



## Faça como eu fiz: publicando o restaurante

Precisamos finalizar o código para nossa classe `RabbitMqClient` a fim de publicar os restaurantes criados. Inicialmente, altere o construtor da classe `RabbitMqClient` :

```
public RabbitMqClient(IConfiguration configuration)
{
    _configuration = configuration;
    _connection = new ConnectionFactory() { HostName =
    _channel = _connection.CreateModel();
    _channel.ExchangeDeclare(exchange: "trigger", type
```

[COPIAR CÓDIGO](#)

Adicione as informações de conexão com o RabbitMQ em seu `appsettings.json` de seu `RestauranteService` :

```
"RabbitMqHost": "rabbitmq-service",
"RabbitMqPort": "8200"
```

[COPIAR CÓDIGO](#)

Implemente a lógica de publicação de um Restaurante dentro do método `PublicaRestaurante` na classe `RabbitMqClient` :

```
public void PublicaRestaurante(RestauranteReadDto restaurantel
{
```

```
string mensagem = JsonSerializer.Serialize(restaurante);  
var body = Encoding.UTF8.GetBytes(mensagem);  
  
_channel.BasicPublish(exchange: "trigger",  
    routingKey: "",  
    basicProperties: null,  
    body: body  
);  
  
}
```

[COPIAR CÓDIGO](#)

< >

Não esqueça de fazer a injeção do serviço em seu `Program.cs` do projeto `RestauranteService` adicionando a linha:

< >

```
builder.Services.AddSingleton<IRabbitMqClient, RabbitMqClient>();
```

[COPIAR CÓDIGO](#)