

RabbitMQSample

RabbitMQ teknolojisinin nasıl kullanılacağına dair örnek bilgi amaçlı hazırlanmış bir konsol uygulaması

Table of Contents


Symbol Reference	1
RabbitMQSample Namespace	1
Classes	1
Kisi Class	1
Kisi Properties	2
Classes	2
Program Class	3
Program Methods	3
Program.Main Method	3
Program.RabbitMQKuyrugGonder Method	4
Program.RabbitMQKuyruktanAI Method	5
Files	6
Kisi.cs	6
Program.cs	7
RabbitMQ.sln	9
RabbitMQSample.csproj	9
Index	a

1 Symbol Reference

1.1 RabbitMQSample Namespace

This is namespace RabbitMQSample.

Classes


	Name	Description
	Kisi (see page 1)	RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde bulunduğu sınıftır

1.1.1 Classes

Summary

The following table lists classes in this documentation.

Classes

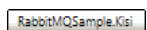
	Name	Description
	Kisi (see page 1)	RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde bulunduğu sınıftır

1.1.1.1 Kisi Class

Summary

RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde bulunduğu sınıftır

Class Hierarchy

 RabbitMQSample.Kisi

C#

```
public class Kisi;
```

File

Kisi.cs ([see page 6](#))

Kisi Properties

	Name	Description
	Adi (see page 2)	Kisinin Adi Bilgisi
	DogumTarihi (see page 2)	Kisinin Dogum Tarihi Bilgisi
	DogumYeri (see page 2)	Kisinin Dogum Yeri Bilgisi
	ID (see page 2)	Kisinin ID Bilgisi (Veri Tabanından geliyor gibi düşünebiliriz)
	Soyadi (see page 2)	Kisinin Soyadi Bilgisi

1.1.1.1.1 Kisi Properties

1.1.1.1.1.1 Kisi.Adi Property

Summary

Kisinin Adi Bilgisi

C#

```
public string Adi;
```

1.1.1.1.1.2 Kisi.DogumTarihi Property

Summary

Kisinin Dogum Tarihi Bilgisi

C#

```
public DateTime DogumTarihi;
```

1.1.1.1.1.3 Kisi.DogumYeri Property

Summary

Kisinin Dogum Yeri Bilgisi

C#

```
public string DogumYeri;
```

1.1.1.1.1.4 Kisi.ID Property

Summary

Kisinin ID Bilgisi (Veri Tabanından geliyor gibi düşünebiliriz)

C#

```
public int ID;
```

1.1.1.1.1.5 Kisi.Soyadi Property

Summary

Kisinin Soyadi Bilgisi

C#


```
public string Soyadi;
```

1.2 Classes

Summary

The following table lists classes in this documentation.

Classes

	Name	Description
	Program (see page 3)	Rabbit MQ uygulamasinin konsolda çalishmasini saglayan metotlari içeren temel siniftir.

1.2.1 Program Class

Summary

Rabbit MQ uygulamasinin konsolda çalısmasini saglayan metotlari içeren temel siniftir.

Class Hierarchy

Program




C#

```
public class Program;
```

File

Program.cs (see page 7)

Program Methods

	Name	Description
	Main (see page 3)	Konsol uygulamasinin çalıstigi ana metottur
	RabbitMQKuyruguaGonder (see page 4)	Ekrandan alınan sayı kadar Kisi olusturup Rabbit MQ kuyruguna göndermeyi ve gönderilen kisiler ile ilgili bilgileri Konsol ekranında göstermeye yarayan metottur
	RabbitMQKuyruktanAl (see page 5)	Rabbit MQ kuyrugunda biriken bilgilerin FIFO mantigina göre Kuyruktan alınarak Kisi bilgilerinin konsol ekranında görüntülenmesini saglayan metottur

1.2.1.1 Program Methods

1.2.1.1.1 Program.Main Method

Summary

Konsol uygulamasinin çalıstigi ana metottur

C#

```
public static void Main(string[] args);
```

Parameters

Parameters	Description
string[] args	Gönderilen Argüman Dizisi

Body Source

```
1: public static void Main(string[] args)
2: {
3:     var programKapansinMi = false;
4:     // Kullanici Programdan çıkmak için e veya E yazmadigi sürece çalısmasini saglayan döngüdür
5:     while (!programKapansinMi) {
6:         Console.WriteLine("RabbitMQ kuyruguna kaç tane kayıt göndermek istiyorsunuz?");
7:         var sayi = Console.ReadLine();
8:         // eger geçerli bir sayi girerse o sayi kadar Kisi kaydi olusturup
9:         // RabbitMQ kuyruguna yollayacak, alacak
10:        // Ve bu bilgileri konsol ekranında yazdiracak kod blogu burasidir
11:        if (int.TryParse(sayi, out int kisiSayisi))
```

```

12:         {
13:             RabbitMQKuyruGonder(kisiSayisi);
14:             Console.WriteLine("-----");
15:             Console.WriteLine("RabbitMQ kuyrugundan alma islemleri için ekrana bir
değer giriniz");
16:             Console.ReadLine();
17:             RabbitMQKuyruktanAl();
18:             Console.WriteLine("-----");
19:         }
20:     else
21:     {
22:         // Eger e veya E harfi girerse kullanıcı konsoldan
23:         // programdan çıkış yapılmasını sağlayacak kod blogudur
24:         if (sayi == "e" || sayi == "E")
25:             programKapanSinMi = true;
26:         // eger geçersiz bir sayı veya metin girerse gösterilecek hata mesajıdır
27:         else
28:             Console.WriteLine($"{sayi} geçerli bir sayı veya uygulama kapatma
istegi degildir");
29:     }
30: }
31: Console.WriteLine("Programı kullandığınız için teşekkürler. İyi günler");
32: Environment.Exit(0);
33: }

```

1.2.1.1.2 Program.RabbitMQKuyruGonder Method

Summary

Ekrandan alınan sayı kadar Kisi oluşturup Rabbit MQ kuyruguna göndermeyi ve gönderilen kişiler ile ilgili bilgileri Konsol ekranında göstermeye yarayan metottur

C#

```
private static void RabbitMQKuyruGonder(int kisiSayisi);
```

Parameters

Parameters	Description
int kisiSayisi	Kisi Sayısı Bilgisi

Body Source

```

1: private static void RabbitMQKuyruGonder(int kisiSayisi)
2: {
3:     for (int i = 0; i < kisiSayisi; i++)
4:     {
5:         // Bogus kütüphanesi kullanılarak sahte veri içeren kişi kayıtlarının
6:         // otomatik olarak oluşturulmasını sağlayan yapıdır
7:         var kisiOlusturucu = new Faker<Kisi>()
8:             .CustomInstantiator(f => new Kisi())
9:             .RuleFor(u => u.Adi, f => f.Name.FirstName())
10:            .RuleFor(u => u.Soyadi, f => f.Name.LastName())
11:            .RuleFor(u => u.DogumYeri, (f, u) => f.Address.City())
12:            .RuleFor(u => u.DogumTarihi, (f, u) => f.Person.DateOfBirth)
13:            .RuleFor(u => u.ID, f => new Random().Next());
14:
15:         var kisi = kisiOlusturucu.Generate();
16:
17:         // Bilgisayara kurulan localhost RabbitMQ Sunucusuna bağlantı için gerekli
bilgileri içeren nesnedir
18:         var factory = new ConnectionFactory() { HostName = "localhost" };
19:         using (IConnection connection = factory.CreateConnection())
20:         using (IModel channel = connection.CreateModel())
21:         {
22:             // burada coderserdar adında bir kanal açarak, onun içerisine ilgili kişi
kayıtlarının gönderilmesi için hazırlamaktadır
23:             // diğer parametreler de kullanılarak kanalın kullanımı özelleştirilebilir
24:             channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive:

```

```

false, autoDelete: false, arguments: null);
25:
26:         string message = JsonConvert.SerializeObject(kisi);
27:         // sinif serialize edilerek byte dizisi haline dönüştürülür
28:         var body = Encoding.UTF8.GetBytes(message);
29:
30:         // burada dönüştürülen byte dizisinin kanala gönderilmesi sağlanır
31:         channel.BasicPublish(exchange: "", routingKey: "coderserdar",
basicProperties: null, body: body);
32:
33:         Console.WriteLine($"Gönderilen kisi: Adi Soyadi: {kisi.Adi} {kisi.Soyadi}

```

1.2.1.1.3 Program.RabbitMQKuyruktanAl Method

Summary

Rabbit MQ kuyrugunda biriken bilgilerin FIFO mantigina göre Kuyruktan alınarak Kisi bilgilerinin konsol ekranında görüntülenmesini sağlayan metottur

C#

```
private static void RabbitMQKuyruktanAl();
```

Body Source

```

1: IFO mantigina göre
2: /// Kuyruktan alınarak
3: /// Kisi bilgilerinin konsol ekranında görüntülenmesini sağlayan metottur
4: /// </summary>
5: private static void RabbitMQKuyruktanAl()
6: {
7:     // Bilgisayara kurulan localhost RabbitMQ Sunucusuna bağlantı için gerekli
bilgileri içeren nesnedir
8:     var factory = new ConnectionFactory() { HostName = "localhost" };
9:     using (IConnection connection = factory.CreateConnection())
10:    using (IModel channel = connection.CreateModel())
11:    {
12:        // burada coderserdar adında bir kanal açarak, onun içerisine ilgili kisi
kayıtlarının gönderilmesi için hazırlamaktadır
13:        // diğer parametreler de kullanılarak kanalın kullanımı özelleştirilebilir
14:        channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive: false,
autoDelete: false, arguments: null);
15:
16:        // burada kuyruğu tüketecek bir consumer nesnesi oluşturulur ve tüketmeye başlar
17:        var consumer = new EventingBasicConsumer(channel);
18:        consumer.Received += (model, ea) =>
19:        {
20:            var body = ea.Body;
21:            var message = Encoding.UTF8.GetString(body.ToArray());
22:            Kisi kisi = JsonConvert.DeserializeObject<Kisi>(message);
23:            Console.WriteLine($"Adi Soyadi: {kisi.Adi} {kisi.Soyadi}
[{kisi.DogumYeri}]");
24:            Console.WriteLine("RabbitMQ ile tanistiniz. İyi günler.");
25:        };
26:
27:        // kanalın kuyruğu tüketerek ilgili bilgilerin konsolda gösterilmesi sağlanır
28:        channel.BasicConsume(queue: "coderserdar", autoAck: true, consumer: consumer);
29:
30:        Console.ReadLine()

```

1.3 Files

Summary

The following table lists files in this documentation.

Files

Name	Description
Kisi.cs (🔗 see page 6)	This is file Kisi.cs.
Program.cs (🔗 see page 7)	This is file Program.cs.
RabbitMQ.sln (🔗 see page 9)	This is file RabbitMQ.sln.
RabbitMQSample.csproj (🔗 see page 9)	This is file RabbitMQSample.csproj.

1.3.1 Kisi.cs

This is file Kisi.cs.

Body Source

```
1: ?namespace RabbitMQSample
2: {
3:     /// <summary>
4:     /// RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde
5:     /// </summary>
6:     public class Kisi
7:     {
8:         /// <summary>
9:         /// Kisinin ID Bilgisi (Veri Tabanından geliyor gibi düşünebiliriz)
10:        /// </summary>
11:        public int ID { get; set; }
12:        /// <summary>
13:        /// Kisinin Adi Bilgisi
14:        /// </summary>
15:        public string Adi { get; set; }
16:        /// <summary>
17:        /// Kisinin Soyadi Bilgisi
18:        /// </summary>
19:        public string Soyadi { get; set; }
20:        /// <summary>
21:        /// Kisinin Dogum Tarihi Bilgisi
22:        /// </summary>
23:        public DateTime DogumTarihi { get; set; }
24:        /// <summary>
25:        /// Kisinin Dogum Yeri Bilgisi
26:        /// </summary>
27:        public string DogumYeri { get; set; }
28:    }
29: }
```

Namespaces

Name	Description
RabbitMQSample (🔗 see page 1)	This is namespace RabbitMQSample.

1.3.2 Program.cs

This is file Program.cs.

Body Source

```

1: ?using Bogus;
2: using Newtonsoft.Json;
3: using RabbitMQ.Client;
4: using RabbitMQ.Client.Events;
5: using RabbitMQSample;
6: using System.Text;
7:
8: /// <summary>
9: /// Rabbit MQ uygulamasinin konsolda calismasini saglayan metotlari içeren temel
siniftir.
10: /// </summary>
11: public class Program
12: {
13:     /// <summary>
14:     /// Konsol uygulamasinin çalistigi ana metottur
15:     /// </summary>
16:     /// <param name="args">Gönderilen Argüman Dizisi</param>
17:     public static void Main(string[] args)
18:     {
19:         var programKapansinMi = false;
20:         // Kullanici Programdan çikmak için e veya E yazmadigi sürece çalismasini
saglayan döngüdür
21:         while (!programKapansinMi) {
22:             Console.WriteLine("RabbitMQ kuyruguna kaç tane kayıt göndermek
istiyorsunuz? (Programdan çikmak için E veya e yaziniz)");
23:             var sayi = Console.ReadLine();
24:             // eger geçerli bir sayi girerse o sayi kadar Kisi kaydi olusturup
25:             // RabbitMQ kuyruguna yollayacak, alacak
26:             // Ve bu bilgileri konsol ekraninda yazdiracak kod blogu burasidir
27:             if (int.TryParse(sayi, out int kisiSayisi))
28:             {
29:                 RabbitMQKuyrugayaGonder(kisiSayisi);
30:                 Console.WriteLine("-----");
31:                 Console.WriteLine("RabbitMQ kuyrugundan alma islemleri için ekrana bir
deger giriniz");
32:                 Console.ReadLine();
33:                 RabbitMQKuyruktanAl();
34:                 Console.WriteLine("-----");
35:             }
36:             else
37:             {
38:                 // Eger e veya E harfi girerse kullanici konsoldan
39:                 // programdan çikis yapilmasini saglayacak kod blogudur
40:                 if (sayi == "e" || sayi == "E")
41:                     programKapansinMi = true;
42:                 // eger geçersiz bir sayi veya metin girerse gösterilecek hata
mesajidir
43:                 else
44:                     Console.WriteLine($"{{sayi}} geçerli bir sayi veya uygulama kapatma
istegi degildir");
45:             }
46:         }
47:         Console.WriteLine("Programi kullandiginiz için tesekkürler. İyi günler");
48:         Environment.Exit(0);
49:     }
50:
51:     /// <summary>
52:     /// Ekrandan alinan sayi kadar Kisi olusturup
53:     /// Rabbit MQ kuyruguna göndermeyi ve gönderilen kisiler ile ilgili bilgileri

```

```

54:     /// Konsol ekranında göstermeye yarayan metottur
55:     /// </summary>
56:     /// <param name="kisiSayisi">Kisi Sayisi Bilgisi</param>
57:     private static void RabbitMQKuyruGonder(int kisiSayisi)
58:     {
59:         for (int i = 0; i < kisiSayisi; i++)
60:         {
61:             // Bogus kütüphanesi kullanılarak sahte veri içeren kisi kayitlarinin
62:             // otomatik olarak olusturulmasini saglayan yapidir
63:             var kisiOlusturucu = new Faker<Kisi>()
64:                 .CustomInstantiator(f => new Kisi())
65:                 .RuleFor(u => u.Adi, f => f.Name.FirstName())
66:                 .RuleFor(u => u.Soyadi, f => f.Name.LastName())
67:                 .RuleFor(u => u.DogumYeri, (f, u) => f.Address.City())
68:                 .RuleFor(u => u.DogumTarihi, (f, u) => f.Person.DateOfBirth)
69:                 .RuleFor(u => u.ID, f => new Random().Next());
70:
71:             var kisi = kisiOlusturucu.Generate();
72:
73:             // Bilgisayara kurulan localhost RabbitMQ Sunucusuna baglanti için gerekli
74:             // bilgileri içeren nesnedir
75:             var factory = new ConnectionFactory() { HostName = "localhost" };
76:             using (IConnection connection = factory.CreateConnection())
77:             using (IModel channel = connection.CreateModel())
78:             {
79:                 // burada coderserdar adında bir kanal açarak, onun içerisine ilgili
80:                 // kisi kayitlarinin gönderilmesi için hazırlamaktadır
81:                 // diger parametreler de kullanılarak kanalın kullanımı
82:                 // özelleştirilebilir
83:                 channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive:
84:                 false, autoDelete: false, arguments: null);
85:
86:                 string message = JsonConvert.SerializeObject(kisi);
87:                 // sinif serialize edilerek byte dizisi haline dönüştürülür
88:                 var body = Encoding.UTF8.GetBytes(message);
89:
90:                 // burada dönüştürülen byte dizisinin kanala gönderilmesi sağlanır
91:                 channel.BasicPublish(exchange: "", routingKey: "coderserdar",
92:                 basicProperties: null, body: body);
93:
94:                 Console.WriteLine($"Gönderilen kisi: Adi Soyadi: {kisi.Adi}
95:                 {kisi.Soyadi} Dogum Tarihi: {kisi.DogumTarihi.ToShortDateString()}");
96:                 Console.WriteLine((i + 1) + ". kisi gönderildi...");
97:             }
98:         }
99:     }
100:     /// <summary>
101:     /// Rabbit MQ kuyrugunda biriken bilgilerin FIFO mantigina göre
102:     /// Kuyruktan alinarak
103:     /// Kisi bilgilerinin konsol ekranında görüntülenmesini saglayan metottur
104:     /// </summary>
105:     private static void RabbitMQKuyruktanAl()
106:     {
107:         // Bilgisayara kurulan localhost RabbitMQ Sunucusuna baglanti için gerekli
108:         // bilgileri içeren nesnedir
109:         var factory = new ConnectionFactory() { HostName = "localhost" };
110:         using (IConnection connection = factory.CreateConnection())
111:         using (IModel channel = connection.CreateModel())
112:         {
113:             // burada coderserdar adında bir kanal açarak, onun içerisine ilgili kisi
114:             // kayitlarinin gönderilmesi için hazırlamaktadır
115:             // diger parametreler de kullanılarak kanalın kullanımı özelleştirilebilir
116:             channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive:
117:             false, autoDelete: false, arguments: null);
118:
119:             // burada kuyruğu tüketecek bir consumer nesnesi olusturulur ve tüketmeye
120:             // baslar
121:             var consumer = new EventingBasicConsumer(channel);


```

```

113:         consumer.Received += (model, ea) =>
114:         {
115:             var body = ea.Body;
116:             var message = Encoding.UTF8.GetString(body.ToArray());
117:             Kisi kisi = JsonConvert.DeserializeObject<Kisi>(message);
118:             Console.WriteLine($"Adi Soyadi: {kisi.Adi} {kisi.Soyadi}
119: [{kisi.DogumYeri}]");
120:             Console.WriteLine("RabbitMQ ile tanistiniz. Iyi günler.");
121:         };
122:         // kanalın kuyruğu tüketerek ilgili bilgilerin konsolda gösterilmesi
123:         channel.BasicConsume(queue: "coderserdar", autoAck: true, consumer:
124:         consumer);
125:         Console.ReadLine();
126:     }
127:     Console.WriteLine("Kuyrugun tamamı tüketildi ve gösterildi. Devam etmek için
128:     Console.ReadLine();
129:     }
130: }

```

Classes

	Name	Description
	Program (see page 3)	Rabbit MQ uygulamasının konsolda çalışmasını sağlayan metotları içeren temel sınıftır.

1.3.3 RabbitMQ.sln

This is file RabbitMQ.sln.

1.3.4 RabbitMQSample.csproj

This is file RabbitMQSample.csproj.

Index

RabbitMQSample.Kisi.DogumYeri 2

RabbitMQSample.Kisi.ID 2

RabbitMQSample.Kisi.Soyadi 2

C

Classes 2

F

Files 6

K

Kisi class 1

about Kisi class 1

Adi 2

DogumTarihi 2

DogumYeri 2

ID 2

Soyadi 2

Kisi.cs 6

P

Program 3

Program class 3

about Program class 3

Main 3

RabbitMQKuyrugGonder 4

RabbitMQKuyruktanAl 5

Program.cs 7

Program.Main 3

Program.RabbitMQKuyrugGonder 4

Program.RabbitMQKuyruktanAl 5

R

RabbitMQ.sln 9

RabbitMQSample 1

RabbitMQSample namespace 1

Classes 1

RabbitMQSample.csproj 9

RabbitMQSample.Kisi 1

RabbitMQSample.Kisi.Adi 2

RabbitMQSample.Kisi.DogumTarihi 2