
* **Description:** Retrieves all product categories.
* **Request Parameters:** None
* **JSON Response:**

["Audio", "Computers", "Accessories"]

* **Status:** 200 OK
* **Json file:** Returns categories list

### **7. Create New Category**

* **HTTP Method:** POST
* **URI:** /api/categories
* **Description:** Adds a new category.
* **Request Parameters:**

{"name": "Networking"}

* **JSON Response:**

{"message": "Category created successfully"}

* **Status:** 201 Created
* **Json file:** Confirmation message

### **8. Get All Brands**

* **HTTP Method:** GET
* **URI:** /api/brands
* **Description:** Retrieves a list of all brands.
* **Request Parameters:** None
* **JSON Response:**

["Logitech", "HyperX", "Razer"]

* **Status:** 200 OK
* **Json file:** List of brands

### **9. Create New Brand**

* **HTTP Method:** POST
* **URI:** /api/brands
* **Description:** Creates a new brand.
* **Request Parameters:**

{"name": "Corsair"}

* **JSON Response:**

{"message": "Brand added"}

* **Status:** 201 Created
* **Json file:** Confirmation message

### **10. Update Category**

* **HTTP Method:** PUT
* **URI:** /api/categories/{id}
* **Description:** Updates a category by ID.
* **Request Parameters:**

{"name": "Computer Accessories"}

* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **11. Get All Users**

* **HTTP Method:** GET
* **URI:** /api/users
* **Description:** Retrieves all registered users.
* **Request Parameters:** None
* **JSON Response:**

[

{"userId": 1, "firstName": "Juan", ”lastName”:”Pérez”, "email": "[juan@example.com](mailto:juan@example.com)", “password”: ”Juan123”, “userType”:”user”, “phoneNumber”: “0987858623”},

{"userId": 2, "firstName": "Ana", “lastName”:”López”, “ "email": "[ana@example.com](mailto:ana@example.com)", “password”: “Ana123”, “userType”:”user”, ”phoneNumber”: “0986523915” }

]

* **Status:** 200 OK
* **Json file:** List of users

### **12. Get User by ID**

* **HTTP Method:** GET
* **URI:** /api/users/{id}
* **Description:** Retrieves a user's details by ID.
* **Request Parameters:** User ID
* **JSON Response:**

{"userId": 1, "firstName": "Juan", ”lastName”:”Pérez”, "email": "[juan@example.com](mailto:juan@example.com)", “password”: ”Juan123”, “userType”:”user”, “phoneNumber”: “0987858623”}

* **Status:** 200 OK
* **Json file:** User data

### **13. Create New User**

* **HTTP Method:** POST
* **URI:** /api/users
* **Description:** Registers a new user.
* **Request Parameters:**

{"firstName": "Laura", “lastName”:”Vega”, "email": "[laura@example.com](mailto:laura@example.com)", "password": "securePass123", “phoneNumber”:”0658492597”}

* **JSON Response:**

{"message": "User created successfully"}

* **Status:** 201 Created
* **Json file:** Confirmation message

### **14. Update User**

* **HTTP Method:** PUT
* **URI:** /api/users/{id}
* **Description:** Updates user profile details.
* **Request Parameters:**

{"firstName": "Laura V. Vega"}

* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **15. Delete User**

* **HTTP Method:** DELETE
* **URI:** /api/users/{id}
* **Description:** Deletes a user account.
* **Request Parameters:** User ID
* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **16. Create New Order**

* **HTTP Method:** POST
* **URI:** /api/orders
* **Description:** Places a new order.
* **Request Parameters:**

{“orderId”: 1, "userId": 1, "orderDate": 05/06/2025, “total”: 125.00, “status”:”pending”}

* **JSON Response:**

{"message": "Order created successfully"}

* **Status:** 201 Created
* **Json file:** Confirmation

### **17. Get Order by ID**

* **HTTP Method:** GET
* **URI:** /api/orders/{id}
* **Description:** Retrieves order details.
* **Request Parameters:** Order ID
* **JSON Response:**

{"orderId": 5, "total": 129.99, "status": "Processing"}

* **Status:** 200 OK
* **Json file:** Order details

### **18. Delete Order**

* **HTTP Method:** DELETE
* **URI:** /api/orders/{id}
* **Description:** Deletes an order permanently.
* **Request Parameters:** Order ID
* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **19. Get All Payments**

* **HTTP Method:** GET
* **URI:** /api/payments
* **Description:** Lists all payment records.
* **Request Parameters:** None
* **JSON Response:**

[

{"paymentId": 1, “userId”; 1, “orderId": 1, “amount": 89.99, “paymentMethod”:”credit card”, paymentDate”: 01/04/2025},

{"paymentId": 3, “userId”; 5, “orderId": 2, “amount": 155.60, “paymentMethod”:”debit card”, paymentDate”: 12/05/2025}

]

* **Status:** 200 OK
* **Json file:** Payment list

### **20. Create New Payment**

* **HTTP Method:** POST
* **URI:** /api/payments
* **Description:** Adds a new payment entry.
* **Request Parameters:**

{ “paymentId”: 6, "userId": 1, "orderId": 5, "amount": 129.99, "paymentMethod": "creditcard", “paymentDate”: 26/03/2024}

* **JSON Response:**

{"message": "Payment successful"}

* **Status:** 201 Created
* **Json file:** Confirmation

**21. Get All Wishlists**

* **HTTP Method:** GET
* **URI:** /api/wishlists
* **Description:** Retrieves all wishlists.
* **Request Parameters:** None
* **JSON Response:**

{"wishlistId": 1, "userId": 3, "createdDate": "2024-05-10"}

* **Status:** 200 OK
* **Json file:** List of wishlists

**22. Get Wishlist by ID**

* **HTTP Method:** GET
* **URI:** /api/wishlists/{id}
* **Description:** Retrieves details of a specific wishlist.
* **Request Parameters:** id of the wishlist
* **JSON Response:**

{"wishlistId": 1, "userId": 3, "createdDate": "2024-05-10"}

* **Status:** 200 OK
* **Json file:** Wishlist details

**23. Create New Wishlist**

* **HTTP Method:** POST
* **URI:** /api/wishlists
* **Description**: Creates a new wishlist.
* **Request Parameters:**

{"userId": 3}

* **JSON Response:**

{"message": "Wishlist created successfully"}

* **Status:** 201 Created
* **Json file:** Confirmation message

**24. Delete Wishlist**

* **HTTP Method:** DELETE
* **URI**: /api/wishlists/{id}
* **Description:** Deletes a wishlist by ID.
* **Request Parameters:** id of the wishlist
* **JSON Response:**

{"message": "Wishlist deleted successfully"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**25. Add Product to Wishlist**

* **HTTP Method:** POST
* **URI:** /api/wishlists/{wishlistId}/products
* **Description:** Adds a product to a wishlist.
* **Request Parameters:**

{"productId": 10}

* **JSON Response:**

{"message": "Product added to wishlist"}

* **Status:** 201 Created
* **Json file:** Confirmation message

**26. Get Products in Wishlist**

* **HTTP Method:** GET
* **URI:** /api/wishlists/{wishlistId}/products
* **Description:** Retrieves all products in a wishlist.
* **Request Parameters:** wishlistId
* **JSON Response:**

[

{"productId": 1, "name": "Gaming Mouse"},

{"productId": 2, "name": "Gaming Headset"}

]

* **Status:** 200 OK
* **Json file:** List of products in wishlist

**27. Remove Product from Wishlist**

* **HTTP Method:** DELETE
* **URI:** /api/wishlists/{wishlistId}/product/{productId}
* **Description:** Removes a product from a wishlist.
* **Request Parameters:** wishlistId, productId
* **JSON Response:**

{"message": "Product removed from wishlist"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**28. Get Cart by User ID**

* **HTTP Method:** GET
* **URI:** /api/cart/user/{userId}
* **Description**: Retrieves a user's cart.
* **Request Parameters:** userId
* **JSON Response:**

{"cartId": 5, "userId": 3, "createdDate": "2024-06-01"}

* **Status:** 200 OK
* **Json file:** User cart

**29. Add Product to Cart**

* **HTTP Method:** POST
* **URI**: /api/cart/{cartId}/products
* **Description**: Adds a product to a cart.
* **Request Parameters:**

{ "productId": 2, "quantity": 1}

* **JSON Response:**

{"message": "Product added to cart"}

* **Status:** 201 Created
* **Json file:** Confirmation message

**30. Update Cart Product Quantity**

* **HTTP Method:** PUT
* **URI:** /api/cart/{cartId}/product/{productId}
* **Description:** Updates quantity of a product in the cart.
* **Request Parameters:**

{"quantity": 3}

* **JSON Response:**

{"message": "Quantity updated"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**31. Remove Product from Cart**

* **HTTP Method:** DELETE
* **URI:** /api/cart/{cartId}/product/{productId}
* **Description:** Removes a product from the cart.
* **Request Parameters:** cartId, productId
* **JSON Response:**

{"message": "Product removed from cart"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**32. Clear Cart**

* **HTTP Method:** DELETE
* **URI:** /api/cart/{cartId}/clear
* **Description:** Clears all products from a cart.
* **Request Parameters:** cartId
* **JSON Response:**

{"message": "Cart cleared"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**33. Get Shipping Info by Order ID**

* **HTTP Method:** GET
* **URI:** /api/shipping/order/{orderId}
* **Description:** Retrieves shipping info for an order.
* **Request Parameters:** orderId
* **JSON Response:**

{"shippingId": 8, "orderId": 4, "shippingAddress": "456 Elm Street"}

* **Status:** 200 OK
* **Json file:** Shipping info

**34. Create Shipping Record**

* **HTTP Method:** POST
* **URI:** /api/shipping
* **Description:** Creates a shipping record.
* **Request Parameters:**

{"orderId": 4, "shippingAddress": "456 Elm Street"}

* **JSON Response:**

{"message": "Shipping created successfully"}

* **Status:** 201 Created
* **Json file:** Confirmation message

**35. Update Shipping Address**

* **HTTP Method:** PUT
* **URI:** /api/shipping/{shippingId}
* **Description:** Updates shipping address.
* **Request Parameters:**

{"shippingAddress": "789 Pine Avenue"}

* **JSON Response:**

{"message": "Shipping address updated"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**36. Get All Products in an Order**

* **HTTP Method:** GET
* **URI:** /api/orders/{orderId}/products
* **Description:** Retrieves products in an order.
* **Request Parameters:** orderId
* **JSON Response:**

[

{"productId": 1, "name": "Mouse", "quantity": 2},

{"productId": 3, "name": "Headset", "quantity": 1}

]

* **Status:** 200 OK
* **Json file:** Order products

**37. Add Product to Order**

* **HTTP Method:** POST
* **URI:** /api/orders/{orderId}/products
* **Description:** Adds a product to an existing order.
* **Request Parameters:**

{"productId": 3, "quantity": 1}

* **JSON Response:**

{"message": "Product added to order"}

* **Status:** 201 Created
* **Json file:** Confirmation message

**38. Update Product Quantity in Order**

* **HTTP Method:** PUT
* **URI:** /api/orders/{orderId}/products/{productId}
* **Description:** Updates product quantity in an order.
* **Request Parameters:**

{"quantity": 4}

* **JSON Response:**

{"message": "Order quantity updated"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**39. Delete Product from Order**

* **HTTP Method:** DELETE
* **URI:** /api/orders/{orderId}/products/{productId}
* **Description:** Removes product from an order.
* **Request Parameters:** orderId, productId
* **JSON Response:**

{"message": "Product removed from order"}

* **Status:** 200 OK
* **Json file:** Confirmation message

**40. Get User Order History**

* **HTTP Method:** GET
* **URI:** /api/users/{userId}/orders
* **Description:** Retrieves all orders made by a user.
* **Request Parameters:** userId
* **JSON Response:**

[

{"orderId": 1, "orderDate": "2024-05-02", "total": 120.50},

{"orderId": 2, "orderDate": "2024-06-12", "total": 89.00}

]

* **Status:** 200 OK
* **Json file:** User orders

## **Functional (Service) URIs**

### **41. Search Products by Keyword**

* **HTTP Method:** GET
* **URI:** /api/products/search?keyword=mouse
* **Description:** Searches products by name or description.
* **Request Parameters:** keyword
* **JSON Response:**

[

{"id": 1, "name": "Gaming Mouse"},

{"id": 4, "name": "Wireless Mouse"}

]

* **Status:** 200 OK
* **Json file:** Matching products

### **42. Filter Products by Price**

* **HTTP Method:** GET
* **URI:** /api/products/filter?minPrice=20&maxPrice=100
* **Description:** Filters products within price range.
* **Request Parameters:** minPrice, maxPrice
* **JSON Response:**

{"id": 3, "name": "Gaming Keyboard", "price": 69.99}

* **Status:** 200 OK
* **Json file:** Filtered products

### **43. Get Top-Selling Products**

* **HTTP Method:** GET
* **URI:** /api/products/top-sellers
* **Description:** Retrieves top-selling products.
* **Request Parameters:** None
* **JSON Response:**

{"id": 5, "name": "Monitor 24''", "sales": 150}

* **Status:** 200 OK
* **Json file:** Top products

### **44. Get Low Stock Products**

* **HTTP Method:** GET
* **URI:** /api/products/low-stock
* **Description:** Lists products with low stock.
* **Request Parameters:** None
* **JSON Response:**

{"id": 6, "name": "HDMI Cable", "stock": 3}

* **Status:** 200 OK
* **Json file:** Low-stock list

### **45. Update Stock of a Product**

* **HTTP Method:** PUT
* **URI:** /api/products/{id}/stock
* **Description:** Updates only the stock quantity.
* **Request Parameters:**

{"stock": 15}

* **JSON Response:** None
* **Status:** 204 No Content
* **Json file:** No content

### **46. Get Discounted Products**

* **HTTP Method:** GET
* **URI:** /api/products/discounted
* **Description:** Lists products on discount.
* **Request Parameters:** None
* **JSON Response:**

{"id": 7, "name": "USB Hub", "discount": "10%"}

* **Status:** 200 OK
* **Json file:** Discounted products

### **47. Submit Product Rating**

* **HTTP Method:** POST
* **URI:** /api/products/{id}/rate
* **Description:** Submits a rating/review.
* **Request Parameters:**

{"rating": 5, "comment": "Excellent product!"}

* **JSON Response:**

{"message": "Review submitted"}

* **Status:** 201 Created
* **Json file:** Confirmation

### **48. Get Average Product Rating**

* **HTTP Method:** GET
* **URI:** /api/products/{id}/average-rating
* **Description:** Retrieves average rating of a product.
* **Request Parameters:** Product ID
* **JSON Response:**

{"averageRating": 4.5}

* **Status:** 200 OK
* **Json file:** Rating summary

### **49. Get Inventory Summary Report**

* **HTTP Method:** GET
* **URI:** /api/reports/inventory-summary
* **Description:** Provides an inventory summary.
* **Request Parameters:** None
* **JSON Response:**

{"totalProducts": 120, "outOfStock": 8, "categories": 10}

* **Status:** 200 OK
* **Json file:** Summary report

### **50. Compare Products by ID**

* **HTTP Method:** GET
* **URI:** /api/products/compare?ids=1,2
* **Description:** Compares selected products by features.
* **Request Parameters:** ids (comma-separated)
* **JSON Response:**

[

{"id": 1, "name": "Gaming Mouse", "features": ["RGB", "Wireless"]},

{"id": 2, "name": "Ergonomic Mouse", "features": ["Wired", "Silent Click"]}

]

* **Status:** 200 OK
* **Json file:** Comparison list

**51. Update Payment Status**

* **HTTP Method:** PUT
* **URI:** /api/payments/:paymentId/status
* **Description:** Updates the status of a specific payment.
* **Request Parameters (Body):**

{

"status": "confirmed"

}

* **JSON Response:**

{

"message": "Payment status updated to confirmed."

}

* **Status:** 200 OK
* **Json file:** Confirmation message of status update

### **52. Checkout Cart**

* **HTTP Method:** POST
* **URI:** /api/cart/checkout
* **Description:** Processes payment and places order from cart.
* **Request Parameters:** Cart ID
* **JSON Response:**

{"message": "Checkout complete"}

* **Status:** 200 OK
* **Json file:** Confirmation

### **53. Get Products By Price Range**

* **HTTP Method:** GET
* **URI:** /products/price/:min/:max
* **Description:** Retrieves a list of products based on a range price.
* **Request Parameters:** price
* **JSON Response:**

{

{"productId": 10, "name": "Gaming Mouse Pad", “description”: “decorated gaming pad”, “brand”: “Logitech”, “price”: 20, “stock”: 6, “categoryId”: 1},

{"productId": 12, "name": "Ergonomic Wrist Rest", “description”: “decorated gaming pad with a wrist rest”, “brand”: “Logitech”, “price”: 20, “stock”: 6, “categoryId”: 1}

}

* **Status:** 200 OK
* **Json file:** List of recommended products

### **54. Cancel Order**

* **HTTP Method:** PUT
* **URI:** /api/orders/{id}/cancel
* **Description:** Cancels an existing order.
* **Request Parameters:** Order ID
* **JSON Response:**

{"message": "Order cancelled"}

* **Status:** 200 OK
* **Json file:** Confirmation

### **55. Send Order Delivery Message via WhatsApp**

* **HTTP Method:** POST
* **URI:** /api/notifications/whatsapp/send
* **Description:** Sends a WhatsApp message to the client with delivery details of their order.
* **Request Parameters:**

{"orderId": 18, “userId”: 14, "phoneNumber": "0987654321", "message": "Hola Juan, tu pedido #18 está listo. Coordina tu entrega respondiendo a este mensaje."}

* **JSON Response:**

{"message": "WhatsApp notification sent successfully"}

* **Status:** 200 OK
* **Json file:** Delivery message confirmation

### **56. Get Order History by user Id**

* **HTTP Method:** GET
* **URI:** /orders/user/:userId
* **Description:** Lists previous orders for a user.
* **Request Parameters:** userId
* **JSON Response:**

[

{"orderId": 4, "total": 99.99, “status”: “pending”},

{"orderId": 5, "total": 129.99, “status”: “paid”}

]

* **Status:** 200 OK
* **Json file:** Order history

### **57. Check Low Stock**

* **HTTP Method:** GET
* **URI:** /api/products/low-stock
* **Description:** Lists products with low inventory.
* **Request Parameters:** None
* **JSON Response:**

{"productId": 6, "name": "HDMI Cable", "stock": 3}

* **Status:** 200 OK
* **Json file:** Alert list

### **58. Notify Product Back in Stock to Wishlist User**

* **HTTP Method:** POST
* **URI:** /api/wishlist/notifyAvailability
* **Description:** Notifies users via WhatsApp or email when a product in their wishlist is back in stock.
* **Request Parameters:**

{"productId": 17}

* **JSON Response:**

{"message": "Users notified about product availability"}

* **Status:** 200 OK
* **Json file:** Notification result

### **59. Get Total Cart Price**

* **HTTP Method:** GET
* **URI:** /api/cart/total?cartId=3
* **Description:** Calculates total price of items in cart.
* **Request Parameters:** cartId
* **JSON Response:**

{"total": 189.98}

* **Status:** 200 OK
* **Json file:** Cart summary

### **60. Validate Payment**

* **HTTP Method:** POST
* **URI:** /api/payments/validate
* **Description:** Validates if a payment was completed successfully.
* **Request Parameters:**

{"paymentId": 2}

* **JSON Response:**

{"valid": true}

* **Status:** 200 OK
* **Json file:** Validation result