

HitachiStore

API Documentation

Lily Sumarta Tarigan | March 11, 2024
082282775880 | lilyursula6@gmail.com

1. About App

App Name is "Hitachi Store". This App build to manage all datas of store, including: User, Customer, Product, and Transaction using API.

No	Name	Note
1	Language	Java Springboot 3.1.10
2	RDBMS	MySQL
3	Postman	Yes
4	Swagger	No

2. How To Access

This API have implement security using *Basic Auth*. But the access still using 1 credential (Admin). The Auth have set up at each API request window on Postman Collection.

No	Access	Credential
1	admin	admin , password
2	user	-

3. API Documentation

a. User (url = [hitachistore/v1/user/](#))

This API build for manage all datas of each user who have registered. User only able created by admin. Remember : When each user is successfully added, *1 record belonging to that user will also be added to the customer table*, but it will only consist of id_user, created_by, and created_at. Therefore, both users and admins need to complete the customer data for that user. This API consist of 4 Methods, which is :

No	Method	Url
1	[POST] Add New User	/addUser
	Request { "hitachiStoreUserUsername": "user6", "hitachiStoreUserEmail": "user6@gmail.com", "hitachiStoreUserPassword": "admin", "hitachiStoreUserRole": "user" }	
	Response {username} Berhasil Ditambahkan!	

2	[Get] Get All User	/getAllUser
	Response <pre>[{ "hitachiStoreUserPassword": "password", "hitachiStoreUserUsername": "admin", "hitachiStoreUserRole": "admin", "hitachiStoreUserId": 1, "hitachiStoreUserEmail": "admin@gmail.com" }, { "hitachiStoreUserPassword": "password", "hitachiStoreUserUsername": "user", "hitachiStoreUserRole": "user", "hitachiStoreUserId": 2, "hitachiStoreUserEmail": "user@gmail.com" }]</pre>	

3	[DELETE] Delete User	/deleteUser/{id}
	Response <pre>{username} Berhasil di Hapus!</pre>	

4	[UPDATE] Update User	/updateUser
	Request <pre>{ "hitachiStoreUserId": 2, "hitachiStoreUserPassword": "password", "hitachiStoreUserEmail": "user@gmail.com", "hitachiStoreUserRole": "user", "hitachiStoreUserUsername": "user" }</pre> Response <pre>{username} Berhasil di Update!</pre>	

a. **Customer** (url = hitachi/v1/customer)

This API build for manage all datas of each customer.

Remember, **You only can add new customer by API Add New User in first step**. Next you can complete other customer data by API Update Data Customer. This API consist of 2 Methods only, which is :

No	Method	Url
1	[GET] All Customer	/getAllCustomer
	Response <pre>[{ "updatedAt": "admin", "customerName": "Jessica Jane", "birthPlace": "Palembang", "userId": 2, "updatedAt": null, "createdBy": "admin", "createdAt": "2023-09-03", "birthDate": "1997-10-09" }]</pre>	

2	[Update] Update Data Customer	/updateCustomer
	Request <pre>{ "userId": "2", "customerName": "Jessica Jane", "birthDate": "1997-10-09", "birthPlace": "Palembang" }</pre> Response {username} Berhasil di Update!	

b. Product (url = /hitachistore/v1/product)

This API build for manage all datas of Product by Admin. This API consist of 4 Methods , which is :

No	Method	Url
1	[POST] Add New Product	/addProduct
	Request <pre>{ "productPrice": 2000000, "productTax": 10, "productName": "monitor" }</pre> Response {productName} Berhasil Ditambahkan!	

2	[Get] Get All Product	/getAllProduct
	Response <pre>{ "productName": "monitor", "productId": 2, "productPrice": 2000000, "productTax": 10 }</pre>]	

3	[DELETE] Delete Product	/deleteProduct/{id}
	Response Produk Berhasil di Hapus!	

4	[UPDATE] Update User	/updateProduct
<p>Request</p> <pre>{ "productPrice": 2000000, "productId": 1, "productName": "Keyboard", "productTax": 10 }</pre> <p>Response</p> <pre>{productName} Berhasil di Update!</pre>		

d. **Transaction** (url = /hitachistore/v1/transaction)

This API build for manage all Transactions data. As request, this API consist of 13 Methods . But this app still have 10 APIs, which is :

No	Method	Url
1	[GET] Get All Transaction	/getAllTransaction
<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-10", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>		

2	[Get] Get By Customer	/ getAllTransaction/{userid}
	<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-10", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>	

3	[Get] Get By Payment Status	/ getAllTransactionByStatus?payment_status=canceled
	<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-10", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>	

4	[Get] Get By Payment Method	/getAllTransactionByMethod?payment_method=debit
	<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-10", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>	

5	[Get] Get By Date Range	/getAllTransactionByDateRange?start_date=&end_date=
	<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-10", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>	

6	[Get] Get By Sort Time	/getAllTransactionSortTime?sort_order=asc
	<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-09", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>	

7	[Get] Get By Staff Created	/getAllTransactionByStaffCreated?staff_id={staff_id}
	<p>Response</p> <pre>[{ "customer": 1, "productId": 1, "totalTaxPaid": 200000, "transactionTime": "2024-03-10", "transactionId": 1, "netAmountPaid": 2000000, "paymentStatus": "canceled", "totalAmountPaid": 2200000, "paymentMethod": "debit", "staffId": 1 }]</pre>	

8	[Get] Total Trx Per Product	/totalAmountSpentPerProduct?product_id={product_id}
	Response total	

9	[Get] Total Trx Per Customer	/totalAmountSpentByCustomer?customer_id={cust_id}
	Response Total	