

CartApi.java

```
1 package ctrl.api;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class CartApi
7  */
8 @WebServlet("/Api/Cart")
9 public class CartApi extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14      */
15     public CartApi() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request,
22      HttpServletResponse response)
23      */
24     protected void doGet(HttpServletRequest request,
25 HttpServletResponse response) throws ServletException, IOException
26     {
27         response.setContentType("application/json");
28         response.setCharacterEncoding("utf-8");
29         response.addHeader("Access-Control-Allow-Origin", "*");
30
31         PrintWriter out = response.getWriter();
32         JsonObject json = new JsonObject();
33         JsonElement resultJson;
34         Model model = Model.getInstance();
35         HttpSession session = request.getSession();
36
37         Order userOrder = (Order) session.getAttribute("order");
38         List<Item> products = userOrder.getItems();
39
40         String addToCart = request.getParameter("addToCart");
41         String removeItem = request.getParameter("removeItem");
42         String updateQty = request.getParameter("updateQty");
43         String productQty = request.getParameter("qty");
44     }
45 }
```

CartApi.java

```
62         json.addProperty("status", 1);
63
64         if (addToCart != null) {
65             addItemToCart(model, products, json, addToCart,
productQty);
66         }
67
68         if (updateQty != null) {
69             updateQty(products, json, updateQty, productQty);
70         }
71
72         if (removeItem != null) {
73             removeItem(products, json, removeItem);
74         }
75
76         resultJson = new Gson().toJsonTree(userOrder);
77         if (!json.has("data")) {
78             json.add("data", resultJson);
79         }
80
81         if (!json.has("action")) {
82             json.addProperty("action", "GET");
83         }
84
85         out.print(json.toString());
86     }
87
88     /**
89      * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
90      */
91     protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
92     {
93         // TODO Auto-generated method stub
94         doGet(request, response);
95     }
96
97     private void addItemToCart(Model model, List<Item> products,
JsonObject json, String addToCart, String productQty) {
98         try {
99             Item product = model.getFoodBy("num", addToCart);
100             if (products.size() > 0) {
101                 if (!products.contains(product)) {
```

CartApi.java

```
101         if (productQty != null && !
productQty.isEmpty()) {
102             product.setQuantity(productQty);
103         }
104
105         products.add(product);
106
107         json.addProperty("status", 1);
108         json.addProperty("action", "ADD");
109         json.addProperty("actionReq", addToCart);
110     } else {
111         // int i = products.indexOf(product);
112         // Item p = products.get(i);
113         // if (productQty != null && !
productQty.isEmpty()) {
114         //     p.increaseQty(productQty);
115         // }
116     }
117     } else {
118         if (productQty != null && !productQty.isEmpty()) {
119             product.setQuantity(productQty);
120         }
121
122         products.add(product);
123         json.addProperty("status", 1);
124
125         json.addProperty("action", "ADD");
126         json.addProperty("actionReq", addToCart);
127     }
128 } catch (Exception e) {
129     json.addProperty("status", 0);
130     json.addProperty("data", e.getMessage());
131 }
132 }
133
134
135 private void updateQty(List<Item> products, JsonObject json,
String updateQty, String productQty) {
136     try {
137         Item product;
138         for (Item it: products) {
139             if (it.getNumber().equals(updateQty)) {
140                 if (productQty != null) {
141                     if (productQty.equals("0")) {
```

CartApi.java

```
142         products.remove(it);
143     } else {
144         product = it;
145         product.setQuantity(productQty);
146     }
147     json.addProperty("status", 1);
148 } else {
149     json.addProperty("status", 0);
150     json.addProperty("data", "No Quantity
Provided.");
151 }
152 json.addProperty("action", "UPDATE");
153 json.addProperty("actionReq", updateQty);
154 }
155 }
156 } catch (Exception e) {
157     json.addProperty("status", 0);
158     json.addProperty("data", e.getMessage());
159 }
160 }
161
162 private void removeItem(List<Item> products, JsonObject json,
String removeItem) {
163     try {
164         for (Item it: products) {
165             if (it.getNumber().equals(removeItem)) {
166                 products.remove(it);
167
168                 json.addProperty("status", 1);
169                 json.addProperty("action", "REMOVE");
170                 json.addProperty("actionReq", removeItem);
171             }
172         }
173     } catch (Exception e) {
174         json.addProperty("status", 0);
175         json.addProperty("data", e.getMessage());
176     }
177 }
178
179 }
180
```