📓

Test record document

Create item

SN	Descriprion	Request type	API	Request body	Expected response status code	Expected response body	Actual response status code	Actual response body
1	Create an item when not logged in	POST	/items	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	401	`{ “message”: string }`	401	`{ “message”: “Unauthenticated” }`
2	Create an item when logged in as a user	POST	/items	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	201	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	201	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`
3	Create an item when logged in as an admin	POST	/items	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	201	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	201	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`
4	Create an item by giving invalid input	POST	/items	`{ “name”: “n”, “description”: ”d” “price”: -100 }`	422	`{ “message”: string , "errors": [ "name": [ string ] , "description": [ string ] , "price": [ string ] , ] }`	422	`{ message: 'The name field must be at least 4 characters. (and 2 more errors)', errors: { name: [ 'The name field must be at least 4 characters.' ], description: [ 'The description field must be at least 10 characters.' ], price: [ 'The price field must be at least 0.01.' ] } }`

Fetch all items

SN	Descriprion	Request type	API	Request body	Expected response status code	Expected response body	Actual response status code	Actual response body
1	Get all items when not logged in	GET	/items		200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`	200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`
2	Get all items when logged in as a user	GET	/items		200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`	200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`
3	Get all items when logged in as an admin	GET	/items		200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`	200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`
4	The structure of the items that were returned	GET	/items		200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`	200	`[ { “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 } ]`

Fetch single item

SN	Descriprion	Request type	API	Request param	Expected response status code	Expected response body	Actual response status code	Actual response body
1	Get an item when not logged in	GET	/items/`{uuid}`		200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`
2	Get an item when logged in as a user	GET	/items/`{uuid}`		200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`

SN	Descripion	Request type	API	Request param	Expected response status code	Expected response body	Actual response status code	Actual response body
3	Get all items when logged in as an admin	GET	/items/`{uuid}`		200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`
4	The structure of the items that were returned	GET	/items/`{uuid}`		200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`

Update an item

SN	Descripion	Request type	API	Request body	Expected response status code	Expected response body	Actual response status code	Actual response body
1	Update an item when not logged in	PUT	/items/`{uuid}`	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	401	`{ “message”: string }`	401	`{ “message”: “Unauthenticated” }`
2	Update an item that was created by another user when logged in as a user	PUT	/items/`{uuid}`	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	403	`{ “message”: string }`	403	`{ “message”: "You are not authorized to update this it" }`
3	Update an item that was created by a user when logged in as the same user	PUT	/items/`{uuid}`	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`
4	Update an item that was created by any user when logged in as an admin	PUT	/items/`{uuid}`	`{ “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`
5	Update an item by giving invalid input	PUT	/items/`{uuid}`	`{ “name”: “n”, “description”: ”d” “price”: -100 }`	422	`{ “message”: string , "errors": [ "name": [ string ] , "description": [ string ] , "price": [ string ] , ] }`	422	`{ message: 'The name field must be at least 4 characters. (and 2 more errors)', errors: { name: [ 'The name field must be at least 4 characters.' ], description: [ 'The description field must be at least 10 characters.' ], price: [ 'The price field must be at least 0.01.' ] } }`

Delete an item

SN	Descripion	Request type	API	Request body	Expected response status code	Expected response body	Actual response status code	Actual response body
1	Delete an item when not logged in	DELETE	/items/`{uuid}`		401	`{ “message”: string }`	401	`{ “message”: “Unauthenticated” }`
2	Delete an item that was created by a user when logged in as the same user	DELETE	/items/`{uuid}`		403	`{ “message”: string }`	403	`{ “message”: "You are not authorized to update this it" }`
3	Delete an item that was created by any user when logged in as a admin	DELETE	/items/`{uuid}`		200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`	200	`{ “id”: uuid , “userId”: uuid , “name”: “test item”, “description”: ”test description” “price”: 100 }`