

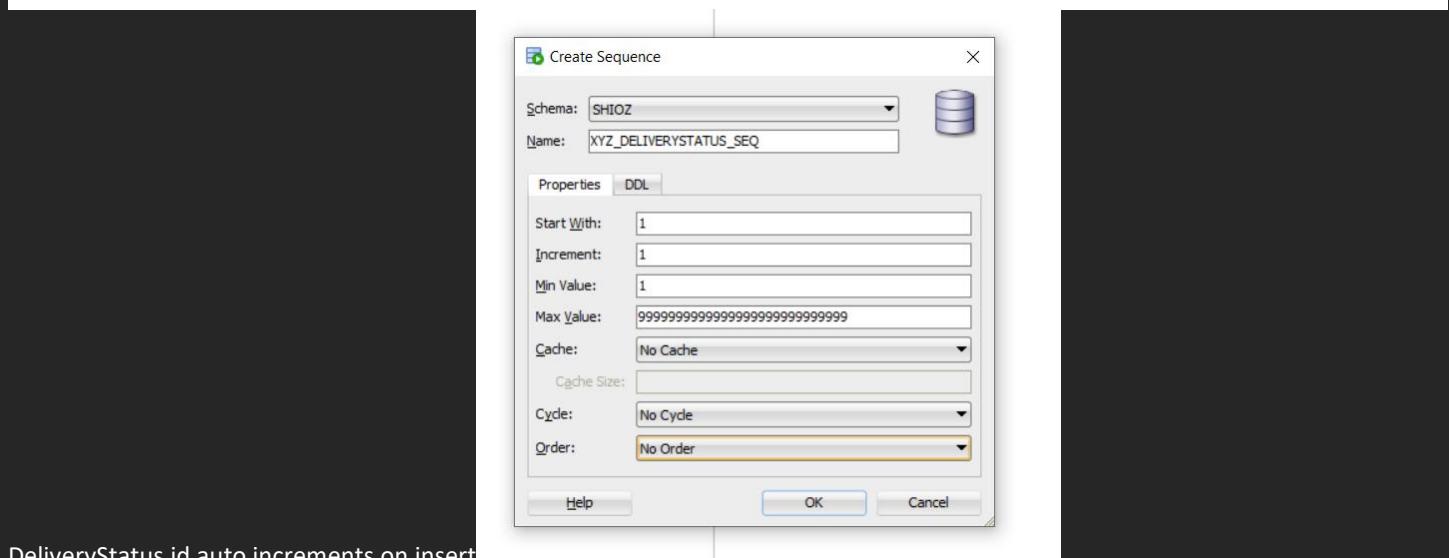
## Answers:

User credential:

“customer3”, “mercury”

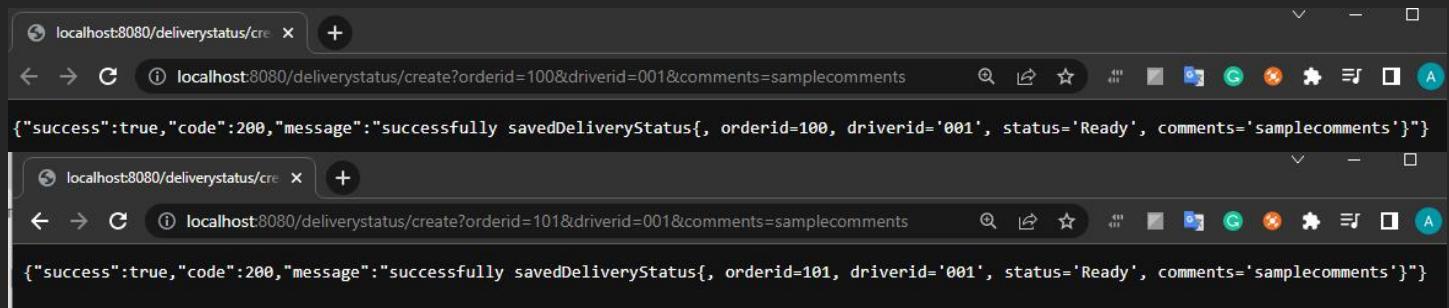
“driver4”, “mercury”

**2.1 Driver should be able to login to the system and input / create new delivery once the food is ready for delivery.**



DeliveryStatus.id auto increments on insert

Using GET method for easier debugging:



Row added with incrementing id:

Start Page × local × XYZ\_DELIVERYSTATUS ×

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

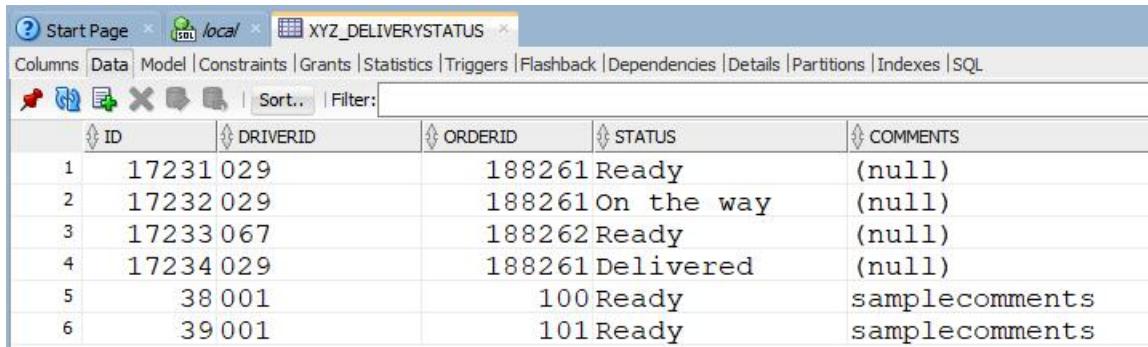
Sort.. Filter:

ID	DRIVERID	ORDERID	STATUS	COMMENTS
1	17231 029	188261	Ready	(null)
2	17232 029	188261	On the way	(null)
3	17233 067	188262	Ready	(null)
4	17234 029	188261	Delivered	(null)
5	38 001	100	Ready	samplecomments
6	39 001	101	Ready	samplecomments

### Requirement:

**Driver need to specify a valid orderid, a valid driverid, a valid status input. Type must not mismatch.**

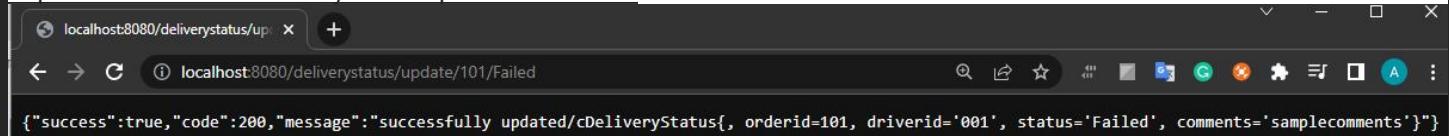
**2.2 Driver should be able to update the delivery status of a particular order, from "Ready" -> "On the Way" -> "Delivered" / "Failed", each update should be recorded into database.**



ID	DRIVERID	ORDERID	STATUS	COMMENTS
1	17231 029	188261	Ready	(null)
2	17232 029	188261	On the way	(null)
3	17233 067	188262	Ready	(null)
4	17234 029	188261	Delivered	(null)
5	38 001	100	Ready	samplecomments
6	39 001	101	Ready	samplecomments

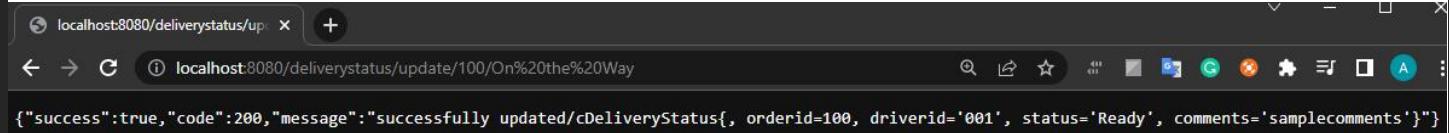
Db was

<http://localhost:8080/deliverystatus/update/101/Failed>



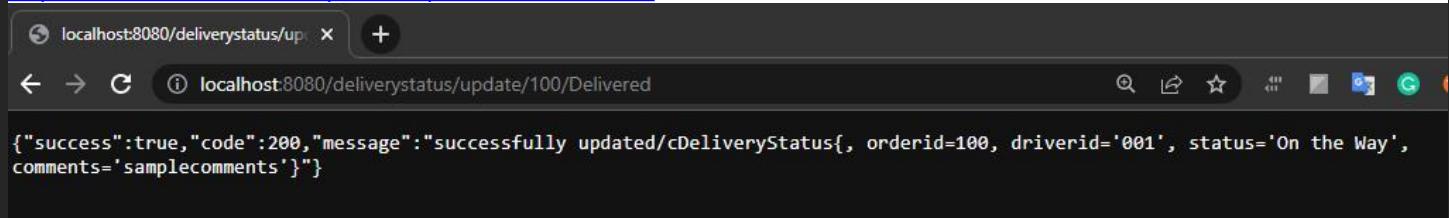
```
{"success":true,"code":200,"message":"successfully updated/cDeliveryStatus{, orderid=101, driverid='001', status='Failed', comments='samplecomments'}"}
```

<http://localhost:8080/deliverystatus/update/100/On%20the%20Way>



```
{"success":true,"code":200,"message":"successfully updated/cDeliveryStatus{, orderid=100, driverid='001', status='Ready', comments='samplecomments'"}
```

<http://localhost:8080/deliverystatus/update/100/Delivered>



```
{"success":true,"code":200,"message":"successfully updated/cDeliveryStatus{, orderid=100, driverid='001', status='On the Way', comments='samplecomments'"}
```

Start Page    local    XYZ\_DELIVERYSTATUS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

ID	DRIVERID	ORDERID	STATUS	COMMENTS
1	17231 029	188261	Ready	(null)
2	17232 029	188261	On the way	(null)
3	17233 067	188262	Ready	(null)
4	17234 029	188261	Delivered	(null)
5	38 001	100	Ready	samplecomments
6	39 001	101	Ready	samplecomments
7	40 001	101	Failed	samplecomments
8	41 001	100	On the Way	samplecomments
9	42 001	100	Delivered	samplecomments

Requirement: the orderid must be a valid one and exist in the table.

**3.1 Customer should be able to login to the system and search the order delivery status by inputting order Id. Customer should only get the current delivery status of this order.**

ID	DRIVERID	ORDERID	STATUS	COMMENTS
1	17231 029	188261	Ready	(null)
2	17232 029	188261	On the way	(null)
3	17233 067	188262	Ready	(null)
4	17234 029	188261	Delivered	(null)
5	38 001	100	Ready	samplecomments
6	39 001	101	Ready	samplecomments
7	40 001	101	Failed	samplecomments
8	41 001	100	On the Way	samplecomments
9	42 001	100	Delivered	samplecomments

Red arrow points to this order's latest DeliverStatus.id

<http://localhost:8080/deliverystatus/latest/188262>

localhost:8080/deliverystatus/latest/188262

```
{"id":17233,"orderid":188262,"driverid":"067","status":"Ready","comments":null}
```

<http://localhost:8080/deliverystatus/latest/188261>

localhost:8080/deliverystatus/latest/188261

```
{"id":17234,"orderid":188261,"driverid":"029","status":"Delivered","comments":null}
```

<http://localhost:8080/deliverystatus/latest/101>

localhost:8080/deliverystatus/latest/101

```
{"id":40,"orderid":101,"driverid":"001","status":"Failed","comments":"samplecomments"}
```

<http://localhost:8080/deliverystatus/latest/100>

localhost:8080/deliverystatus/latest/100

```
{"id":42,"orderid":100,"driverid":"001","status":"Delivered","comments":"samplecomments"}
```

Requirement: the orderid must be a valid one and exist in the table.

## Security part:

Logged in as customer:  
using  
“customer3”, “mercur

```
{"success":true,"code":200,"message":"Login successfully."}
```

```
Cutomer can check the latest order/delivery status  
← → C ⓘ localhost:8080/deliverystatus/latest/188262  
  
{"id":17233,"orderid":188262,"driverid":"067","status":"Ready","comments":null}
```

A screenshot of a browser window. The address bar shows the URL "localhost:8080/logout". The main content area displays a JSON response: {"success":true,"code":200,"message":"Logout successfully."}.

Logined as driver:

① localhost:8080/login

Please sign in

driver4
.....

```
{"success":true,"code":200,"message":"Login successfully."}
```

```
{"success":true,"code":200,"message":"successfully savedDeliverystatus{, orderid=201, driverid='001', status='Ready', comments='samplecomments'}"}  
Driver can create status
```

	42 001	100 Delivered	samplecomments
10	43 001	101 Ready	samplecomments
11	44 001	201 Ready	samplecomments

Driver can also update status

localhost:8080/deliverystatus/update/201/On%20the%20Way

```
{"success":true,"code":200,"message":"successfully updated/cDeliveryStatus{, orderId=201, driverId='001', status='Ready', comments='samplecomments'}"}
```

	45 001	101 Ready	samplecomments
11	44 001	201 Ready	samplecomments
12	45 001	201 On the Way	samplecomments

But driver does not have power to check latest status

localhost:8080/deliverystatus/latest/201

# Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tables created:

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' sidebar is open, showing various database objects like Functions, Operators, Queues, Triggers, Types, Sequences, and Materialized Views. In the center, a tab titled 'XYZ\_DELIVERYSTATUS\_SEQ' is active. Below it, the 'Details' tab is selected, displaying properties for the sequence. The properties listed are:

Name	Value
1 CREATED	03-JUN-22
2 LAST DDL TIME	03-JUN-22
3 SEQUENCE OWNER	SHIOZ
4 SEQUENCE NAME	XYZ DELIVERYSTATUS SEQ
5 MIN VALUE	1
6 MAX VALUE	99999999999999999999999999999999
7 INCREMENT BY	1
8 CYCLE FLAG	N
9 ORDER FLAG	N
10 CACHE SIZE	0
11 LAST NUMBER	43

The screenshot shows the 'XYZ\_DELIVERYSTATUS' table definition. The 'Columns' tab is selected. The table has five columns:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER	No	(null)	1	(null)
2 ORDERID	NUMBER	No	(null)	2	(null)
3 DRIVERID	VARCHAR2 (20 BYTE)	No	(null)	3	(null)
4 STATUS	VARCHAR2 (20 BYTE)	No	(null)	4	(null)
5 COMMENTS	VARCHAR2 (20 BYTE)	Yes	(null)	5	(null)

The screenshot shows the 'XYZ\_ORDER' table definition. The 'Columns' tab is selected. The table has four columns:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER	Yes	(null)	1	(null)
2 CUSTOMERID	VARCHAR2 (20 BYTE)	Yes	(null)	2	(null)
3 DETAIL	VARCHAR2 (200 BYTE)	Yes	(null)	3	(null)
4 COMMENTS	VARCHAR2 (200 BYTE)	Yes	(null)	4	(null)

The screenshot shows the 'XYZ\_USER' table definition. The 'Columns' tab is selected. The table has three columns:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 USERNAME	VARCHAR2 (100 BYTE)	No	(null)	1	(null)
2 PASSWORD	VARCHAR2 (100 BYTE)	No	(null)	2	(null)
3 ID	NUMBER	No	(null)	3	(null)

The screenshot shows two database tables in Oracle SQL Developer:

XYZ_USER_PROFILE	Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
	<input type="button" value="Actions..."/>												
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS								
1 ID	NUMBER	No	(null)	1	(null)								
2 TYPE	VARCHAR2 (100 BYTE)	No	(null)	2	(null)								

XYZ_USER_XYZ_USER_PROFILE	Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
	<input type="button" value="Actions..."/>												
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS								
1 ID	NUMBER	No	(null)	1	(null)								
2 USER_ID	NUMBER	No	(null)	2	(null)								
3 USER_PROFILE_ID	NUMBER	No	(null)	3	(null)								

@PreAuthorize("permitAll()")

Other requests that work for debugging():

<http://localhost:8080/orders/188261>  
<http://localhost:8080/orders/>  
<http://localhost:8080/deliverystatus>  
<http://localhost:8080/deliverystatus/17234>  
<http://localhost:8080/deliverystatus/orderid/188261>  
<http://localhost:8080/deliverystatus/orderid/188262>  
<http://localhost:8080/deliverystatus/deliveredorderid/188261>  
empty:  
<http://localhost:8080/deliverystatus/deliveredorderid/188262>