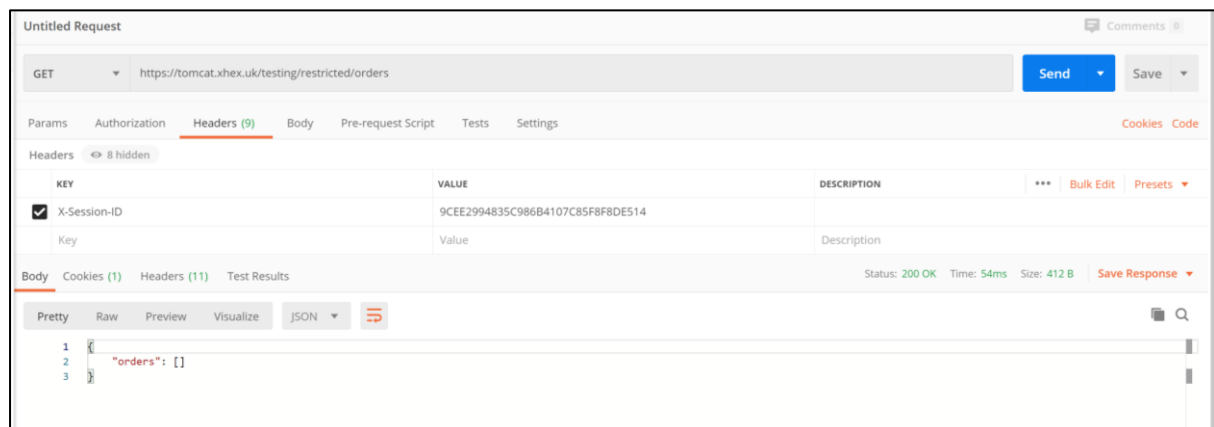


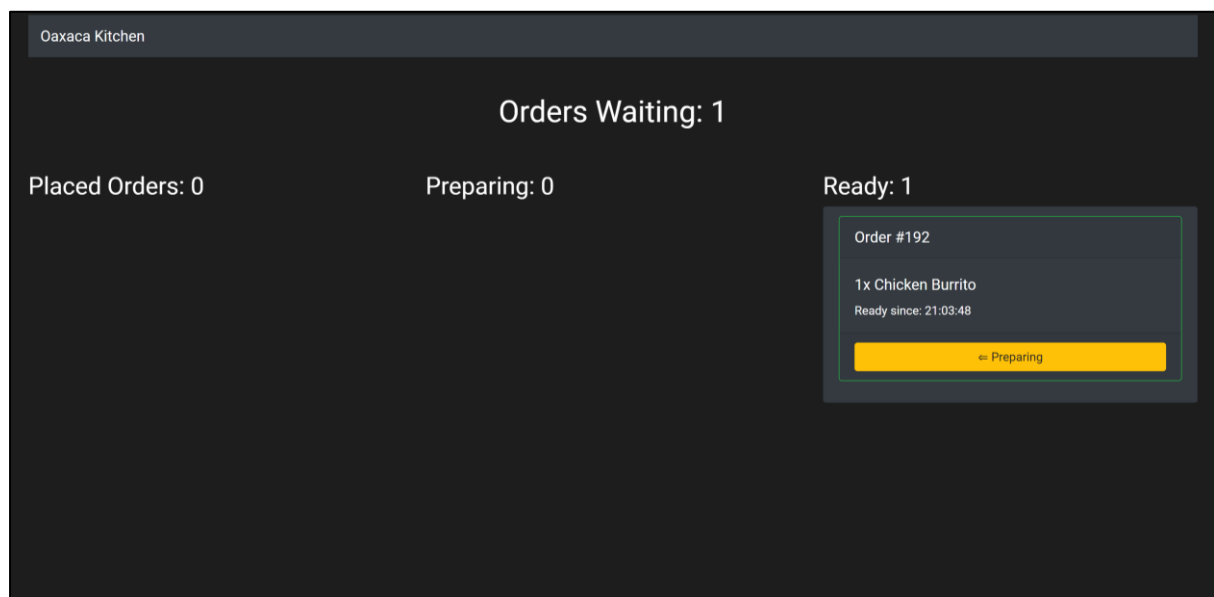
## TEST 1 (Using POSTMAN)

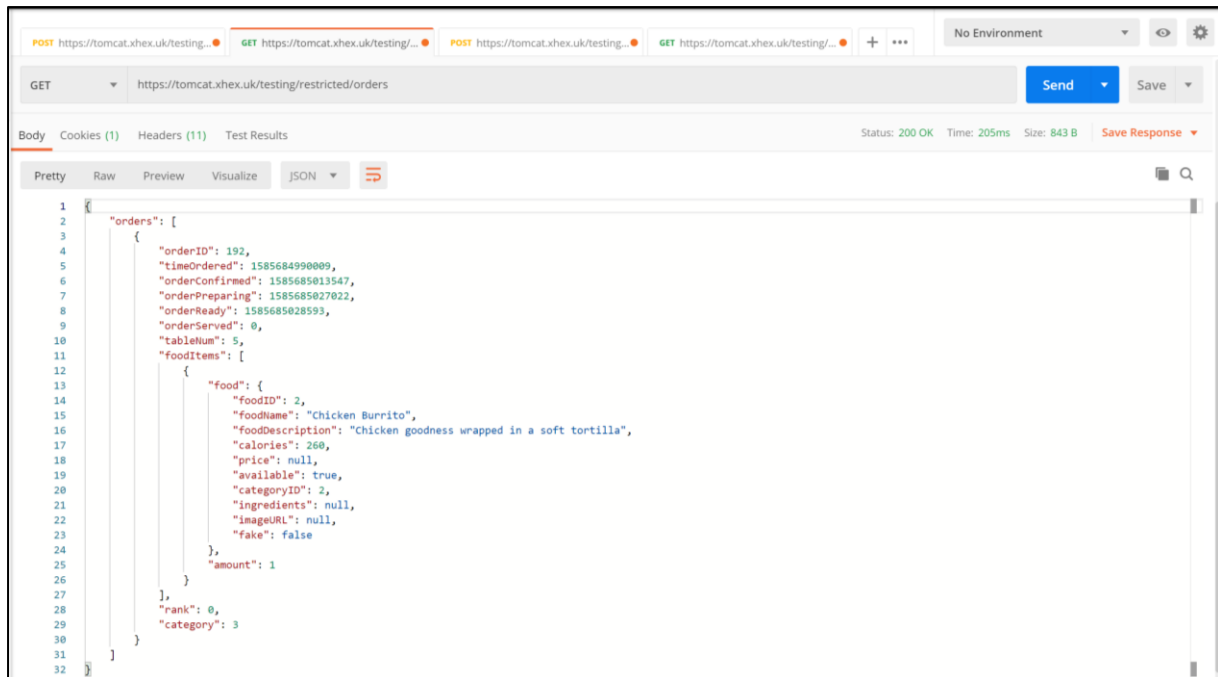
### Test - Getting orders from kitchen queue using GET method from 'OrdersToFrontend' endpoint

The way our team will carry out testing is by testing the endpoints which is done by using Postman to test the GET and POST methods which are available. To use Postman, I first deployed the WAR file onto Tomcat which allows me to create an application for me to test with. From Postman, I can send requests using either GET, POST, DELETE and more plus by inserting the correct URL. For some of the tests I will have to place parameters in the Params tab, also a session ID to authorise the connection to Tomcat.



As you can see above, we are using the 'restricted/orders' URL add-on to test the endpoint for getting the orders from the kitchen queue. When I sent the request, it returned with an empty array called orders. This is because the kitchen is empty as of now. Now I am going to add an item to the order and checkout to the kitchen, then run the test again.





After an item has been added to the kitchen queue and running the request on Postman, I was shown a different response. The array contained information on the single order which was processed. It contains all the information such as the orderID, table number, foodname and more. This helps the user and frontend know everything on the orders which are in the kitchen. To prove this information, we can look at the database where the orders are stored.

	orde...	1	table_num	time_ordered	order_confirmed	order_preparing	order_ready	order_served	order_cancelled
76		160		1584047479802	1584047560716	0	0	1	
77		161		1584047612025	1584047615347	0	0	1	
78		162		1584093400928	0	0	0	1	
79		163		1584093675098	0	0	0	1	
80		164		1584094566980	1584094571221	0	0	1	
81		165		1584094787036	0	0	0	0	
82		166		1584094894823	1584094944841	1585078580336	1585079232034	1585084829010	
83		167		1584094910957	1584094947531	1585078570164	1585080928772	1585084265724	
84		168		1584725043685	0	0	0	0	
85		169		1584898378777	0	0	0	0	
86		170		1584898405160	0	0	0	0	
87		171		1584908620182	0	0	0	0	
88		172		1584908825339	0	0	0	0	
89		173		1584911043563	0	0	0	0	
90		174		1584911616731	0	0	0	0	
91		175		1584966293028	0	0	0	0	
92		176		1584966305491	0	0	0	0	
93		177		1585067421927	0	0	0	0	
94		178		1585069020626	0	0	0	0	
95		179		1585069050914	0	0	0	0	
96		180		1585069083624	1585069100332	1585671907658	1585671908307	1585671943524	
97		181		1585069115122	1585069123101	1585078569012	1585078576590	1585081058830	
98		182		1585069736434	1585069748829	1585076383076	1585076383803	1585084278039	
99		183		1585077550177	0	0	0	0	
00		184		1585077591480	0	0	0	0	
01		185		1585077647084	0	0	0	0	
02		186		1585078459246	0	0	0	0	
03		187		1585079092213	1585079133702	1585081425969	1585081429122	1585081431993	
04		188		1585084343787	1585084573080	1585084813280	1585084818837	1585671955345	
05		189		1585684799498	0	0	0	0	
06		190		1585684812103	0	0	0	0	
07		191		1585684867270	0	0	0	0	
08		192		1585684990009	1585685013547	1585685027022	1585685028593	0	

We can see that the order number 192 is the same as that from what Postman showed from the test. Also, since we have not confirmed the order yet, that field is left blank until the order has gone from the kitchen and to the customer.