October 8 2015

1. First find the view\_count\_instance of the dicks sporting good

select \* from view\_count\_instances order by client\_code

1. Get the count instance id from the query and store number from the dart web to display the store id

select store\_id,\* from view\_stores where count\_instance\_id = 17 and store\_number = 66666

1. After you get the store id from the above query run this query to find data that should be displayed in that particular dashlet.

select \* from view\_totals\_summary where store\_id = 75

1. How to Update the data in Sql

begin transaction

update item\_xref set xref\_barcode = '123aBc' where item\_xref\_id = 1

commit

1. How to Get the item\_xref:

Select top 100 \* from item\_xref

1. **How to test of the getItemsxref is case sensitive or no :**

Put a breakpoint in line 231 in correction repository, run the program

Select the Fanzz in the selection

Click the store no 66666

Modify inventory

Enter the store number 1

Add another Barcode

And move away the mouce and it should hit the break point

Do F10 and go through the list and if the item is already in the db it will return 1 and if it is not in the db it will return 0

1. **How to test Unresolved Invalid Barcodes:**
2. Run the Dartweb against the Test Environment
3. Select Michael Kors USA
4. Select store 7004
5. Click Count Shortcut
6. Under the Store Setup scroll to the right to the unresolved Invalid Barcodes

Click the number and you should see a popup window “Unresolved Invalid Barcodes for store 70004”

Under the Resolution, select the override from the dropdown

You need to get a valid barcode and paste it here. (see below to get a valid barcode)

Make sure to put the breakpoint on line 161 method “checkOverrideBarcode() in InvalidBarcodesModel.cs class

**How to get a valid barcodes:**

1. Select Micael cores and click stores
2. Click the store 66667
3. Click a number under the Areas Counted column
4. You will see a pop up window Areas counted for store 66667 for offsite no
5. Click a number under the Area
6. You should see another window “Items counted for store 66667 ….”
7. Copy the Barcode that has description, First scan, style, color, dept populated on the table. If the barcode is missing the above field it may not a correct barcode.

**How to test some methods:**

Before you test a feature, first need to figure out when it is done. Is it done at the Pre-Count or PostCount level.

**How Test Modify Inventory:**

1. Select the Dicks sporting goods and select the store number 33333,
2. Make sure that the Count shortcut is selected
3. Click modify inventory
4. Enter the fixture number 1
5. Enter a numerical associate number like 2546 instead of email
6. Type the reason
7. Enter a new amount under the new units
8. Hit save
9. If you put a breakpoint on SaveModifyInventory() on line 191 on ModifyInventoryModel.cs class. It should get hit.
10. There are times it will time out due to the call to the webApi

response = communicationService.TransmitChange(transmitMessage);

If you go to the web.config and check the WebApi address in line 113

<add key="webApiAddress" value="https://devapi.datascan.com/api/" />

You can copy the above url and paste it in the browser to see if it is running. If the browser is not working then there is a problem with the API not the code in the ModifyInventory or in the longbow.

**October 12, 2015**

**Comparison:**

If it is a sql comparison it is case insensitive but if it is a C# or Linq comparision, it is case sensitive.

For Sql use this syntax: join ix in this.ImportViewItemXrefs on i.xref\_barcode.ToLower() equals ix.xref\_barcode.ToLower() into ixJoin

For LInq Syntax:

var viewItemXref = viewItemXrefs.FirstOrDefault(c => c.xref\_barcode.Equals(xrefBarcode, StringComparison.OrdinalIgnoreCase));

**Source Tree:**

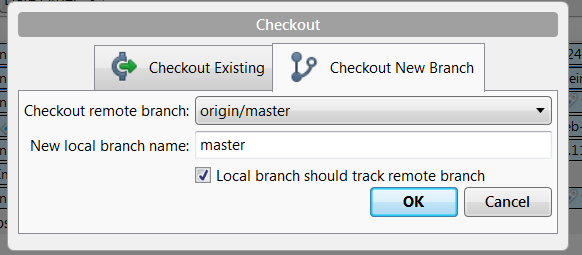
In order to create an feature you need to initialize it first:

**How to initialize the master branch:**

Expand the origin, under Remotes:

Right Click Master

Checkout origin/Master



Make sure it matches with the above settings

Click OK.

After this make sure to double click the develop branch and work on it.

**How to check out a branch?**

Right click the branch and click checkout. For eg. Right click release and click checkout.

**What to do if you started working but someone else already pushed changes and you are behind in git**

First commit your changes, then pull, which will merge others push with your commit

**How to pass command line parameters in a debug mode in Visual Studio?**

Right click the project and click properties

Click Build

Enter arguments in the “Command Line Arguments”

This way you do not have to enter any of these.

**How to make a C# Project to run on 32 bit or 64 bit solutions?**

Right click the project, properties and select the Build

Configuaration on the top left : select to All configuaration

Platform Target to 64.

Let’s say you have 5 different projects in your solutions, some are targeting 32 bit and others are targeting 64 bit, it will throw an error. You will need to change to all the projects .

**Stream:**

Read (byte[]buffer, int offset, int count)

Buffer: the array to store the read data

Offset: Buffer off set at which to start placing read data.

Count: represents the max no of bytes to read. Actually it may be less.

**How Reading works:**

The proper way to read data

First instantiate the data byte[] dataToLoad = new byte[stream.Length]

Int totalBytesRead = 0;

Int chunkbybytesread = 1;

While(totalBytesRead<dataToRead.Length && chunkBytesRead >0)

{

chunkBytesRead = stream.Read(dataToRead, totalBytesRead, dataToRead.Length-totalBytesRead);

totalBytesRead = totalBytesRead+chunkBytesRead;

return datToLoad;

}

**Using File Class to instantiate a stream : 3 ways**

**OpenRead:**

FileStream fs = File.openRead(@”c:myfiles\test.txt”

**If the file does not already exist:**

OpenWrite: Creates the file and returns a write-only stream

FileStream fs = File.OpenWrite(C:myfiles\data.txt”)

**If the file already exists:**

Create: truncates existing content

OpenWrite: leaves existing content and sets position to 0

Using Filestream constructors to instantiate the stream:

15 constructor overloads

FileStream fs = new FileStream(string path);

**FileMode:**

CreateNew - if a file already exists an exception is thrown

Create: creates a new file if a file already exists it is overwritten

Open : opens the existing file , if a file not found exception is thrown

OpenorCreate: Opens an existing file if the file does not exist, an exception is thrown

Truncate : open a file and deletes the file content. It is set to zero bytes

Append: Opens a file and sets the position to the end of the file, if a file does not exist, it is created.

**File Access:**

By default, File Stream will open a file in read and write access mode. With exception to the FileMode.Append option.

Read: File can only be read

Write : can only be written to

Read/Write: both allowed

**FileShare:**

Files gets locked by a FileStream until the stream is closed

Fileshare Enumeration can change this behavior

None: Sharing is not allowed by Default

Delete: Subsequest streams can delete the file

Inheritable: File handle can be inherited by child processes

Read: Stream can open file for reading only

Write: Other Streams are allowed to open for writing only

ReadWrite: other streams can open for both read and write

Be Careful when you allow other streams to open the file , it is different from the multiple threading .

**How to pass parameter via command line:**

-i "C:\Source\Dart.FixtureMessageDateChanger\Dart.FixtureMessageDateChangerCLI\bin\Debug\V2PublishFixtureMessage.json" -d "10/19/2018 12:00" -o "C:\Source\Dart.FixtureMessageDateChanger\Dart.FixtureMessageDateChangerCLI\bin\Debug\WriteFile.json" –w

**Why Command Line Parser ?**

Command Line parser is used to read arguments that is being passed via command line. There is a nuget library called CommandLineParser.

Few tricks :

For Boolean: You do not pass true or false . If you need to pass true, pass it like –w , -w is the option from the options class. If you want to pass false, leave the parameter blank. Make sure to annotate this value “Required=False” in the options class.

For String: If you pass – in front of a string for eg : -i “-test.txt”. It won’t go inside the if statement. It does not throw an exception, you should log the error in the catch block as below

if (CommandLine.Parser.Default.ParseArguments(args, options))

{// do something

}else {

Log.logger.Error(“Invalid format or missing the parameter”);

}

**How to implement serilog and configure it to write to the seq server ?**

1. First download Serilog, SerilogMetrics, Serilog seq and Serilog console via nuget package. If they are not available in VS go to the Nuget.org and get download them via package manager console.
2. On the Web program or Exe or class with Main method. You need to copy and paste right after the main method:

var environment = ConfigurationManager.AppSettings["Environment"] ?? "Missing";

Log.Logger = new LoggerConfiguration().ReadFrom.AppSettings()

                 .Enrich.WithMachineName()

                 .Enrich.WithProcessId()

                 .Enrich.WithThreadId()

                 .Enrich.FromLogContext() // Required for Context Logging as described below.

                 .Enrich.WithProperty("Environment", environment)

                 .WriteTo.LiterateConsole()

                 .CreateLogger();

1. Go to the app.config file and paste this inside the configuration :

<appSettings>

<add key="Environment" value="Dev"/>

<add key="serilog:minimum-level" value="Debug"/>

<add key="serilog:using" value="Serilog.Sinks.Seq" />

<add key="serilog:write-to:Seq.serverUrl" value="http://192.168.90.12" />

<add key="serilog:write-to:Seq.apiKey" value="api from seq server" />

</appSettings>

1. For the ApiKey , go to the seq server as admin ( username admin and password is :scanner\*1. Go to settings, Api key, Add Api key

Enter the Title and leave everything blank. Click Save Changes.

Go back and under the applied properties

AppName = Name of the app and click Add

Key = copy the token from top and paste it here and click add

1. Configuration is done.

For the Library. You need to install the Liblog instead.

Download liblog via nuget in vs

And initialize it :

private ILog logger = LogProvider.GetCurrentClassLogger();

You can use it by

try

{

   // some code

}

catch (Exception ex)

{

   this.logger.ErrorException("Error saving store {store\_id}", ex, model.storeId);

}

**Try Parse Vs Parse**

DateTimeOffset date;

if (DateTimeOffset.TryParse(options.ChangedDate, out date)) {

// do something here

}

else

{

Log.Logger.Error("Invalid Date format, should be mm/dd/yy hh:mm");

}

You should try to use tryparse instead of parse method.

Parse : Throws an exception if it cannot parse the value.

TryParse: Does not just try/catch internally, whole point is that it implements without an exception so that it is fast.

When to pick Parse vs TryParse: If you are sure that the value will be valid, use parse otherwise use TryParse

**Notes on Serilog:**

Minium Level:

Verbose: way too much detail, rarely enabled in production.

Debug: Used for internal system events, mostly to log when a method is called, helps to determine how something happened.

Information: describe thinkg happening in the system

Warning: When service degraded or endangered.

Error: Functionality is unavailable or expectations broken

Fatal: Most critical level, demand immediate attention

**Sample logging:**

Log.Warning("Disk quota {Quota} MB exceeded by {User}", quota, user);

Disk quota 1024 MB exceeded by "nblumhardt"

Assuming quota is an integer and user as in string. String is displayed in the quotes.

**Message Template Syntax:**

1. Property names are written between { and } brackets
2. Brackets can be escaped by doubling them {{ will be rendered as {
3. Formats that use numeric property like {0} and {1}, will be matched with the log methods parameters by treating the property name as indexes this is identical to string.format()’s behavior.

Don’t:

Log.Information(“The time is : “ + DateTime.Now);

Do:

Log.Information(“The time is {Now}”, DateTime.Now);

**Git Bash:**

First create a repo in github, go to your folder and right click and run git bash window

Git init

Git add .

Git commit –m “First Commit”

**(Only for the first time otherwise skip to push)**

Git remote add origin “https://github.com/padam10/test.git --

Git push –u origin master

*Sometime you get an error because the master may be ahead of you, if it is the case you need to pull first and push again.*

Git pull origin master – to pull

………………………………………………………..

Git diff - see what was changed

**GitFlow:**

First create a new feature

Make any changes to the code

Commit

Close the feature

Push to develop branch – be careful here.

**How Merger in Git ? ( for eg merge develop to release)**

1. Check out the branch that you would like to merge to (release in this case)
2. Right click the branch that you are merging from (Develop in this case)
3. Merge develop into current branch.

**GitFlow:**

**Asp.net Web API**

**Representation State Transfer(REST)**

Rest is a way of creating client server communication that is stateless.

**Difference between Model vs Viewmodel**

Check this link: <http://stackoverflow.com/questions/11064316/what-is-viewmodel-in-mvc>

A view model represents only the data that you want to display on your view/page, whether it be used for static text or for input values (like textboxes and dropdowns).

For example: You have an employee class

public class Employee : IEntity

{

public int Id { get; set; } // Employee's unique identifier

public string FirstName { get; set; } // Employee's first name

public string LastName { get; set; } // Employee's last name

public DateTime DateCreated { get; set; } // Date when employee was created

}

View models differs from domain models in that view models only contain the data that you are going display in the view. For example, you do not want the id to set from the view, it is probably auto generated by the Employee table. Also DateTime may be set in the store procedure.

public class CreateEmployeeViewModel

{

public string FirstName { get; set; }

public string LastName { get; set; }

}

**DbContext and Dbset:**

Normally DbContext corresponds to the Database and Dbset corresponds to the table.

You will be using DbContext object to get access to your tables and views(which will be represented by DbSet’s) and you will be using your DbSet’s to get access , create, update, delete and modify your table data.

Let’s say you have 10 tables in your database and your application works with 5 of them. Let’s call them Table1 to Table5. It makes sense to access it using a MyAppContext object where MyAppContext class is defined as below;

public class MyAppContext : DbContext

{

public MyAppContext () : ;

public DbSet<Table1> Table1 { get; set; }

public DbSet<Table2> Table2 { get; set; }

public DbSet<Table3> Table3 { get; set; }

public DbSet<Table4> Table4 { get; set; }

public DbSet<Table5> Table5 { get; set; }

}

public class Table1

{

public int Id {get; set;}

public string AStringField {get; set;}

//etc.

}

**Http Fundamentals:**

Uniform Resource Locator : <http://news.google.com> , <http://news.yahoo.com>

<http://food.com/receipe/grilled-cauliflower>

**http**: is the url scheme

**food.com**: host – tells which computer Ip address ( DNS server)

**/receipe/grilled-cauliflower**: Url path. It is like the file path – grilled cauliflower is more useful for search engine than it is for the file system.

IIS : Internet information Server: Web server that runs on windows machine

**Types of URLs:**

<http://food.com:80/receipes/squash>

80 is the port number, it is a default and so you do not need to include the port number in the url. If the server is listening other than the port 80, then you would need to include the port number at the url. Most commercial do not want to include the port number because it is hard to remember and long.

<http://bing.com/search?q=apple>

http: url scheme, bing.com: host, search : url path, q=apple : is a query or **query string.**

Web site can take this information and need to interpret. It is mostly name value pair.

<http://wikipedia.org/wiki/jabuticaba#Description>

http: url scheme, Wikipedia.org = host, wiki/jabutica: url path, **Description**: Fragment

Fragment is not processed by the server, it only used in the client.

**URL Encoding:**

Unsafe character should not show up in a url. For e.g. # or space. They may create error.

**Content Types:**

**Common MIME types**

**Type/Subtype Description**

Application/atom+xml Atom Feed

Application/json JSON Data

Image/gif GIF image

Image/png PNG image

Text/xml XML

Text/html HTML

Text/plain Just text

**Http Messages:**

Two types of Messages:

Http Request and Http Response

Standards are defined on what is sent and what is received so that the both client and server understand and full the request.

**HTTP Request :** Primarily Get and Post is used for HTML.

**Get – for Retrieve**

**Post : Update a resource -**

Put: Store a resource

Delete: Remove a resource

Head: Retrieve the header for a resource.

**Get :** is considered as safe method because you only retrieve the data from the server

**Post**: is considered as unsafe method for eg like cc process, send order etc. It may change the state.

Due to the nature of their request web server treats Gets and Post differently.

To avoid multiple post, Post/Redirect/Get is used extensively now a days.

@Request.QueryString[.

Searching is inheritably a safe operation

HttpPost is done through message so it is not so safe operation.

**Request Messages:**

[method] [URL] [Version]

[headers] – Generally contain server information. Like language acceptance etc. Date header. Header is optional but when it appears it must follow the protocol.

[body]

**Common Request Headers:**

**Header Description**

Refer The URL of the referring Page

User-Agent Information of the browser

Accept Preferred media types

Accept-Language Preferred Language

Cookie Cookie information

If-modified-Since Date of last retrievel

Date Create timestamp for the message

**Response Message:**

[version][status][reason]

[headers]

[body]

Status Code Categories:

Range Category

100 -199 Information;

200 -299 Successful

300-309 Redirections

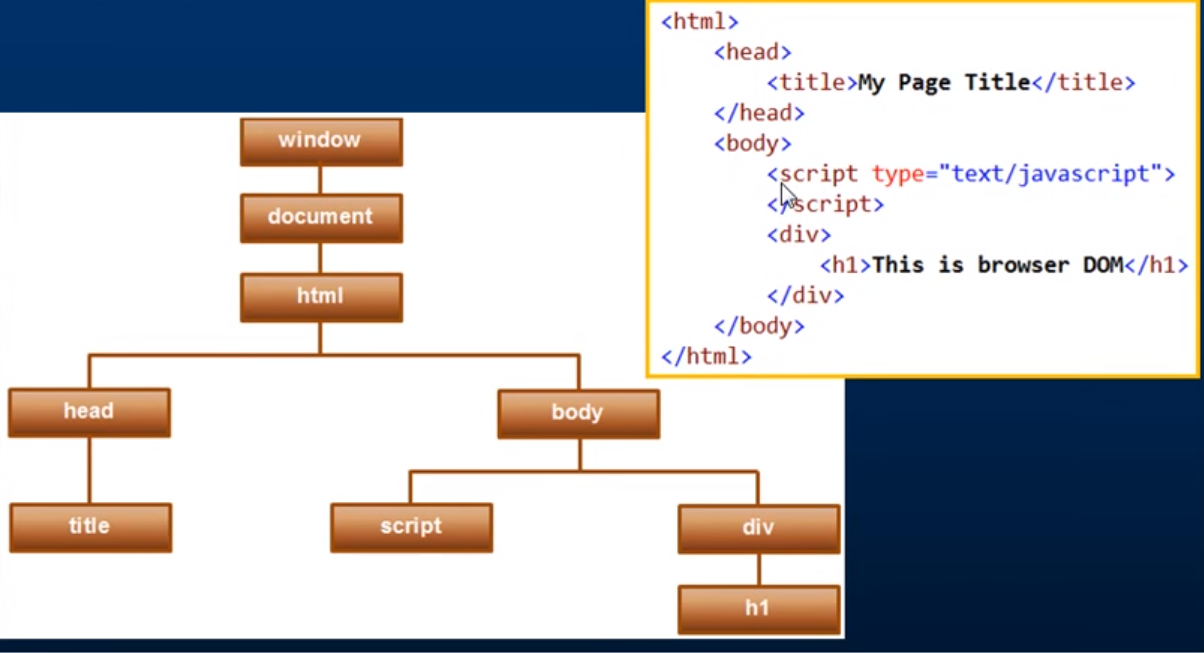
400-499 Client Error

500-599 ServerError.

Popular Status Codes

|  |  |  |
| --- | --- | --- |
| Code | Reason | Description |
| 200 | OK | Success |
| 301 | Moved Permanently | Resource moved don’t check again |
| 302 | Moved Temporarily | Resource moved check here again |
| 304 | Not modified | Resource has not changed since last retrieval |
| 400 | Bad Request | Bad Syntax |
| 403 | Forbidden | Refusal access |
| 404 | Not found | Resource do not exist |
| 500 | Internal Server Error | Something went wrong during processing |
| 503 | Service Unavailable | Server will not service the request, can not handle any more request |

DOM – Document Object Model



Javascript == vs ===

=== no type conversion is done

== type conversion is done.

4==”4” will return true

4===”4” will return false

**Nullables in C#:**

C# provides a special data types, the **Nullable** types to which you can assign normal range ofvalues as well as null values.

int? num1 = null;

int? num2 = 45;

Console.Writeline(“Values are {0}, {1}”,num1, num2);

Out put is : ,45.

**?? in C#(The Null Coalescing Operator)**

It is used with the nullable value types and reference types.

int? num1 = null;

int? num2 = 45;

int num3;

num3 =num1?? 5;

console.writeline(num3);

num3=num2?? 5

Output would be :

**Default Value Types in C#:**

bool false

int 0

byte 0

**Difference between Build and Rebuild:**

Build only builds the changes you made

Rebuild : Cleans all the projects and builds all the projects in the solution. That is why it takes longer.

**Why use context ?**

Context will allow to hit the database only when you need to access the data. This will minimize the memory used by the data. Rather getting all the data and storing in a memory. It will allow you to get data only when you need it

**How to clear DNS Cache? It is also called DNS flush.**

**Ipconfig /flushdns**

This task does not require you to have a administrative credentials. Therefore it is a best security practice. This allows you to reset the contents of the DNS client solver cache.

If you are running into a lot of “Page not found ” errors, try flushing your DNS cache to have your computer request new information.

**What is DNS ?**

Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.

When you type in a web address, e.g., [www.jimsbikes.com](http://www.jimsbikes.com/), your Internet Service Provider views the DNS associated with the domain name, translates it into a machine friendly IP address (for example 216.168.224.70 is the IP for jimsbikes.com) and directs your Internet connection to the correct website.

**How to rename a method name ?**

Right click the method and refactor. Change the name and click next, resharper will go find all the method and rename the method name.

**What do you do when you need to call a method that is in a separate class twice from two different methods ?**

First try to avoid instantiating the class twice. You can use the property. It will instantiate only when a method calls it. For eg. There is a method Add in SessionService class. You can call by

SessionService.add();

You put the following code on the top of the class body. Any method that needs to call a method of this class will use the same object. If an object is already created, it just returns the existing one, otherwise it will create one for you and return one.

private SessionService sessionService;

public SessionService SessionService

{

get

{

if (this.sessionService == null)

{

this.sessionService = new SessionService();

}

return this.sessionService;

}

}

What do you do when HttpBrowserCapabilitiesBase is not available in a class where the method is but it is available in a calling class ?

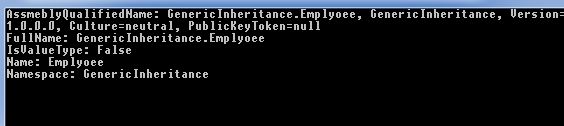
You can pass HttpBrowserCapabilitiesBase type as a parameter and you would be able to access the property that way.

public void LogBrowserInfo(string userLogin, HttpBrowserCapabilitiesBase browser)

**Difference Between typeof() and GetType()**

**typeof** keyword takes the Type itself as an argument and returns the underline Type of the argument whereas GetType() can only be invoked on the instance of the type.  
  
**typeof**  
*System.Type t1= typeof(Employee); // Employee is a Type*  
  
**GetType()**  
*Employee employee= new Employee();  
System.Type t2= employee.GetType(); // employee is an instance of the type Employee.*  
  
Both of examples create an instance of a type class for the Employee type. After you have a reference to the type, you can extract extra information from them as shown below.

1. Console.WriteLine("AssmeblyQualifiedName: {0}", t1.AssemblyQualifiedName);
2. Console.WriteLine("FullName: {0}", t1.FullName);
3. Console.WriteLine("IsValueType: {0}", t1.IsValueType);
4. Console.WriteLine("Name: {0}", t1.Name);
5. Console.WriteLine("Namespace: {0}", t1.Namespace);

**Output:**  
  
  
  
If you want to execute a method call on an instance, use reflection to load assembly first and retrieve methods info and execute method invocation.

**Fixture message process:**

After a fixture is scanned, scanner api sends the fixture message to web api and the message is saved to mongo db and in may be sql as well. If the message is ok it sends back a message to scanner api saying it is ok to delete.

Notes on RabbitMq:

Create a Queue and Exchange. You can create by using C#.

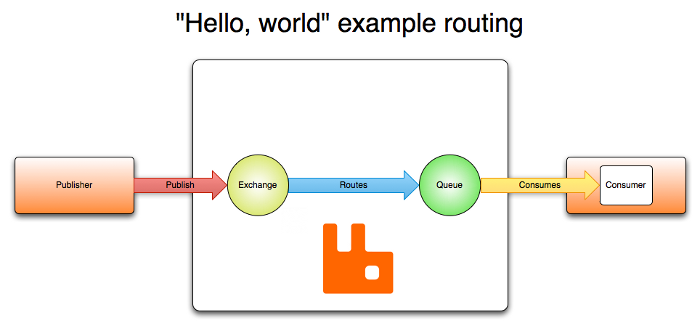
**Brokers and their roles:**

Messaging brokers receive message from publishers and route them to consumers. Since it is a network protocol, the publishers, consumers and the broker can all reside on different machines.

**High Level Overview of RMQ**

Messages are published to the Exchanges, Exchanges distribute message copies to the queue using rules called bindings. Then Amqp brokers either deliver the message or the consumers subscribed queue or consumer fetch/pull message from queues on demand.

Networks are unreliable and applications may fail to process messages therefore the AMQP model has a notion of message acknowledgements: when a message is delivered to a consumer the consumer notifies to broker. When message acknowledgements are in use, a broker will only completely remove message from a queue when it receives a notification for that message.



**Types of Exchanges:**

1. Direct Exchange: delivers messages to queues based on the message routing key K.
2. Fanout Exchange: It routes messages to all the queues that are bound to it and routing key is ignored.
3. Topic Exchange: It routes messages to one or many queues based on matching between a message routing key and the pattern that was used to bind a queue to an exchange.
4. Headers Exchange: A headers exchange is designed for routing on multiple attributes that are more easily express as a message.

**How to bind a queue to the Exchange?**

Open the queue and go to the binding . Enter the name of the exchange and routing key. If the exchange is the direct exchange, enter the routing key and you can leave the arguments empty.

**How to publish a message?**

Go to the Exchange and click the exchange that you want to publish the message from and go to the publish section at the bottom. Enter a message in the Payload and routing key if it is a Direct Exchange.

Click Publish. It is going to send message to all the queues that are bind to this exchange. For example if you have two queue are binded to an Exchange, you are going to see same message in 3 different queues. If you delete from queue1 it is not going to delete the message from other two. Even though it is exactly the same message.

**How to read a Message?**

Click the queue and click the queue that you would like to read the message from, Go to the get messages section and click Get Message. **Very Imp Note: if you read a message this way, it is going to flag it as a read message and it is going to send an acknowledgement to the producer. If requeue is not selected the message is gone forever. Even it is set to requeue, it will set as re-delivered.** This is why it is known as a destructive action.

**How to create a queue during the run time ?**

Check the project RabbitMQDemo in documents folder to see how to use it . You need to use the Nuget package manager called RabbitMQ.Client.

Steps:

1. Initialize the RabbitMQ.Client factory.
2. Create a connection : connectionfactory.CreateConnection();
3. Create a Model : connection.CreateModel();
4. Declare a queue : model.QueueDeclare("PadamQueue", true, false, false, null);
5. Declare an Exchange and types of Exchange: model.ExchangeDeclare("PadamExchange",ExchangeType.Direct);
6. Third you bind the queue with the Exchange with routing key. model.QueueBind("PadamQueue","PadamExchange","ct4")// ct4 is the routing key

Done

App.config vs Web.config:

These two files are basically the same. When you create an console application , app.config is created automatically and when you create a web application or a libraray web.config is created. App.config is based on xml format. When you build an application, exe is created in the bin folder and also a projectname.exe.config is created . In production, you can access the exe, you will only open the .exe.config to change the parameters. You can read the parameters from app.config or from the web.config. It is stored under the app setting. They are stored as a key value pair. Machine.config is mother of both app.config and web.config files and any other config file and for all the .net applications running in that machine. . It is located in : C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config. Do not change the values though. If you want to make global variable, you can change it here. Avoid using this file though.

App.config and web.config inherit the value from the machine.config

Unhandled exception at line 4900, column 5 in http://localhost:54929/Scripts/DataTables/media/js/jquery.dataTables.js

0x800a138f - JavaScript runtime error: Unable to get property '0' of undefined or null reference

Unhandled exception at line 3108, column 3 in http://localhost:54929/Scripts/DataTables/media/js/jquery.dataTables.js

0x800a138f - JavaScript runtime error: Unable to get property 'sSearch' of undefined or null reference