

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Threading.Tasks;
5
6 namespace DeliveryApp.Model
7 {
8     public class Delivery
9     {
10         public string Id { get; set; }
11         public string Name { get; set; }
12         public double OrginLong { get; set; }
13         public double OrginLat { get; set; }
14         public double DestLong { get; set; }
15         public double DestLat { get; set; }
16         /// <summary>
17         /// 0= waiting for pickup
18         /// 1= on the way
19         /// 2= delivered
20         /// </summary>
21         public int Status { get; set; }
22
23         public string DeliveryPersonId { get; set; }
24         public static async Task<List<Delivery>> GetDleivery()
25         {
26             List<Delivery> deliveries = new List<Delivery>();
27             deliveries = await AzureHelper.mobile.GetTable<Delivery>().Where      ↗
                (d=>d.Status!=2).ToListAsync();
28             return deliveries;
29         }
30         public static async Task<List<Delivery>> GetDelivered()
31         {
32             List<Delivery> deliveries = new List<Delivery>();
33             deliveries = await AzureHelper.mobile.GetTable<Delivery>().Where(d => ↗
                d.Status == 2).ToListAsync();
34             return deliveries;
35         }
36
37         public static async Task<List<Delivery>> GetDelivered(string UserID)
38         {
39             List<Delivery> deliveries = new List<Delivery>();
40             deliveries = await AzureHelper.mobile.GetTable<Delivery>().Where(d => ↗
                d.Status == 2 && d.DeliveryPersonId== UserID).ToListAsync();
41             return deliveries;
42         }
43
44         public static async Task<List<Delivery>> GetOnTheWayDelivery(string      ↗
            PersonId)
45         {
```

```
46     List<Delivery> deliveries = new List<Delivery>();
47     deliveries = await AzureHelper.mobile.GetTable<Delivery>().Where(d =>
48         d.Status == 1 && d.DeliveryPersonId== PersonId).ToListAsync();
49     return deliveries;
50 }
51
52 public static async Task<List<Delivery>> GetWaitingForDelivery()
53 {
54     List<Delivery> deliveries = new List<Delivery>();
55     deliveries = await AzureHelper.mobile.GetTable<Delivery>().Where(d =>
56         d.Status == 0).ToListAsync();
57     return deliveries;
58 }
59
60 public static async Task<bool> InsertDelivery(Delivery delivery)
61 {
62     return await AzureHelper.SaveData<Delivery>(delivery);
63 }
64
65 public static async Task<bool> PickedUpPackage(Delivery delivery, string
66     PersonId)
67 {
68     try
69     {
70         delivery.Status = 1;
71         delivery.DeliveryPersonId = PersonId;
72         await AzureHelper.mobile.GetTable<Delivery>().UpdateAsync
73             (delivery);
74         return true;
75     }
76     catch (Exception ex)
77     {
78         return false;
79     }
80 }
81
82 public static async Task<bool> DeliveredPackage(Delivery delivery)
83 {
84     try
85     {
86         delivery.Status = 2;
87         await AzureHelper.mobile.GetTable<Delivery>().UpdateAsync
88             (delivery);
89         return true;
90     }
91     catch (Exception ex)
92     {
93         return false;
94     }
95 }
```

```
90     }
91
92     public static async Task<bool> PickedUpPackage(string deliveryId, string
93         PersonId)
94     {
95         try
96         {
97             var delivery = (await AzureHelper.mobile.GetTable<Delivery>
98                 ().Where(d => d.Id == deliveryId).ToListAsync
99                 ()).FirstOrDefault();4ioklcv iokliokl34erw
100
101             delivery.Status = 1;
102             delivery.DeliveryPersonId = PersonId;
103             await AzureHelper.mobile.GetTable<Delivery>().UpdateAsync
104                 (delivery);
105             return true;
106         }
107         catch (Exception ex)
108         {
109             return false;
110         }
111     }
112
113     public static async Task<bool> DeliveredPackage(string deliveryId)
114     {
115         try
116         {
117             delivery.Status = 2;
118             await AzureHelper.mobile.GetTable<Delivery>().UpdateAsync
119                 (delivery);
120             return true;
121         }
122         catch (Exception ex)
123         {
124             return false;
125         }
126     }
127
128     public override string ToString()
129     {
130         return $"{Name} - {Status}";
131     }
132 }
```