No	Test Cases	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	Link to Bug Report

1	Adding to an existing product ID to the kit.	1. The user selects POST request with URL + api/v1/kits/:id/products
		
		2. The user sends a request by using an existing product ID in the path params and request body.
		
		3. The user adds existing products to the kit in the request body.
			Path params: id=6
			
			{
			"productsList": [
			{
			"id": 3,
			"quantity": 3
			}
			]
			}
				- Added existing products to the kit
				
				- Status: 200 OK
					- Added existing products to the kit
					
					- Status: 200 OK
						Passed	

2	Adding a product to an empty kit	1. The user selects POST request with URL + /api/v1/kits
		
		2. The user sends a request by using a new kit name and cardid params in the request body.
		{
		"name": "ryley1",
		"cardId": 9
		}
		3. The user gets status 201 created on the response and a new kit ID (8) in the response body
		
		4. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		5.The user adds existing product to kit in the request body
			Path params: id=8
			
			{
			"productsList": [
			{
			"id": 4,
			"quantity": 1
			}
			]
			}
				- created new kit
				- added new product to new kit
				- Status: 200 OK
					- created new kit
					- added new product to new kit
					- Status: 200 OK
						Passed	

3	Adding multiple existing products to an existing kit.	1. The user selects POST request with URL + api/v1/kits/:id/products
		
		2. The user sends a request by using an existing product ID in the path params and request body.
		
		3. The user adds existing products to the kit in the request body.
			Path param id=8
			
			{
			"productsList": [
			{
			"id": 15,
			"quantity": 4
			},
			{
			"id": 18,
			"quantity": 2
			}
			]
			}
				- Added multiple existing products to the kit
				
				- Status: 200 OK
					- Added multiple existing products to the kit
					
					- Status: 200 OK
						Passed	

4	Adding a product to a kit when the kit already has **30** products	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param
		(the kit chosen must have more than 30 products)
		
		3.The user attempts to add existing product to kit in the request body
			Path params id=1
			
			{
			"productsList": [
			{
			"id": 49,
			"quantity": 5
			}
			]
			}
				-products not added
				
				{
				"code": 400,
				"message": "No more than 30 items per set"
				}
					-products not added
					
					{
					"code": 400,
					"message": "No more than 30 items per set"
					}
						Passed	

5	Adding a **multiple products** to a kit when the kit already has **30** products	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param
		(the kit chosen must have less than 30 products)
		
		3.The user adds existing muliple product to kit in the request body
			Path params id=1
			
			{
			"productsList": [
			{
			"id": 15,
			"quantity": 4
			},
			{
			"id": 18,
			"quantity": 2
			}
			]
			}
				-products not added
				
				{
				"code": 400,
				"message": "No more than 30 items per set"
				}
					-products not added
					
					{
					"code": 400,
					"message": "No more than 30 items per set"
					}
						Passed	

6	Adding a **Single Product with Multiple Quantities** to a Kit with Fewer Than 30 Items to Reach Exactly **29** Items	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param (the kit chosen must have less than 30 products)
		
		3.The user attempts to add an existing product with a specified quantity to a kit, causing the total count of items to exactly reach **29**.
			Path param id=6
			
			{
			"productsList": [
			{
			"id": 17,
			"quantity": 19
			}
			]
			}
				STATUS 200
				ADDITION OF EXISTING PRODUCTS TO KIT
				
				PRODUCT COUNT SHOULD NOW EQUAL **29**
					STATUS 200
					
					PRODUCT COUNT EQUAL TO **29**
						Passed	

7	Adding a **Single Product with Multiple Quantities** to a Kit with Fewer Than 30 Items to Reach Exactly **30** Items	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param (the kit chosen must have less than 30 products)
		
		3.The user attempts to add an existing product with a specified quantity to a kit, causing the total count of items to exactly reach **30**.
			Path param id=6
			
			{
			"productsList": [
			{
			"id": 17,
			"quantity": 20
			}
			]
			}
				- Added existing products to the kit
				
				- Status: 200 OK
					- Added existing products to the kit
					
					- Status: 200 OK
						Passed	

8	Adding a **Single Product with Multiple Quantities** to a Kit with Fewer Than 30 Items to Reach Exactly **31** Items	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param (the kit chosen must have less than 30 products)
		
		3.The user attempts to add an existing product with a specified quantity to a kit, causing the total count of items to exactly reach **31**.
			Path param id=6
			
			{
			"productsList": [
			{
			"id": 17,
			"quantity": 21
			}
			]
			}
				-products not added
				
				{
				"code": 400,
				"message": "No more than 30 items per set"
				}
					- Added existing products to the kit causing it to exceed the 30 count limit by 1
					
					- Status: 200 OK
						Failed	https://rj178997.atlassian.net/browse/S4-1?atlOrigin=eyJpIjoiZTNlNjE2N2YzOTJlNDQyYmIwMGJhZDlkOTg3OTdhOGUiLCJwIjoiaiJ9

9	Adding **Single Quantities of Multiple Products** to a kit with fewer than 30 items, thereby increasing the count to reach exactly **29**.	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param (the kit chosen must have less than 30 products)
		
		3.The user attempts to add multiple existing products, causing the total count of items to exactly reach 29.
			Path param id=6
			
			{
			"productsList": [
			{"id": 30, "quantity": 1},
			{"id": 31, "quantity": 1},
			{"id": 32, "quantity": 1},
			{"id": 33, "quantity": 1},
			{"id": 34, "quantity": 1},
			{"id": 35, "quantity": 1},
			{"id": 36, "quantity": 1},
			{"id": 37, "quantity": 1},
			{"id": 38, "quantity": 1},
			{"id": 39, "quantity": 1},
			{"id": 40, "quantity": 1},
			{"id": 41, "quantity": 1},
			{"id": 42, "quantity": 1},
			{"id": 43, "quantity": 1},
			{"id": 44, "quantity": 1},
			{"id": 45, "quantity": 1},
			{"id": 46, "quantity": 1},
			{"id": 47, "quantity": 1},
			{"id": 48, "quantity": 1}
			]
			}
				STATUS 200
				ADDITION OF EXISTING PRODUCTS TO KIT
				
				PRODUCT COUNT SHOULD NOW EQUAL **29**
					STATUS 200
					
					PRODUCT COUNT EQUAL TO **29**
						Passed	

10	Adding **Single Quantities of Multiple Products** to a kit with fewer than 30 items, thereby increasing the count to reach exactly **30.**	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param (the kit chosen must have less than 30 products)
		
		3.The user attempts to add multiple existing products, causing the total count of items to exactly reach 30.
			Path param id=6
			
			{
			"productsList": [
			{"id": 30, "quantity": 1},
			{"id": 31, "quantity": 1},
			{"id": 32, "quantity": 1},
			{"id": 33, "quantity": 1},
			{"id": 34, "quantity": 1},
			{"id": 35, "quantity": 1},
			{"id": 36, "quantity": 1},
			{"id": 37, "quantity": 1},
			{"id": 38, "quantity": 1},
			{"id": 39, "quantity": 1},
			{"id": 40, "quantity": 1},
			{"id": 41, "quantity": 1},
			{"id": 42, "quantity": 1},
			{"id": 43, "quantity": 1},
			{"id": 44, "quantity": 1},
			{"id": 45, "quantity": 1},
			{"id": 46, "quantity": 1},
			{"id": 47, "quantity": 1},
			{"id": 48, "quantity": 1},
			{"id": 49, "quantity": 1}
			]
			}
				- Added existing products to the kit
				
				- Status: 200 OK
					- Added existing products to the kit
					
					- Status: 200 OK
						Passed	

11	Adding **single quantities of multiple products** to a kit with fewer than 30 items, thereby increasing the count to **31.**	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a prexisting kit id path param (the kit chosen must have less than 30 products)
		
		3.The user attempts to add existing individual products to kit causing it to exceed the count of 30 in the request body
			Path params id=3
			
			{
			"productsList": [
			{"id": 14, "quantity": 1},
			{"id": 20, "quantity": 1},
			{"id": 19, "quantity": 1},
			{"id": 21, "quantity": 1},
			{"id": 22, "quantity": 1},
			{"id": 23, "quantity": 1},
			{"id": 24, "quantity": 1},
			{"id": 25, "quantity": 1},
			{"id": 26, "quantity": 1},
			{"id": 27, "quantity": 1},
			{"id": 28, "quantity": 1},
			{"id": 29, "quantity": 1}
			]
			}
				-products not added
				
				{
				"code": 400,
				"message": "No more than 30 items per set"
				}
					-products not added
					
					{
					"code": 400,
					"message": "No more than 30 items per set"
					}
						Passed	

12	adding non-existent numberic product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a preexisting kit id path param
		
		3.The user attempts to add a non existent product id to kit in the request body
			Path params id=2
			
			{
			"productsList": [
			{
			"id": 407,
			"quantity": 1
			}
			]
			}
				- products not added
				
				-Status 400 Bad Request
					- product count increased
					
					- Status 200 OK
						Failed	https://rj178997.atlassian.net/browse/S4-2?atlOrigin=eyJpIjoiZWY3Yjk1OTM4NTk3NDdkODgzMDJlOTIwNmY2YzNjNWQiLCJwIjoiaiJ9

13	adding exisiting products to a non-existing kit ID	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a non existent kit id path param
		
		3.The user attempts to add an existing product id to kit in the request body
			Path params id=200
			
			{
			"productsList": [
			{
			"id": 40,
			"quantity": 1
			}
			]
			}
				-kit not found
				
				-Status 404 not found
					-kit not found
					
					-Status 404 not found
						Passed	

14	inputing a fractional quantity value of a product addition to a kit	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a existent kit id path param
		
		3.The user attempts to add an existing product id to kit in the request body
		with a fractional quantity value
			Path params id=3
			
			{
			"productsList": [
			{
			"id": 3,
			"quantity": 3.5
			}
			]
			}
				400 Bad Request	500 Internal Server Error returned
					
					{
					"code": 500,
					"message": "invalid input syntax for integer: \"28.5\""
					}
						Failed	https://rj178997.atlassian.net/browse/S4-3?atlOrigin=eyJpIjoiZGUwMWZlZGZkODk5NGQwNGFiMzk4ZTE5ODNkNTAwN2UiLCJwIjoiaiJ9

15	inputing fractional numbers as Product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a existent kit id path param
		
		3.The user attempts to add fractional number as a product id to kit in the request body
			Path params id=3
			
			{
			"productsList": [
			{
			"id": 3.5,
			"quantity": 3
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					
					<!DOCTYPE html>
					<html lang="en">
					
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					
					<body>
					<pre>Internal Server Error</pre>
					</body>
					
					</html>
						Failed	https://rj178997.atlassian.net/browse/S4-4?atlOrigin=eyJpIjoiYWVmZjU3MWJmOTU4NDVjOGIzMTJlZWJkYTA5MTJmMGQiLCJwIjoiaiJ9

16	adding latin characters as product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a existent kit id path param
		
		3.The user attempts to add latin letters as a product id to kit in the request body
			Path param id=4
			
			{
			"productsList": [
			{
			"id": "milk",
			"quantity": 3
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					
					<!DOCTYPE html>
					<html lang="en">
					
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					
					<body>
					<pre>Internal Server Error</pre>
					</body>
					
					</html>
						Failed	https://rj178997.atlassian.net/browse/S4-5?atlOrigin=eyJpIjoiNDI2ZjE2Yjg1MWYwNDI5ZWEzMjM2YzJkNWFmY2UxNjUiLCJwIjoiaiJ9

17	adding latin characters as quantity values with existing product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a existent kit id path param
		
		3.The user attempts to add latin letters as a quantity value to kit in the request body
			Path param id=4
			
			{
			"productsList": [
			{
			"id": 5,
			"quantity": "three"
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					
					{
					"code": 500,
					"message": "invalid input syntax for integer: \"33three\""
					}
						Failed	https://rj178997.atlassian.net/browse/S4-6?atlOrigin=eyJpIjoiNGU2YTZhNjY4NTYzNGVhZGFjOGQwMDgxZmJhZTlmNzliLCJwIjoiaiJ9

18	adding non latin characters as product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a existent kit id path param
		
		3.The user attempts to add non latin letters as a product id to kit in the request body
			Path param id=4
			
			{
			"productsList": [
			{
			"id": "молоко",
			"quantity": 3
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					
					<!DOCTYPE html>
					<html lang="en">
					
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					
					<body>
					<pre>Internal Server Error</pre>
					</body>
					
					</html>
						Failed	https://rj178997.atlassian.net/browse/S4-7?atlOrigin=eyJpIjoiOTE0Zjg1M2JlYWEzNGI4NGE0NWY4ZDMwN2I5YzU3YmMiLCJwIjoiaiJ9

19	adding non latin characters as quantity values with existing product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user attempts to add non latin letters as a quantity value to kit in the request body
			Path param id=4
			{
			"productsList": [
			{
			"id": 5,
			"quantity": "четыре"
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					{
					"code": 500,
					"message": "invalid input syntax for integer: \"33четыре\""
					}
						Failed	https://rj178997.atlassian.net/browse/S4-8

20	adding symbols as quantity values with existing product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user attempts to add symbols as a quantity value to kit in the request body
			Path param id=4
			{
			"productsList": [
			{
			"id": 5,
			"quantity": "$#%"
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					{
					"code": 500,
					"message": "invalid input syntax for integer: \"33$#%\""
					}
						Failed	https://rj178997.atlassian.net/browse/S4-10

21	adding symbols as product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user attempts to add symbols as a product id to kit in the request body
			Path param id=4
			{
			"productsList": [
			{
			"id": "$@^",
			"quantity": 3
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	https://rj178997.atlassian.net/browse/S4-9

22	adding negative numbers as product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user attempts to add negative numbers as a product id to kit in the request body
			Path param id=4
			{
			"productsList": [
			{
			"id": -1,
			"quantity": 5
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	https://rj178997.atlassian.net/browse/S4-11

23	adding negative numbers as quantity values with existing product IDs to existing kits	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user attempts to add negative numbers as a quantity value to kit in the request body
			Path param id=4
			{
			"productsList": [
			{
			"id": 6,
			"quantity": -4
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	https://rj178997.atlassian.net/browse/S4-12

24	Absence of Product ID in request body	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user omits product id in the request body
		4. The user inputs an existent quantity value in the request body
			Path param id=4
			{
			"productsList": [
			{
			"id": "",
			"quantity": 4
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	bug report same issue as above

25	Absence of Product ID Param in request body	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3.The user omits entire product id param in the request body
		4. The user inputs an existent quantity value in the request body
			Path param id=4
			{
			"productsList": [
			{
			"quantity": 4
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	bug report same issue as above

26	Absence of request body	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3. The user omits entire request body
			Path param id=4
			{
			"productsList": [
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	bug report same issue as above

27	Absence of ProductList in request body	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		2. The user sends a request by using a existent kit id path param
		3. The user omits productList param
			Path param id=4
			{
			"": [
			{
			"id": 6,
			"quantity": 4
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					<!DOCTYPE html>
					<html lang="en">
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					<body>
					<pre>Internal Server Error</pre>
					</body>
					</html>
						Failed	bug report same issue as above

28	Absence of quantity in request body	1. The user selects POST request with URL+ /api/v1/kits/:id/products.
		
		2. The user sends a request by using a existent kit id path param
		
		3.The user inputs and existent product id in the request body
		
		4. The user omits a quantity value in the request body
			Path param id=4
			
			{
			"productsList": [
			{
			"id": 5,
			"quantity": " "
			}
			]
			}
				400 Bad Request	500 Internal Server Error
					
					<!DOCTYPE html>
					<html lang="en">
					
					<head>
					<meta charset="utf-8">
					<title>Error</title>
					</head>
					
					<body>
					<pre>Internal Server Error</pre>
					</body>
					
					</html>
						Failed	bug report same issue as above

1	Inputing a valid request meeting the requirements of fast delivery mode	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = 20	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>20</deliveryTime> </InputModel>	200 OK Response body: <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK Response body: <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
2	Ordering Fast Delivery with a **minimum product weight** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **0**, and deliveryTime = 9	<InputModel> <productsCount>5</productsCount> <productsWeight>0</productsWeight> <deliveryTime>9</deliveryTime> </InputModel>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
3	Ordering Fast Delivery with a **minimum product weight and minimum product count for free client delivery** **PRODUCTWEIGHT BV** **PRODUCTCOUNT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **1**, productsWeight = **0**, and deliveryTime = 9	<InputModel> <productsCount>1</productsCount> <productsWeight>0</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
9	Ordering Fast Delivery with Slightly **Exceeding Product Weight Limit** for **Host Delivery Cost of $3** and Slightly Exceeding Weight Limit for **Client to Receive Free Delivery** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **6.01**, and deliveryTime = 10	<InputModel> <productsCount>5</productsCount> <productsWeight>6.01</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="9"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="6"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Failed	https://rj178997.atlassian.net/browse/S4-13?atlOrigin=eyJpIjoiZDBhOTBmNWQzZjdmNGRmN2JkZmZiYTY3Y2QwMWM5YTUiLCJwIjoiaiJ9
10	Ordering Fast Delivery with **product weight at exact limit for hostdelivery price to equal $3** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **2.5**, and deliveryTime = 10	<InputModel> <productsCount>5</productsCount> <productsWeight>2.5</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
11	Ordering Fast Delivery with **product weight at just below the limit for hostdelivery price to equal $3** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **2.49**, and deliveryTime = 10	<InputModel> <productsCount>5</productsCount> <productsWeight>2.49</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
12	Ordering Fast Delivery with **product weight at just above the limit for hostdelivery price to equal $3** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **2.51**, and deliveryTime = 10	<InputModel> <productsCount>5</productsCount> <productsWeight>2.51</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	****Host Delivery Cost: Unknown (due to gap in defined ranges) Note: The current requirements have a gap between the $3 price range (0-2.5 kg) and the $6 price range (2.6-6 kg). Please revise the requirement to specify the weight ranges with greater precision (e.g., 2.50 kg - 2.59 kg and 2.60 kg - 6 kg) to remove any ambiguity.	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	inconclusive	
13	Ordering Fast Delivery with **product weight at slightly below for hostdelivery price to equal $6** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **2.59**, and deliveryTime = 10	<InputModel> <productsCount>5</productsCount> <productsWeight>2.59</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	****Host Delivery Cost: Unknown (due to gap in defined ranges) Note: The current requirements have a gap between the $3 price range (0-2.5 kg) and the $6 price range (2.6-6 kg). Please revise the requirement to specify the weight ranges with greater precision (e.g., 2.50 kg - 2.59 kg and 2.60 kg - 6 kg) to remove any ambiguity.	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="3" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	inconclusive	
14	Ordering Fast Delivery with **Exact Minimum Product Weight for Host Delivery Price of $6** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **2.6**, and deliveryTime = 10	<InputModel> <productsCount>5</productsCount> <productsWeight>2.6</productsWeight> <deliveryTime>10</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
15	Verifying **Host Delivery Price of $6** for Fast Delivery with **Slightly Above Minimum Product Weight** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **2.61**, and deliveryTime = 11	<InputModel> <productsCount>5</productsCount> <productsWeight>2.61</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	

16	Ordering Fast Delivery with **Below Maximum Product Count and Slighty Below Maximum Product Weight for client price to be $0** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **5**, productsWeight = **5.99**, and deliveryTime = 11	<InputModel> <productsCount>5</productsCount> <productsWeight>5.99</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
17	ordering fast delivery with above **maximum product count and below maximum product weight** **PRODUCTCOUNT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **15**, productsWeight = **2**, and deliveryTime = 11	<InputModel> <productsCount>15</productsCount> <productsWeight>2</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="9"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="6"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Failed	https://rj178997.atlassian.net/browse/S4-14
18	Ordering Fast Delivery with Exact Maximum Product Count and Exact Maximum Product Weight **PRODUCTCOUNT BV** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **14**, productsWeight = **6**, and deliveryTime = 11	<InputModel> <productsCount>14</productsCount> <productsWeight>6</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
19	Ordering Fast Delivery with **Above Maximum Product Count and Below Maximum Product Weight limits for host price to be $3** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **8**, productsWeight = **5**, and deliveryTime = 11	<InputModel> <productsCount>8</productsCount> <productsWeight>5</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
20	Ordering Fast Delivery with **EXACTLY Maximum Product Count and Below Maximum Product Weight limits for host price to be $3** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **7**, productsWeight = **5**, and deliveryTime = 11	<InputModel> <productsCount>7</productsCount> <productsWeight>5</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	
21	Ordering Fast Delivery with **BELOW MINIMUM Product Count and Below Maximum Product Weight limits for host price to be $3** **PRODUCTWEIGHT BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = **0**, productsWeight = **5**, and deliveryTime = 11	<InputModel> <productsCount>0</productsCount> <productsWeight>5</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	i would expect this to be 400 bad request because how can a customer order 0 products	<InputModel> <productsCount>0</productsCount> <productsWeight>5</productsWeight> <deliveryTime>11</deliveryTime> </InputModel>	Failed	
22	Ordering fast delivery **at the minimum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **7**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>7</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK "<response name=""Fast Delivery"" isItPossibleToDeliver="" true"" hostDeliveryCost=""6"" clientDeliveryCost=""0""> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>"	Passed	
23	ordering fast delivery at **one minute before minimum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **6:59**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>6:59</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="false" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery"/>	Failed	https://rj178997.atlassian.net/browse/S4-15
24	Ordering fast delivery **at the maximum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **21**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>21</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK "<response name=""Fast Delivery"" isItPossibleToDeliver="" true"" hostDeliveryCost=""6"" clientDeliveryCost=""0""> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>"	Passed	
25	ordering fast delivery at **one minute before maximum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **20:59**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>20:59</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK "<response name=""Fast Delivery"" isItPossibleToDeliver="" true"" hostDeliveryCost=""6"" clientDeliveryCost=""0""> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>"	Passed	
26	ordering fast delivery at **one minute after maximum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **21:01**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>20:01</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="false" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK "<response name=""Fast Delivery"" isItPossibleToDeliver="" true"" hostDeliveryCost=""6"" clientDeliveryCost=""0""> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>"	Failed	https://rj178997.atlassian.net/browse/S4-16
27	ordering fast delivery **1 hour past maximum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **22**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>22</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="false" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery"/>	Failed	https://rj178997.atlassian.net/browse/S4-17

28	ordering fast delivery **1 hour before minimum operating hours** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **6**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>6</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="false" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery"/>	Failed	same bug report issues as above
29	Fast Delivery Order Placement **During Operational Time Periods Utilizing Time Format 12:00** **OPERATING HOURS BV**	1. Send POST request to URL + /fast-delivery/v3.1.1/calculate-delivery.xml 2. input a productsCount = 2, productsWeight = 5.115, and deliveryTime = **12:00**	<InputModel> <productsCount>2</productsCount> <productsWeight>5.115</productsWeight> <deliveryTime>12:00</deliveryTime> </InputModel>	<response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	200 OK <response name="Fast Delivery" isItPossibleToDeliver="true" hostDeliveryCost="6" clientDeliveryCost="0"> <toBeDeliveredTime> <min>25</min> <max>30</max> </toBeDeliveredTime> </response>	Passed	

https://wary-ethernet-059.notion.site/URBAN-GROCERS-API-TESTING-REPORT-fd0b0c91093a4db6b0e7ddd618da14a5

^^^^^^^^FULL REPORT ^^^^^^^^^