

API Documentation - Chris Bookstore

Products

GET /products – get all books	
Parameters	none
Responses	content type – application/json
Code – 200	<p>description – success example –</p> <pre>{ "data": { "books": [{ "author": "Ernest Hemingway", "book_id": 987, "isbn": "0684801221", "price": 20.0, "publisher": "Scribner Paperback Fiction", "stock": 500, "title": "The Old Man and The Sea" }, { "author": "George Orwell", "book_id": 988, "isbn": "9780151010264", "price": 80.0, "publisher": "Harcourt", "stock": 100, "title": "Animal Farm and 1984" }, { "author": "Herman Melville", "book_id": 989, "isbn": "1101100435", "price": 100.0, "publisher": "Penguin Group", "stock": 65, "title": "Moby-Dick or The Whale" }, { "author": "Charles Dickens", "book_id": 990, "isbn": "1411433238",</pre>

	<pre> "price": 70.0, "publisher": "Barnes & Noble Classics", "stock": 45, "title": "A Tale of Two Cities" }, { "author": "Tolkien J.R.R.", "book_id": 991, "isbn": "9780547928210", "price": 70.0, "publisher": "Allen & Unwin", "stock": 35, "title": "The Fellowship of the Ring" }, { "author": "Dumas, Alexandre", "book_id": 992, "isbn": "9780140449266", "price": 85.0, "publisher": "Penguin Classics", "stock": 95, "title": "The Count of Monte Cristo" }] } </pre>
Code – 404	<p>description – no books</p> <p>example –</p> <pre> { "message": "There are no books." } </pre>
GET /products/<int:book_id> – find book by ID	
Parameters	
Name – book_id (int, <i>path</i>)	<p>description – ID of book to return</p> <p>example – 987</p>
Responses	content type – application/json
Code – 200	<p>description – success</p> <p>example –</p> <pre> { "data": { "author": "Ernest Hemingway", "book_id": 987, "isbn": "0684801221", </pre>

	<pre> "price": 20.0, "publisher": "Scribner Paperback Fiction", "stock": 500, "title": "The Old Man and The Sea" } } </pre>
Code – 404	<p>description – book not found</p> <p>example –</p> <pre> { "message": "Book not found." } </pre>
POST /products – create book	
Parameters	
Name – body (object, <i>body</i>)	<p>description – book to create</p> <p>example –</p> <pre> { "author": "David Kopec", "isbn": "1617295981", "price": 100.0, "publisher": "Manning Publications", "stock": 25, "title": "Classic Computer Science Problems in Python" } </pre>
Responses	content type – application/json
Code – 201	<p>description – success</p> <p>example –</p> <pre> { "data": { "author": "David Kopec", "book_id": 1013, "isbn": "1617295981", "price": 100.0, "publisher": "Manning Publications", "stock": 25, "title": "Classic Computer Science Problems in Python" } } </pre>
Code – 500	<p>description – error</p> <p>example –</p> <pre> { "message": "An error occurred creating the </pre>

	<pre>book.", "error": <specific details of the error> }</pre>
PUT /products/<int:book_id> – replace book	
Parameters	
Name – book_id (int, <i>path</i>)	description – ID of book to replace example – 987
Name – body (object, <i>body</i>)	description – book to replace example – <pre>{ "author": "David Kopec", "isbn": "1617295981", "price": 100.0, "publisher": "Manning Publications", "stock": 25, "title": "Classic Computer Science Problems in Python" }</pre>
Responses	content type – application/json
Code – 200	description – success example – <pre>{ "data": { "author": "David Kopec", "book_id": 987, "isbn": "1617295981", "price": 100.0, "publisher": "Manning Publications", "stock": 25, "title": "Classic Computer Science Problems in Python" } }</pre>
Code – 404	description – book not found example – <pre>{ "message": "Book not found." }</pre>
Code – 500	description – error example – <pre>{ "message": "An error occurred replacing the</pre>

	<pre>book.", "error": <specific details of the error> }</pre>
PATCH /products/<int:book_id> – update book	
Parameters	
Name – book_id (int, <i>path</i>)	description – ID of book to update example – 988
Name – body (object, <i>body</i>)	description – values of book to update example – <pre>{ "price": 70.0, "stock": 120, }</pre>
Responses	content type – application/json
Code – 200	description – success example – <pre>{ "data": { "author": "George Orwell", "book_id": 988, "isbn": "9780151010264", "price": 70.0, "publisher": "Harcourt", "stock": 120, "title": "Animal Farm and 1984" } }</pre>
Code – 404	description – book not found example – <pre>{ "message": "Book not found." }</pre>
Code – 500	description – error example – <pre>{ "message": "An error occurred updating the book.", "error": <specific details of the error> }</pre>
PATCH /products/<int:book_id> – reserve stock	
Parameters	

Name – book_id (int, <i>path</i>)	description – ID of book to update example – 988
Name – body (object, <i>body</i>)	description – amount of stock to reserve example – <pre>{ "reserve": 35 }</pre>
Responses	content type – application/json
Code – 200	description – success example – <pre>{ "data": { "author": "George Orwell", "book_id": 988, "isbn": "9780151010264", "price": 80.0, "publisher": "Harcourt", "stock": 65, "title": "Animal Farm and 1984" } }</pre>
Code – 404	description – book not found example – <pre>{ "message": "Book not found." }</pre>
Code – 500	description – error example – <pre>{ "message": "An error occurred updating the book.", "error": "Insufficient books in stock." }</pre>
DELETE /products/<int:book_id> – delete book	
Parameters	
Name – book_id (int, <i>path</i>)	description – ID of book to delete example – 990
Responses	content type – application/json
Code – 200	description – success example – <pre>{</pre>

	<pre> "data": { "book_id": 990 } } </pre>
Code – 404	description – book not found example – <pre> { "message": "Book not found." } </pre>
Code – 500	description – error example – <pre> { "message": "An error occurred deleting the book.", "error": <specific details of the error> } </pre>

Orders

GET /orders – get all orders	
Parameters	none
Response	content type – application/json
Code – 200	description – success example – <pre> [{ "id": 3, "userId": 1, "datetime": "some datetime", "totalPrice": 100, "details": [{ "productId": 1, "quantity": 10, "bookTitle": null, </pre>

	<pre> "price": 10 }, { "productId": 1, "quantity": 5, "bookTitle": null, "price": 10 }] },] </pre>
GET /orders/<int:order_id> – find order by ID	
Parameters	name – order_id(int) description – ID of order to return
Response	content type – application/json
Code – 200	description – success example – <pre> { "id": 90, "userId": 110, "datetime": "2021-11-10 05:40:58.401360", "totalPrice": 250, "details": [{ "productId": 976, "quantity": 1, "bookTitle": "Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib", "price": 50 }, </pre>

	<pre> { "productId": 977, "quantity": 1, "bookTitle": "Classic Computer Science Problems in Python", "price": 100 }, { "productId": 978, "quantity": 1, "bookTitle": "Help Your Kids with Computer Science: A Unique Visual Step-by-Step Guide to Computers, Coding, and Communication", "price": 100 }] } </pre>
Code – 404	<p>description – order not found</p> <p>example –</p> <pre> { "timestamp": "2021-11-10T05:45:29.713+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/orders/9000" } </pre>
POST /orders – create order	
Parameters	
Name	Description
Body object (body)	Order to create

	<p>Example Value</p> <pre>{ "userId": 110, "datetime": "2021-11-10 05:18:02.162803", "totalPrice": 100, "details": [{ "productId": 978, "quantity": 1, "bookTitle": "Help Your Kids with Computer Science: A Unique Visual Step-by-Step Guide to Computers, Coding, and Communication", "price": 100 }] }</pre>
Response	content type – application/json
Code – 200	<p>description – success example –</p> <pre>{ "id": 96, "userId": 110, "datetime": "2021-11-10 05:18:02.162803", "totalPrice": 100, "details": [{ "productId": 978, "quantity": 1, "bookTitle": "Help Your Kids with Computer Science: A Unique Visual Step-by-Step Guide to Computers, Coding, and Communication",</pre>

	<pre> "price": 100 }] } </pre>
DELETE /orders/<int:order_id> – delete order by ID	
Parameters	name – order_id(int) description – ID of order to delete
Response	
Code – 200	description – success
Code – 404	description – order not found example – <pre> { "timestamp": "2021-11-10T05:45:29.713+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/orders/9000" } </pre>
GET /ordersByUserId/<int:user_id> – get orders by user ID	
Parameters	name – user_id(int) description – ID of user to retrieve orders
Response	
Code – 200	description – success <pre> [{ "id": 3, "userId": 1, "datetime": "some datetime", "totalPrice": 100, "details": [</pre>

	<pre> { "productId": 1, "quantity": 10, "bookTitle": null, "price": 10 }, { "productId": 1, "quantity": 5, "bookTitle": null, "price": 10 }] }]</pre>
--	---

Users

GET /users – get all users	
Parameters	none
Response	content type – application/json
Code – 200	<p>description – success example –</p> <pre> [{ "id": 109, "name": "user1", "phone": null, "email": "wjtay1998@gmail.com", "password": null }]</pre>

]
GET /users/<int:user_id> – find user by ID	
Parameters	name – user_id(int) description – ID of userto return
Response	content type – application/json
Code – 200	description – success example – <pre>{ "id": 109, "name": "user1", "phone": null, "email": "wjtay1998@gmail.com", "password": null }</pre>
Code – 404	description – user not found example – <pre>{ "timestamp": "2021-11-10T05:45:29.713+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/users/9000" }</pre>
POST /user – create user	
Parameters	
Name	Description
Body object (body)	User to create Example Value <pre>{ "name": "user1", "phone": null,</pre>

	<pre> "email": "wjtay1998@gmail.com", "password": null } </pre>
Response	content type – application/json
Code – 200	description – success example – <pre> { "id": 109, "name": "user1", "phone": null, "email": "wjtay1998@gmail.com", "password": null } </pre>
DELETE /users/<int:user_id> – delete user by ID	
Parameters	name – user_id(int) description – ID of user to delete
Response	
Code – 200	description – success
Code – 404	description – user not found example – <pre> { "timestamp": "2021-11-10T05:45:29.713+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/users/9000" } </pre>
GET /users/find/<String:name> – get users by name	
Parameters	name – name (String) description – name of user to be retrieved

Response	
Code – 200	description – success <pre>{ "id": 109, "name": "user1", "phone": null, "email": "wjtay1998@gmail.com", "password": null }</pre>
Code – 404	description – user not found example – <pre>{ "timestamp": "2021-11-10T05:45:29.713+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/users/9000" }</pre>

Place-order

GET /placeorder/health – health check for placeorder	
Parameters	none
Response	content type – application/json
Code – 200	description – success example – <pre>{ "message": "service is healthy." }</pre>
POST /createorder - create an order	

Parameters	
Name	Description
Body object (body)	<p>Example Value</p> <pre>{ "user_id":110, "product_list":[{"product_id":978, "quantity":1}], "card_details":{"cardNumber" : "4000056655665556", "expiryMonth":12, "expiryYear":2023, "cvv":"123"} }</pre>
Response	content type – application/json
Code – 200	<p>description – success example –</p> <pre>{ "message": "Order placed." }</pre>
Code - 400	<pre>{ "error": <specific error message>, "message": "An error occurred creating the order.1" }</pre>
POST /deleteorder - delete order by ID	
Parameters	
Name	Description
Body object (body)	<p>Example Value</p> <pre>{ "order_id":110 }</pre>

Response	
Code – 200	description – success <pre>{ "message": "Order deleted." }</pre>
Code – 404	description – order not found example – <pre>{ "timestamp": "2021-11-10T05:45:29.713+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/deleteorder" }</pre>

Wishlist

GET /wishlist – get all wishlist items	
Parameters	none
Responses	content type – application/json
Code – 200	description – success example – <pre>{ "data": { "items": [{ "book_id": 987, "item_id": 10, "user_id": 1 }, { "book_id": 988, "item_id": 11, "user_id": 2 }, { "book_id": 989,</pre>

	<pre> "item_id": 12, "user_id": 2 }] } }</pre>
Code – 404	description – no items example – <pre> { "message": "There are no items." }</pre>
GET /wishlist?user=<int:user_id> – get all items by user ID	
Parameters	
Name – user_id (int, <i>query</i>)	description – user ID of items to return example – 1
Responses	content type – application/json
Code – 200	description – success example – <pre> { "data": { "items": [{ "book_id": 987, "item_id": 10, "user_id": 1 }] } }</pre>
Code – 404	description – no items example – <pre> { "message": "There are no items." }</pre>
GET /wishlist?book=<int:book_id> – get all items by book ID	
Parameters	
Name – book_id (int, <i>query</i>)	description – book ID of items to return example – 989
Responses	content type – application/json

Code – 200	description – success example – <pre> { "data": { "items": [{ "book_id": 989, "item_id": 12, "user_id": 2 }] } } </pre>
Code – 404	description – no items example – <pre> { "message": "There are no items." } </pre>
GET /wishlist/<int:item_id> – find item by ID	
Parameters	
Name – item_id (int, <i>path</i>)	description – ID of item to return example – 11
Responses	content type – application/json
Code – 200	description – success example – <pre> { "data": { "book_id": 988, "item_id": 11, "user_id": 2 } } </pre>
Code – 404	description – item not found example – <pre> { "message": "Item not found." } </pre>
POST /wishlist – create item	
Parameters	
Name – body (object, <i>body</i>)	description – item to create example –

	<pre>{ "book_id": 990, "user_id": 1 }</pre>
Responses	content type – application/json
Code – 201	description – success example – <pre>{ "data": { "book_id": 990, "item_id": 13, "user_id": 1 } }</pre>
Code – 500	description – error example – <pre>{ "message": "An error occurred creating the item.", "error": <specific details of the error> }</pre>
DELETE /wishlist/<int:item_id> – delete item	
Parameters	
Name – item_id (int, <i>path</i>)	description – ID of item to delete example – 10
Responses	content type – application/json
Code – 200	description – success example – <pre>{ "data": { "item_id": 10 } }</pre>
Code – 404	description – item not found example – <pre>{ "message": "Item not found." }</pre>
Code – 500	description – error example – <pre>{</pre>

	<pre> "message": "An error occurred deleting the item.", "error": <specific details of the error> } </pre>
--	--

Save Wishlist

POST /save-wishlist – save wishlist	
Parameters	
Name – body (object, <i>body</i>)	<p>description – books to add to wishlist example –</p> <pre> { "user_id": 1, "wishlist_books": [{ "book_id": 991 }, { "book_id": 992 }] } </pre>
Responses	content type – application/json
Code – 201	<p>description – success example –</p> <pre> { "message": "Wishlist saved.", "data": [{ "book_id": 991, "item_id": 13, "user_id": 1 }, { "book_id": 992, "item_id": 14, "user_id": 1 }] } </pre>
Code – 500	<p>description – unable to find book example –</p> <pre> { </pre>

	<pre> "message": "Unable to save wishlist.", "error": "Unable to find book record." } </pre>
Code – 500	<p>description – unable to create item</p> <p>example –</p> <pre> { "message": "Unable to save wishlist.", "error": "Unable to create item record." } </pre>