

NARA DB API Documentation

Product API Documentation

Base URL: `/api/products`

1. Get All Products

- **Endpoint:** `GET /api/products`
- **Description:** Retrieves a list of all products.
- **Response:**
 - **200 OK:** Returns the list of products.
 - **204 No Content:** If no products are found.

2. Get Product by ID

- **Endpoint:** `GET /api/products/{id}`
- **Description:** Retrieves a specific product by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the product.
- **Response:**
 - **200 OK:** Returns the product data.
 - **404 Not Found:** If the product with the given ID is not found.

3. Get Product ID by Product Name

- **Endpoint:** `GET /api/products/id`
- **Description:** Retrieves the ID of a product by its name.
- **Query Parameters:**
 - `productName` (String): The name of the product.
- **Response:**
 - **200 OK:** Returns the product ID.
 - **404 Not Found:** If the product name is not found.

4. Create Product

- **Endpoint:** `POST /api/products`
- **Description:** Creates a new product.
- **Request Body:**
 - A JSON object representing the product.
- **Response:**
 - **201 Created:** Returns the saved product.

- **500 Internal Server Error:** If an error occurs while creating the product.

5. Update Product

- **Endpoint:** `PUT /api/products/{id}`
- **Description:** Updates an existing product by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the product to update.
- **Request Body:**
 - A JSON object representing the updated product.
- **Response:**
 - **200 OK:** Returns the updated product.
 - **500 Internal Server Error:** If an error occurs while updating the product.

6. Delete Product

- **Endpoint:** `DELETE /api/products/{id}`
- **Description:** Deletes a product by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the product to delete.
- **Response:**
 - **200 OK:** If the product is successfully deleted.
 - **404 Not Found:** If the product with the given ID is not found.
 - **500 Internal Server Error:** If an error occurs while deleting the product.

7. Add Material to Product

- **Endpoint:** `POST /api/products/{productId}/materials/{materialId}`
- **Description:** Adds a material to a product.
- **Path Parameters:**
 - `productId` (int): The ID of the product.
 - `materialId` (int): The ID of the material to add.
- **Response:**
 - **200 OK:** If the material is successfully added to the product.
 - **404 Not Found:** If the product or material is not found.
 - **500 Internal Server Error:** If an error occurs while adding the material.

8. Add Test to Product

- **Endpoint:** `POST /api/products/{productId}/tests/{testId}`
- **Description:** Adds a test to a product.
- **Path Parameters:**
 - `productId` (int): The ID of the product.
 - `testId` (int): The ID of the test to add.

- **Response:**
 - **200 OK:** If the test is successfully added to the product.
 - **400 Bad Request:** If the product or test is not found.
 - **500 Internal Server Error:** If an error occurs while adding the test.

9. Remove Material from Product

- **Endpoint:** `DELETE /api/products/{productId}/materials/{materialId}`
- **Description:** Removes a material from a product.
- **Path Parameters:**
 - `productId` (int): The ID of the product.
 - `materialId` (int): The ID of the material to remove.
- **Response:**
 - **200 OK:** If the material is successfully removed from the product.
 - **404 Not Found:** If the product or material is not found.
 - **500 Internal Server Error:** If an error occurs while removing the material.

10. Remove Test from Product

- **Endpoint:** `DELETE /api/products/{productId}/tests/{testId}`
- **Description:** Removes a test from a product.
- **Path Parameters:**
 - `productId` (int): The ID of the product.
 - `testId` (int): The ID of the test to remove.
- **Response:**
 - **200 OK:** If the test is successfully removed from the product.
 - **404 Not Found:** If the product or test is not found.
 - **500 Internal Server Error:** If an error occurs while removing the test.

Material API Documentation

Base URL: `/api/materials`

1. Get All Materials

- **Endpoint:** `GET /api/materials`
- **Description:** Retrieves a list of all materials.
- **Response:**
 - **200 OK:** Returns the list of materials.
 - **204 No Content:** If no materials are found.

2. Get Material by ID

- **Endpoint:** `GET /api/materials/{id}`
- **Description:** Retrieves a specific material by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the material.
- **Response:**
 - **200 OK:** Returns the material data.
 - **404 Not Found:** If the material with the given ID is not found.

3. Get Material ID by Material Name

- **Endpoint:** `GET /api/materials/id`
- **Description:** Retrieves the ID of a material by its name.
- **Query Parameters:**
 - `materialName` (String): The name of the material.
- **Response:**
 - **200 OK:** Returns the material ID.
 - **404 Not Found:** If the material name is not found.

4. Create Material

- **Endpoint:** `POST /api/materials`
- **Description:** Creates a new material.
- **Request Body:**
 - A JSON object representing the material.
- **Response:**
 - **201 Created:** Returns the saved material.
 - **500 Internal Server Error:** If an error occurs while creating the material.

5. Update Material

- **Endpoint:** `PUT /api/materials/{id}`
- **Description:** Updates an existing material by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the material to update.
- **Request Body:**
 - A JSON object representing the updated material.
- **Response:**
 - **200 OK:** Returns the updated material.
 - **500 Internal Server Error:** If an error occurs while updating the material.

6. Delete Material

- **Endpoint:** `DELETE /api/materials/{id}`
- **Description:** Deletes a material by its ID.

- **Path Parameters:**
 - `id` (int): The ID of the material to delete.
 - **Response:**
 - **200 OK:** If the material is successfully deleted.
 - **404 Not Found:** If the material with the given ID is not found.
 - **500 Internal Server Error:** If an error occurs while deleting the material.
-

Instrument API Documentation

Base URL: `/api/instruments`

1. Get All Instruments

- **Endpoint:** `GET /api/instruments`
- **Description:** Retrieves a list of all instruments.
- **Response:**
 - **200 OK:** Returns the list of instruments.
 - **204 No Content:** If no instruments are found.

2. Get Instrument by ID

- **Endpoint:** `GET /api/instruments/{id}`
- **Description:** Retrieves a specific instrument by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the instrument.
- **Response:**
 - **200 OK:** Returns the instrument data.
 - **404 Not Found:** If the instrument with the given ID is not found.

3. Get Instrument ID by Instrument Name

- **Endpoint:** `GET /api/instruments/id`
- **Description:** Retrieves the ID of an instrument by its name.
- **Query Parameters:**
 - `instrumentName` (String): The name of the instrument.
- **Response:**
 - **200 OK:** Returns the instrument ID.
 - **404 Not Found:** If the instrument name is not found.

4. Create Instrument

- **Endpoint:** `POST /api/instruments`
- **Description:** Creates a new instrument.

- **Request Body:**
 - A JSON object representing the instrument.
- **Response:**
 - **201 Created:** Returns the saved instrument.
 - **500 Internal Server Error:** If an error occurs while creating the instrument.

5. Update Instrument

- **Endpoint:** `PUT /api/instruments/{id}`
- **Description:** Updates an existing instrument by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the instrument to update.
- **Request Body:**
 - A JSON object representing the updated instrument.
- **Response:**
 - **200 OK:** Returns the updated instrument.
 - **500 Internal Server Error:** If an error occurs while updating the instrument.

6. Delete Instrument

- **Endpoint:** `DELETE /api/instruments/{id}`
- **Description:** Deletes an instrument by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the instrument to delete.
- **Response:**
 - **200 OK:** If the instrument is successfully deleted.
 - **404 Not Found:** If the instrument with the given ID is not found.
 - **500 Internal Server Error:** If an error occurs while deleting the instrument.

Test API Documentation

Base URL: `/api/tests`

1. Get All Tests

- **Endpoint:** `GET /api/tests`
- **Description:** Retrieves a list of all tests.
- **Response:**
 - **200 OK:** Returns the list of tests.
 - **204 No Content:** If no tests are found.

2. Get Test by ID

- **Endpoint:** `GET /api/tests/{id}`
- **Description:** Retrieves a specific test by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the test.
- **Response:**
 - **200 OK:** Returns the test data.
 - **404 Not Found:** If the test with the given ID is not found.

3. Get Test ID by Test Name

- **Endpoint:** `GET /api/tests/id`
- **Description:** Retrieves the ID of a test by its name.
- **Query Parameters:**
 - `testName` (String): The name of the test.
- **Response:**
 - **200 OK:** Returns the test ID.
 - **404 Not Found:** If the test name is not found.

4. Create Test

- **Endpoint:** `POST /api/tests`
- **Description:** Creates a new test.
- **Request Body:**
 - A JSON object representing the test.
- **Response:**
 - **201 Created:** Returns the saved test.
 - **500 Internal Server Error:** If an error occurs while creating the test.

5. Update Test

- **Endpoint:** `PUT /api/tests/{id}`
- **Description:** Updates an existing test by its ID.
- **Path Parameters:**
 - `id` (int): The ID of the test to update.
- **Request Body:**
 - A JSON object representing the updated test.
- **Response:**
 - **200 OK:** Returns the updated test.
 - **500 Internal Server Error:** If an error occurs while updating the test.

6. Delete Test

- **Endpoint:** `DELETE /api/tests/{id}`
- **Description:** Deletes a test by its ID.

- **Path Parameters:**
 - `id` (int): The ID of the test to delete.
- **Response:**
 - **200 OK:** If the test is successfully deleted.
 - **404 Not Found:** If the test with the given ID is not found.
 - **500 Internal Server Error:** If an error occurs while deleting the test.