**API & Database Test Cases for /articles?include=author**

## ****API Test Cases****

### ****Positive API Test Cases****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC #** | **Test Scenario** | **Test Data** | **Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Explanation** |
| **1** | Verify response contains only allowed fields for articles | { "title": "JSON:API paints my bikeshed!", "body": "The shortest article. Ever." } | Send a GET /articles?include=author request and check attributes | title, body, and author should be present, no extra fields | Extra fields created and updated present | ❌ Fail | Extra fields created and updated are included, violating expected attributes. |
| **2** | Verify response contains only name field for people | { "name": "John" } | Send a GET /articles?include=author request and check included objects | Only name should be present, no extra fields like age or gender | Extra fields age and gender present | ❌ Fail | age and gender should not be in included, only name is expected. |
| **3** | Verify response when author relationship is present | { "title": "Sample Title", "body": "Sample Body", "author": { "id": "42", "type": "people" } } | Send a GET /articles?include=author request with author relationship present | Response should include author details under relationships | Author relationship present | ✅ Pass | The response correctly includes author under relationships. |
| **4** | Verify response when author relationship is missing | { "title": "Sample Title", "body": "Sample Body" } | Send a GET /articles request without include=author | Articles should be returned without author field, no errors | No author relationship present | ✅ Pass | The response correctly excludes the author when not requested. |
| **5** | Verify API supports multiple authors if allowed | { "title": "Multi-author Article", "body": "Content", "authors": [{"id": "42", "name": "John"}, {"id": "43", "name": "Jane"}] } | Send a GET /articles?include=author request with multiple authors per article | API should return multiple authors or 400 Bad Request | TBD | TBD | Needs execution. |

### ****Negative API Test Cases****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC #** | **Test Scenario** | **Test Data** | **Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Explanation** |
| **6** | Verify response fails when extra fields are present in attributes | { "title": "Sample", "body": "Text", "created": "2025-02-12T14:56:29.000Z" } | Send a GET /articles?include=author request and check attributes | API should return 400 Bad Request or sanitize extra fields | Extra fields included in response | ❌ Fail | Extra fields (created, updated) should not be returned. |
| **7** | Verify response fails when extra fields are present in included | { "name": "John", "age": 40, "gender": "male" } | Send a GET /articles?include=author request and check included section | API should return 400 Bad Request or remove extra fields | Extra fields included in response | ❌ Fail | included contains extra fields (age, gender) violating expected schema. |
| **8** | Verify response format follows JSON:API specification | { "data": [{"type": "articles", "id": "1"}] } | Send a GET /articles?include=author request and validate format | API should return 400 Bad Request if format is incorrect | Extra fields present but JSON:API format intact | ✅ Pass | Response structure follows JSON:API, but extra fields exist. |
| **9** | Verify response when ?include= is given an invalid value (e.g., number) | ?include=123 | Send a GET /articles?include=123 request | API should return 400 Bad Request | TBD | TBD | Needs execution. |
| **10** | Verify response when id mismatch occurs | { "relationships": { "author": { "data": { "id": "99", "type": "people" } } }, "included": [{ "id": "42", "type": "people" }] } | Change id in included to not match relationships.author.data | API should return 400 Bad Request | TBD | TBD | Needs execution. |
| **11** | Verify response when multiple articles have the same author | { "articles": [{ "title": "A1", "author": "42" }, { "title": "A2", "author": "42" }] } | Send a GET /articles?include=author request with duplicate authors | included should contain unique author entries | TBD | TBD | Needs execution. |

## ****Database Test Cases****

### ****Positive Database Test Cases****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC #** | **Test Scenario** | **Test Data** | **Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Explanation** |
| **12** | Verify API returns empty array when no articles exist | Database has no articles | Send a GET /articles?include=author request when DB is empty | API should return 200 OK with "data": [] | Empty response with data: [] | ✅ Pass | API correctly returns an empty array when no articles exist. |

### ****Negative Database Test Cases****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC #** | **Test Scenario** | **Test Data** | **Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Explanation** |
| **13** | Verify response when requested author ID does not exist | { "id": "999", "type": "people" } | Request an author ID that does not exist in DB | API should return 404 Not Found | TBD | TBD | Needs execution. |
| **14** | Verify database consistency when deleting an author | { "id": "42", "type": "people" } | Delete an author from DB and check article relationships | Articles should handle missing author gracefully | TBD | TBD | Needs execution. |