

Test Case	Endpoint	Details	Owner	Automated	Date	Status	Notes
TC1	GET /Books	Validate each book matches schema	Lucas Astor	Yes	9/19/2025	Validated OK	The API work as expected returning a 200 status and the books retrieved matches the schema
TC2	GET /Books	Validate receiving 200 status and an array of books is returned	Lucas Astor	Yes	9/19/2025	Validated OK	The API work as expected returning a 200 and an array of books is returned
TC3	GET /Books	Validate receiving 405 status when using unsupported method (e.g. OPTIONS method)	Lucas Astor	Yes	9/19/2025	Validated OK	The API work as expected returning 405 status when a non-supported method is used
TC4	GET /Books/{id}	Validate that retrieving an existing ID returns correct book	Lucas Astor	Yes	9/19/2025	Validated OK	The API works as expected returning a 200 status and the information about the book retrieved.
TC5	GET /Books/{id}	Validate receiving 404 status when retrieving a non existing ID	Lucas Astor	Yes	9/19/2025	Validated OK	The API work as expected and returns a 404 status when trying to get a non existing ID (e.g. -1)
TC6	POST /Books	Validate receiving a 201 status when creating a book with all the fields needed - A book is returned	Lucas Astor	Yes	9/19/2025	Validated OK	The API works as expected returning a 200 status and the information about the book created
TC7	POST /Books	Validate receiving a 201 status when creating a book not including the nullable fields	Lucas Astor	Yes	9/19/2025	Validated OK	The API work as expected returning a 200 status when creating a book not including the nullable fields
TC8	POST /Books	Validate receiving a 400 status when creating a book with no request body	Lucas Astor	Yes	9/19/2025	Validated OK	The API work as expected returning a 400 status when creating a book not including even the curly braces, so no JSON was sent as a body.
TC9	POST /Books	Validate receiving a 400 status when using an invalid date format	Lucas Astor	Yes	9/22/2025	Validated OK	The API work as expected. Returns a 400 status and exception in the correct field.
TC10	POST /Books	Validate receiving a 400 status when adding an "extra" property inside the book body	Lucas Astor	Yes	9/22/2025	Not working	I assume that the API should control that the POST request body must be correctly formed
TC11	PUT /Books/{id}	Validate receiving a 200 when updating an existing book	Lucas Astor	Yes	9/22/2025	Validated OK	The API work as expected. Returns a 200 status and the updated info in the body of the response
TC12	PUT /Books/{id}	Validate receiving a 400 when updating a book using an invalid type (e.g. Invalid date)	Lucas Astor	Yes	9/22/2025	Validated OK	The API work as expected. Returns a 400 status and exception in the correct field.
TC13	DELETE /Books/{id}	Validate receiving a 200 status when deleting a book	Lucas Astor	Yes	9/22/2025	Validated OK	The API work as expected. Returns a 200 status when deleting an existing book. The correct way to test this should be also trying to GET the DELETED book and making sure it's not present, but knowing that this is a testing API the expected result is to be able to find the DELETED book when trying to GET it.
TC14	DELETE /Books/{id}	Validate receiving a 400 status when deleting an invalid id (e.g. negative number)	Lucas Astor	Yes	9/22/2025	Not working	I assume that the API should control that the ID to delete must be a positive integer (or the type of ID the API uses)
TC15	DELETE /Books/{id}	Validate receiving a 400 status when deleting an integer with more than 10 digits	Lucas Astor	Yes	9/22/2025	Validated OK	The API work as expected and return a 400 status when trying to delete a book with a >10 digits as an ID