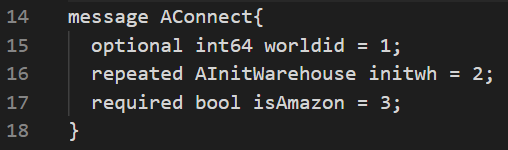
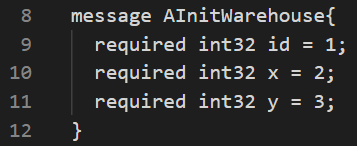
**Mini Amazon Server**

1. **Create the world and connect to the world**
2. Amazon -> World:
3. Create new world: But here we amazon create world ourselves first.
4. Amazon connection request: send an AConnect message with the worldid that you want and receive an AConnected response.
5. Initiate warehouse: (x,y) is the address of warehouse.

**Resquest:**



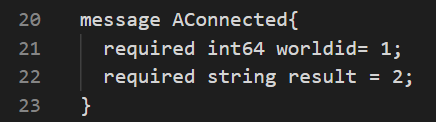






1. World -> Amazon: Amazon connect response: the result string in AConnected will be “connected!”, otherwise it will be an error message starting with “error:”.

**Response:**





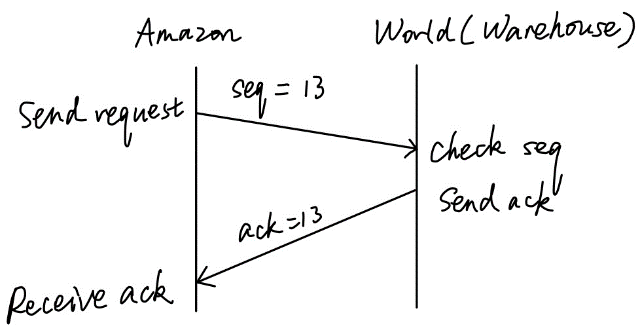
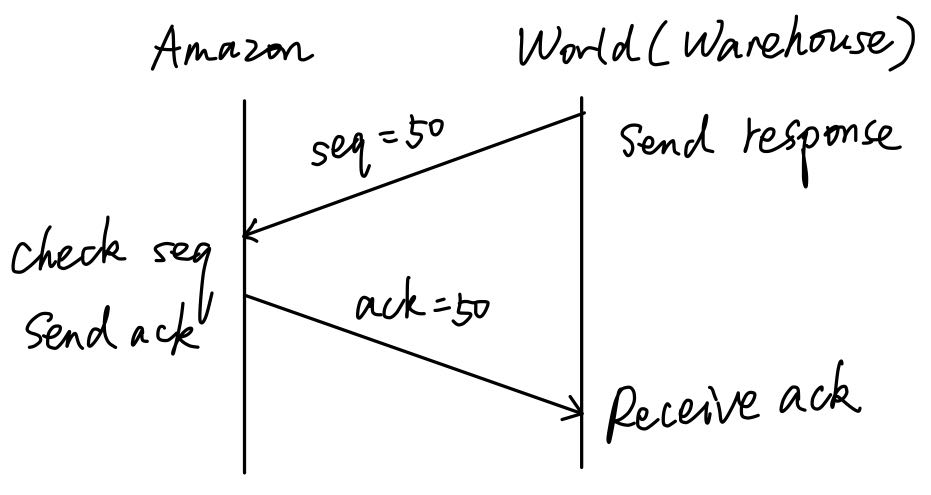
1. Ups -> Amazon: UPS send a Uconnect request without specifying a worldid number, the World would create one and return its ID in UConnected response (worldid). UPS tell amazon the worldid.

**Response:**

(Receive from Ups)

1. After connection, send ACommands and receive AResponse.
2. **Amazon Commands details: (note all commands include a sequence number for acknowledgement as described above)**

**Note: Acks process:**

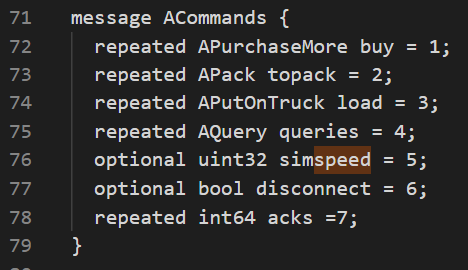
****

1. **Buy: Ask products to be delivered to a warehouse.**

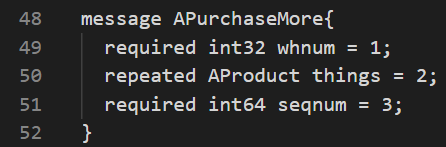
**Arrived: Get a notification that your orders have arrived.**

1. Amazon -> World: Specify item id, description (any text) and the quantity you want. Different descriptions for the same product id, the behavior is undefined.

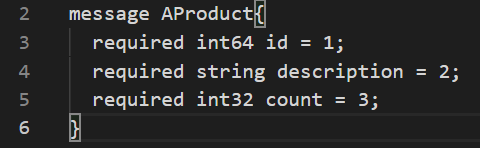
**Resquest:**







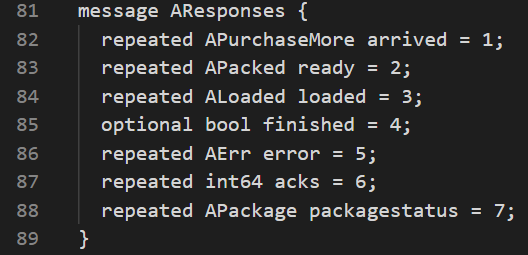




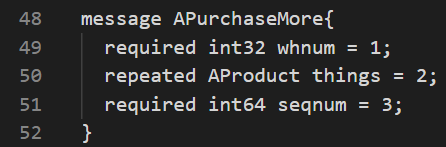


1. World -> Amazon: When you buy, you will later get a notification that your orders have arrived

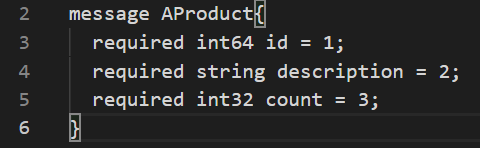
**Response:**













1. **Note: Topack and pickup can be sent at the same time!**

**Topack: Pack a shipment for delivery.**

**Ready: Notification that packing is complete**

1. Amazon -> Ups: When a BUY occurs and they recognize the item is in a warehouse, Amazon MUST send a request to UPS to bring a truck to the specified warehouse (along with the created packageID) with a sequence number until an acknowledgment is received.

**Request:**

(Send to Ups)

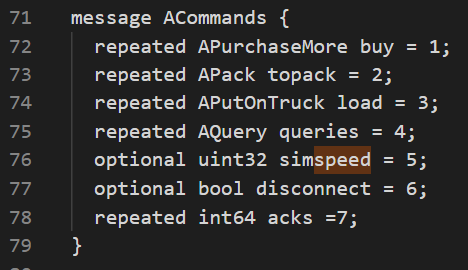
1. Ups -> Amazon: Send an ack, to tell Amazon it get the pickup message.

**Response:**

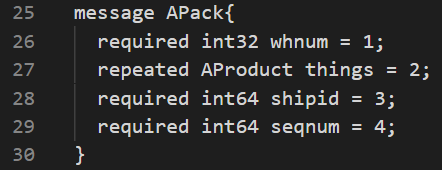
(Receive from Ups)

1. Amazon -> World: The warehouse that you request to pack the shipment MUST have sufficient inventory (and the inventory will be reduced accordingly).

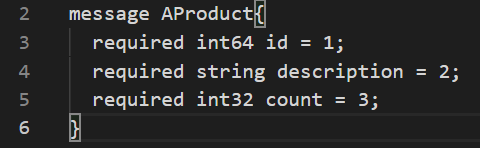
**Resquest:**







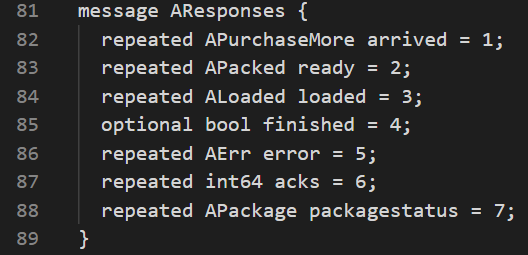




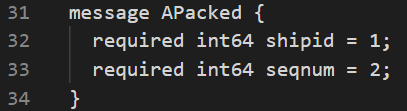


1. World -> Amazon: You will be notified when it is ready. Response shipid and seqnum.

**Response:**









1. **Note: Load only happens after package is ready and truck is arrived!**

**Load: Load a shipment on to a truck.**

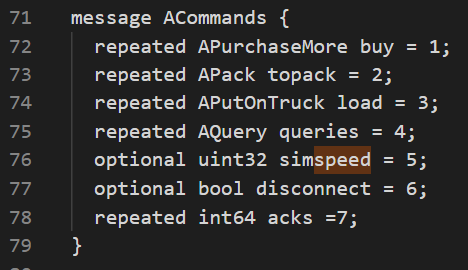
**Loaded: Notification that you have finished loading a shipment onto a truck**

1. Ups -> Amazon: Ups notify Amazon that the truck is arrived.

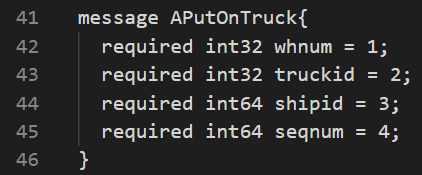
**Response:**

(Receive from Ups)

1. Amazon -> World: The shipment MUST be packed AND the truck MUST be at the warehouse, ready to receive the shipment (the shipper must have sent them to pickup and they must have received notification of completion).



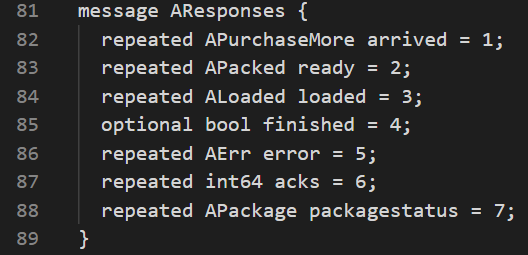




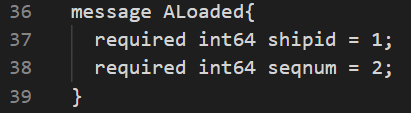


1. World -> Amazon: Notification that you have finished loading a shipment onto a truck.

**Response:**









1. Amazon -> Ups: Notification to Ups that the truck is loaded.

**Request:**

(Send to Ups)

1. Ups -> Amazon: If the package is sent, Ups notifies Amazon that the package is delivered.

**Response:**

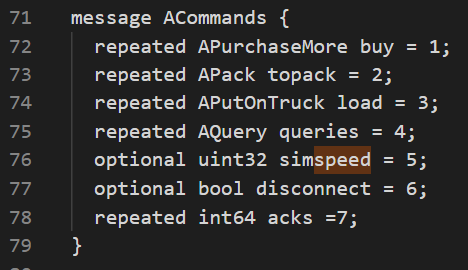
(Receive from Ups)

1. **Queries: ask the status of a package by specifying the packageid.**

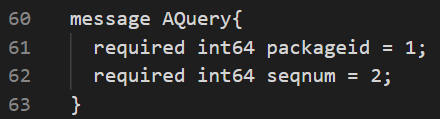
**Packagestatus: tell the current status of one package that you queried.**

1. Amazon -> World: Ask the status of a package by specifying the packageid.

**Resquest:**









1. World -> Amazon: Possible packagestatus: packing, packed, loading, loaded, delivering, delivered.

packing: Amazon told World topack, but World haven’t sent back Ready;

packed: Amazon told World topack, and World sent back ready;

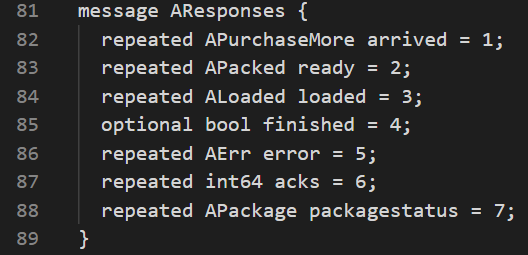
loading: Packed and Ups have told Amazon that truck is arrived, but World haven’t sent back loaded;

loaded: Packed and truck arrived, and World sent back loaded;

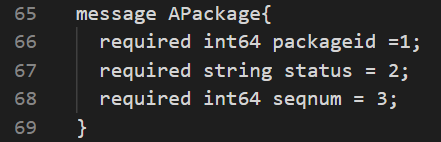
delivering: Loaded, Amazon told Ups to deliver and Ups ack;

delivered: Amazon receive notification from Ups that the package is delivered.

**Response:**



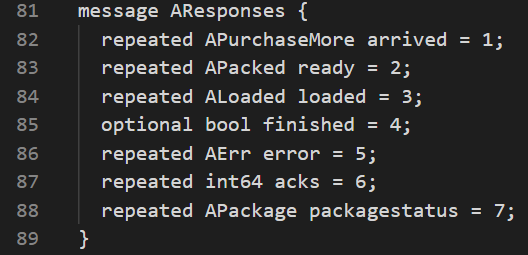




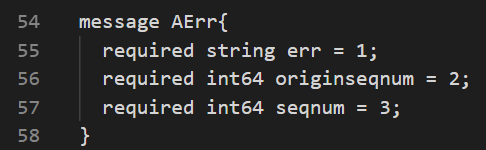


1. **Error: indicates that you failed to meet any of the MUST requirements specified at “Amazon Commands details” above.**
2. World -> Amazon: Read the err string carefully for more information.

**Response:**









1. Ups -> Amazon: Read the err from Ups.

**Response:**

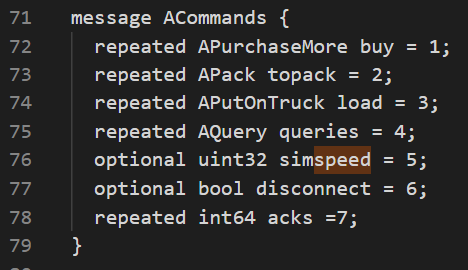
(Receive from Ups)

1. **Disconnect:**
2. To disconnect the server and amazon:

Amazon -> World: Acommands disconnect = true,

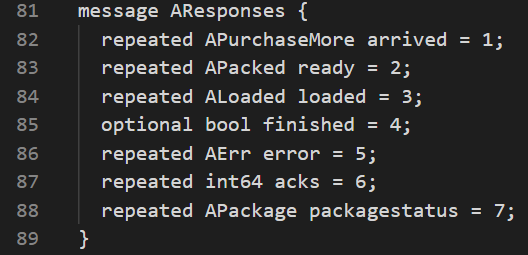
World -> Amazon: AResponse finished = true,

**Request:**





**Response:**





1. To disconnect the Ups and Amazon:

Amazon -> Ups:

Ups -> Amazon:

**Request:**

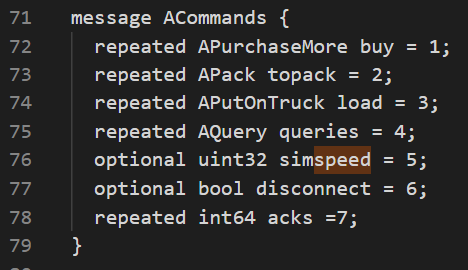
(Send to Ups)

**Response:**

(Receive from Ups)

1. **Simspeed:**
2. Adjust simulation speed for debug, default = 100, set high for many actions quickly.

**Request:**





**Mini Amazon Database**

1. **Account**

|  |  |  |  |
| --- | --- | --- | --- |
| user\_id | username | password | email |
| int | string | string | string |

1. **Product**

|  |  |  |  |
| --- | --- | --- | --- |
| product\_id | product\_name | description | amount |
| int64 | string | string | int32 |

1. **Warehouse**

|  |  |  |
| --- | --- | --- |
| warehouse \_id | x | y |
| int32 | int32 | int32 |

1. **Order**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| package\_id/ship\_id/Tracking Number | User\_id | product\_name | warehouse\_id | truck\_id  (optinal) | status | Address\_x | Address\_y |
| int64 | int32 | string | int32 | int32 | string | int32 | int32 |

**Mini Amazon Webpage**

1. Login
2. Register
3. Create order
4. Check order status