

Clarification Table

web service API operation	Endpoint	Requirement	Notes	Example Usage
Class: ProductController Function: addProduct	/product/add	Not explicit in requirements, but for functionality purpose	It requires a Body Parameter. An example is in Example Usage	METHOD : POST URL : http://localhost:8080/product/add BODY : { "serialNumber": "1", "name": "fan", "vendor": "LG", "category": "HOME", "price": 120.0, "availablePiecesNumber": 10 }
Class: ProductController Function: removeProduct	/product/remove/{serialNumber}	Not explicit in requirements, but for functionality purpose	It doesn't require Body Parameter, only a valid serial number in the URL	METHOD : DELETE URL : http://localhost:8080/product/remove/1 BODY :
Class: ProductController Function: listAvailableProducts	/product/list	1. A list of all the products currently available for purchase	It doesn't require anything but the URL.	METHOD : GET URL : http://localhost:8080/product/list BODY :
Class: ProductController Function: countProducts	/product/{categoryName}	1. A count of the remaining parts from each category should be available.	It doesn't require Body Parameter, only a valid category name in the URL	METHOD : GET URL : http://localhost:8080/product/HOME BODY :
Class: NotificationsController Function: getNotifications	/notifications	8. list the current content of notifications Queue.	in the bonus part it's required to remove the notifications from the queue after a configured time so in our code if the notification stayed more than 3 minutes in the queue it will be removed	METHOD : GET URL : http://localhost:8080/notifications BODY :

Class: NotificationsController Function: getMostNotified	/mostNotified	10. get The most notified email-address/phone-number (bonus)	METHOD : GET URL : http://localhost:8080/mostNotified BODY :
Class: NotificationsController Function: getMostUsedTemp	/mostUsedTemplate	10. get The most sent notification template. (bonus)	METHOD : GET URL : http://localhost:8080/mostUsedTemplate BODY :
Class: OrderController Function: placeSimpleOrder	/order/placeSimpleOrder	3. A customer can place a simple order, where such an order would include a single product or several products.	Use: - a registered email, - the previously added products, - valid amounts of each product, to complete the order placement and get status code 201 "Created", otherwise it ends with status code 400 "Bad request". The default shipping fee of the simple order is 50. Order ID begins from 1 and is autoincremented. METHOD : POST URL : http://localhost:8080/order/placeSimpleOrder BODY : { "accEmail": "example@example.com", "prodSerialNum": [{"1": 2}] }
Class: OrderController Function: placeCompoundOrder	/order/placeCompoundOrder{email}	3. A customer can make a compound order, 5. To place a compound order, the customer needs to pick his products, and his friends' orders as well.	Use: - a registered email, - the previously added products, - valid amounts of each product, to complete the order placement and get status code 201 "Created", otherwise it ends with status code 400. The default shipping fee of each simple order inside the compound order is 20. METHOD : POST URL : http://localhost:8080/order/placeCompoundOrder/e: BODY : [{ "accEmail": "example1@example.com", "prodSerialNum": [{"1": 2}, {"2": 2}] }, { "accEmail": "example2@example.com", "prodSerialNum": [{"1": 1}, {"2": 1}] }]

			Order ID begins from 1 and is autoincremented.	<pre>] }] </pre>
Class: OrderController Function: listOrders	<code>/order/listOrders</code>	4. You should be able to list all the details of both simple and compound orders.		METHOD : GET URL : http://localhost:8080/order/listOrders BODY :
Class: OrderController Function: cancelOrder	<code>/order/cancelOrder/{id}</code>	9 - bonus: Customers can cancel an order placement, or cancel only its shipping within a pre-configured automated duration. Such cancellations should be updated appropriately within the system.	Use: - any of the generated order IDs, whether it's simple or compound, to complete the order cancellation and get status code 202 "Accepted", otherwise it ends with status code 403 "Forbidden".	METHOD : Delete URL : http://localhost:8080/order/cancelOrder/1 BODY :
Class: OrderController Function: cancelOrderShipping	<code>/order/cancelOrderShipping/{id}</code>	9 - bonus: Customers can cancel an order placement, or cancel only its shipping within a pre-configured automated duration. Such cancellations should be updated appropriately within the system.	Use: - any of the generated order IDs, whether it's simple or compound, to complete the order cancellation and get status code 202 "Accepted", otherwise it ends with status code 403 "Forbidden".	METHOD : Delete URL : http://localhost:8080/order/cancelOrderShipping/1 BODY :

<p>Class: AccountController</p> <p>Function: createAccount</p>	<p><code>/createAccount</code></p>	<p>2. A customer should be able to create an account and put a specific balance using that account. Such balance would be utilized during future purchasing operations.</p>	<p>It requires a Body Parameter and there is validation on Body content.</p> <p>VALIDATION:</p> <p>Name: can not be empty.</p> <p>Email: must follow email format.</p> <p>Password: must be at least 8 characters, and contain at least one uppercase letter and one digit.</p> <p>PhoneNumber: must be exactly 11 digits and contain only numeric digits.</p> <p>Language: must be either German or English.</p> <p>Balance: must be a numeric value with up to 1000000000 integer digits and 10 fraction digits.</p> <p>NotificationChannel: notification channel must be SMS , Email or both</p> <p>An example is in Example Usage</p>	<p>METHOD :</p> <p>POST</p> <p>URL :</p> <p>http://localhost:8080/createAccount</p> <p>BODY :</p> <pre>{ "name": "aya", "email": "aya@gmail.com", "password": "Aya12345", "phoneNumber": "01034567888", "language": "English", "notificationChannel": "Both", "balance": 1000.5 }</pre>
--	------------------------------------	---	---	--

The Sequence You Need To Follow To Test All The Features Successfully :

- you need to create an account
 - refer to row number 13 in the table
- then you need to add products to the store to test the features of listing products / place order / ...
 - refer to row number 1 in the table
- you can remove that product if you want
 - refer to row number 2 in the table
- you can list the products in the store
 - refer to row number 3 in the table
- Or you can see count of the remaining parts of each category :
 - refer to row number 4 in the table
- Or you can place an order with the type you want (simple / compound)
 - refer to row number 8 / 9 in the table

- If you want to cancel this order or cancel it's shipping you have to do that before the `ALLOWED_DURATION` which is 120 seconds **(Bonus)**
 - refer to row number 11 / 12 in the table

-
- then you can list the orders you made
 - refer to row number 10 in the table

-
- Or list the notifications in the queue
 - refer to row number 5 in the table
 - The notifications are removed from the queue after the `ALLOWED_DURATION` for them to stay which is 180 seconds **(Bonus)**

-
- you can get the most notified phone number or email **(Bonus)**
 - refer to row number 6 in the table

-
- you can get the most used template **(Bonus)**
 - refer to row number 7 in the table