Order App Networking Standard:

Standard Version 2 (Version number 1):

Each packet begins with a header:

Version number: short  
The only current version number is 1, representing version 2 (0-indexed)

Packet Type: short

0 -> New connection/client requesting information. (client->server)

1 -> Order being sent (client->server)

2 -> Server sending information (server->client)

3 -> Server confirming reception of packet (server->client)

4 -> Client confirming reception of packet (client->server)

5 -> Client disconnecting from server (client->server)

Idempotency Token: int

The Idempotency Token is arbitrary, provided no two different packets on a connection from the same device use the same idempotency token twice in a row. If a packet is received with the same idempotency token as the previous packet, no action should be taken, but the return packet should be re-sent to the device that sent the packet with a duplicate Idempotency Token. This allows the client to re-try requests without risking causing duplicate actions.

Packet Body:

Packet Type 0:

Empty packet

Packet Type 1:

The customer name: String

The name of the customer, as entered into the app by the user. See below for how strings are sent.

The number of non-0 items: Short

The number of non-0 items being sent.

For every non-0 item:

The Item ID: Short

The Item ID is the ID (position in the list) of the item being ordered. This allows for data to be saved on typical orders, where the customer will not order more than half of the available items. This means that only the non-0 values need to be sent

The item quantity: Integer

The item quantity is the amount of the item ordered. This should never be 0, as it should be omitted from the packet.

Packet type 2:

Number of items: short

This is the number of items that are going to be sent

For as many times as there are items:

Item name: String

This is the name of the item, as a string. See below for how strings are sent.

Item quantity: Integer

This is the number of items available to be ordered.

Packet type 3 & 4:

Successful packet reception: Boolean

True if the packet was received correctly, false otherwise. If this is false, it indicates that the message was corrupted in some way and could not be read by the receiver. This indicates that the sender should send the packet again.

Packet type 5:

Empty packet

On reception of a type 0 packet, the server should send the information the client requires (Packet type 2) to the client, but not confirmation packet should be sent.

On reception of a type 2 packet, the client does not send a confirmation packet. If the client receives no data after sending a type 0 packet, the client should send the type 0 packet again, with the same idempotency token.

On reception of a type 1 packet, the server should process the order. Assuming that processing the order was successful, it should send a type 3 packet back to the client, with the same idempotency token as the type 1 packet received by the server, with true for successful packet reception. If processing the order fails, a false should be sent instead.

On reception of a type 3 or type 4 packet, the receiver of the packet should check the Boolean. If it is true, no action should be taken. If it is false, the idempotency token should be read, and the original packet sent by the client with that idempotency token should be repeated.

On reception of a type 5 packet, the server should forget the idempotency tokens received from that address. If another connection is established by the same device, it should act as if it is entirely new to the server.

Strings:

Strings are sent preceded by the length of the string being sent. So, a string consists of:

String length: Integer

This is the number of characters in the string.

For the number of times specified by length:

Character: char

A character in the string. The first character sent is the first character in the string, and so on.