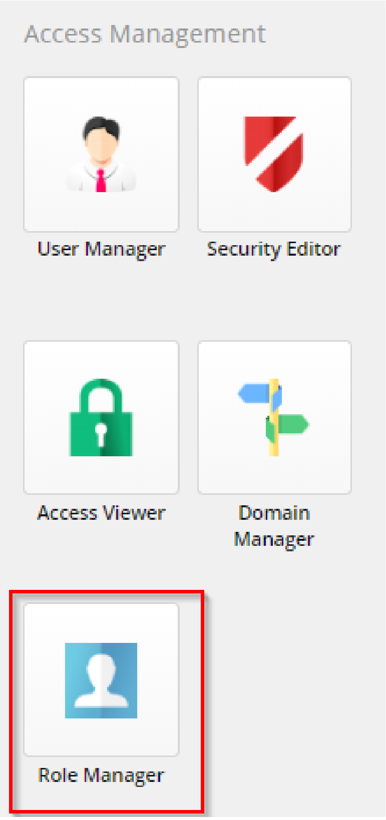
Sitecore 10 Training – Day 3

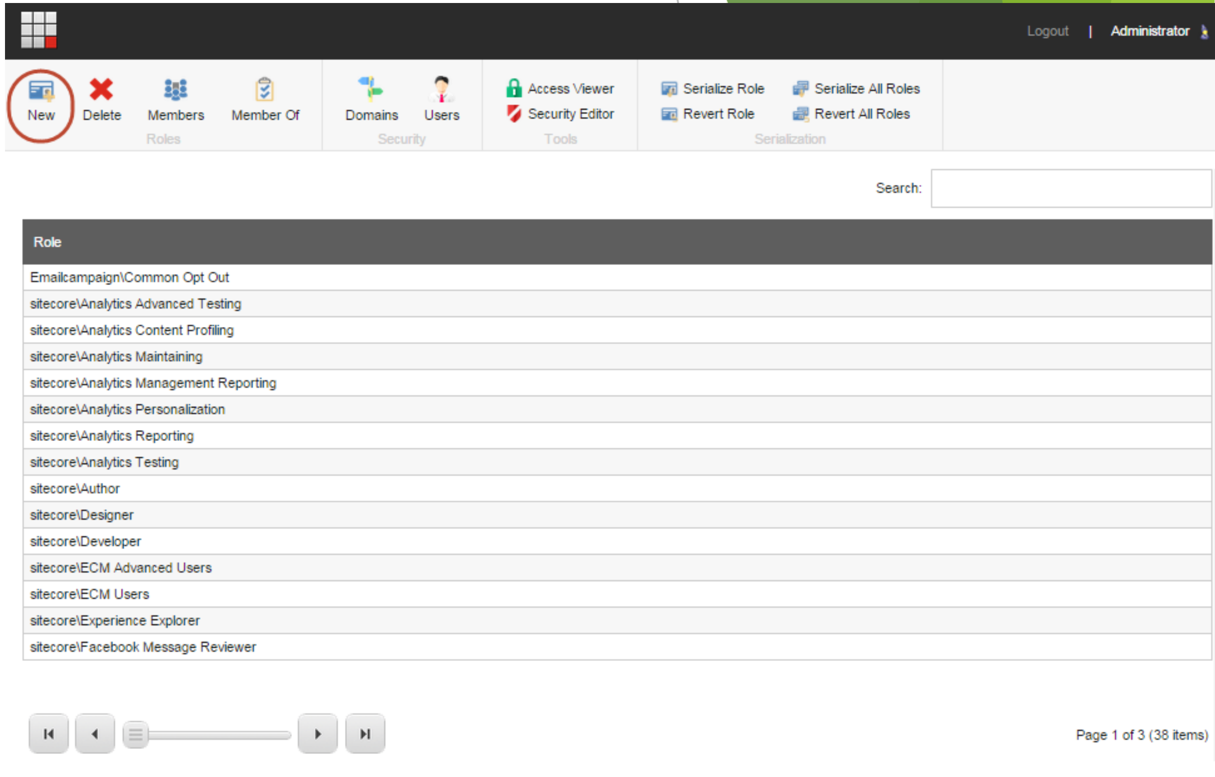
By Surendra Sharma

MODULE 5 – Securities and Customizations

Roles

- You can use roles to manage website authorization.
- Roles allow the grouping of users into structured units, such as managers, sales staff, anonymous users, and so on. This makes it easier to organize security access because you can use a single role to assign security access rights to multiple users.
- Roles give you the flexibility to change permissions and to add or remove users without having to make changes to the whole website.
- Create a role
  - Log in to Sitecore, and on the Launchpad, click the **Role Manager** icon.
  - In the **Role Manager** window, on the ribbon, click **New**.
  - In the **NewRole dialog box**, enter the name and domain of the new role.



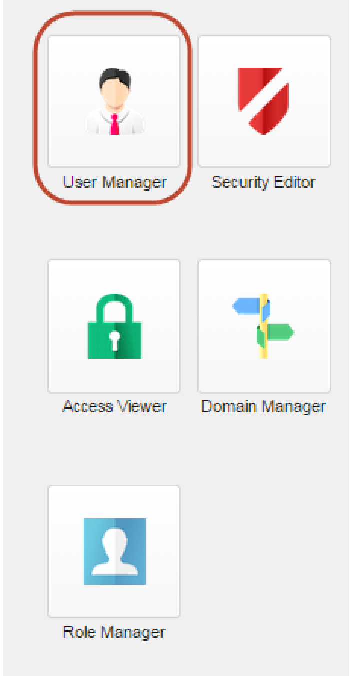


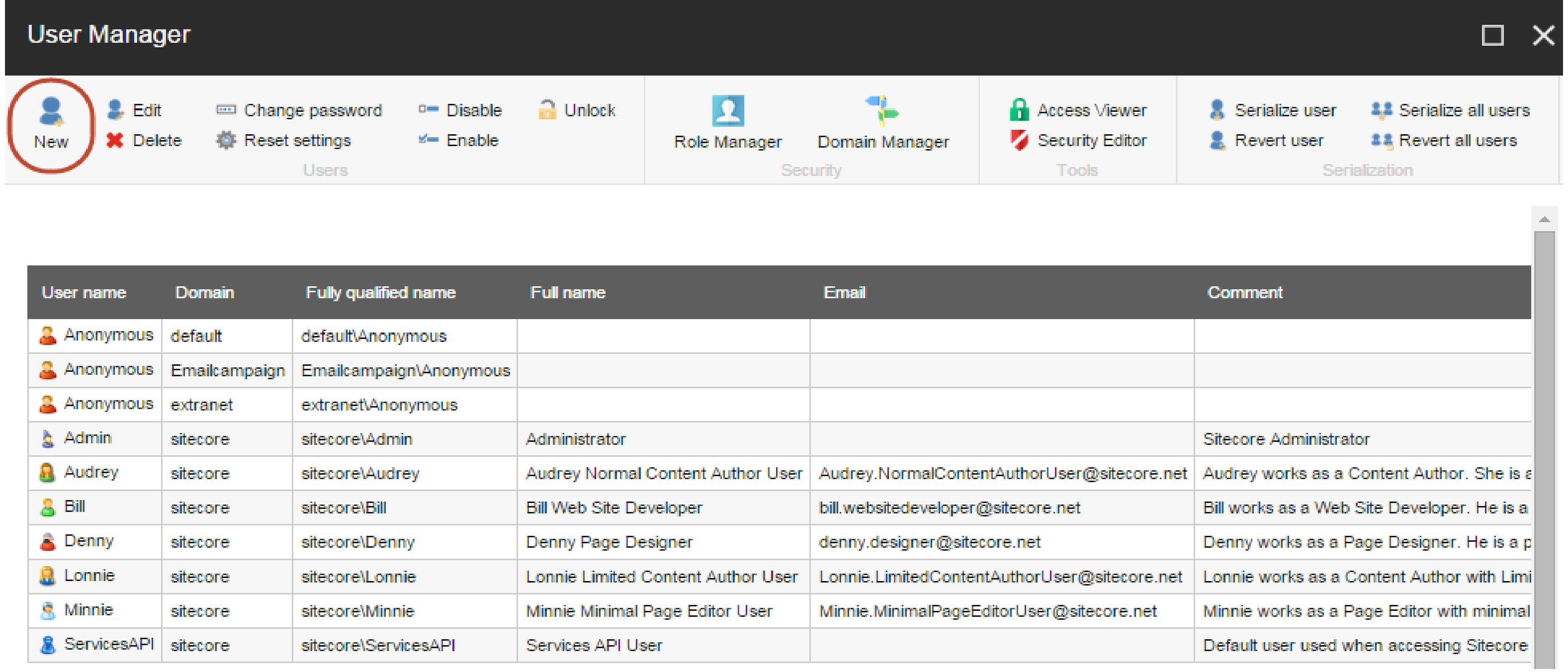
Roles required for creating editor and reviewer role in Sitecore

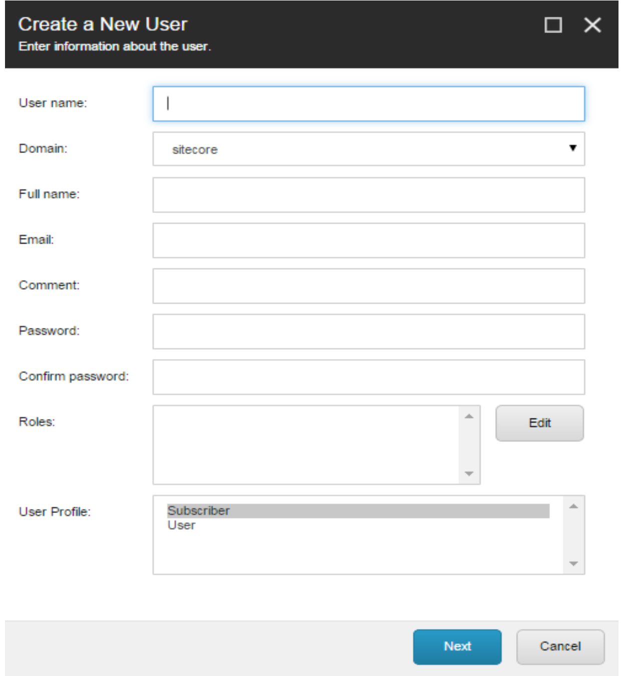
Give *Read* permission to "*/sitecore/client/Applications/Launchpad/PageSettings/Buttons*" in CORE database to editor group for dashboard icons access

- For creating Editor Group you need below roles
  - sitecore\Author

- For creating Reviewer Group, you need all editor roles + Sitecore Client Publishing role as below
  - sitecore\Author [or Editor Group]
  - sitecore\Sitecore Client Publishing





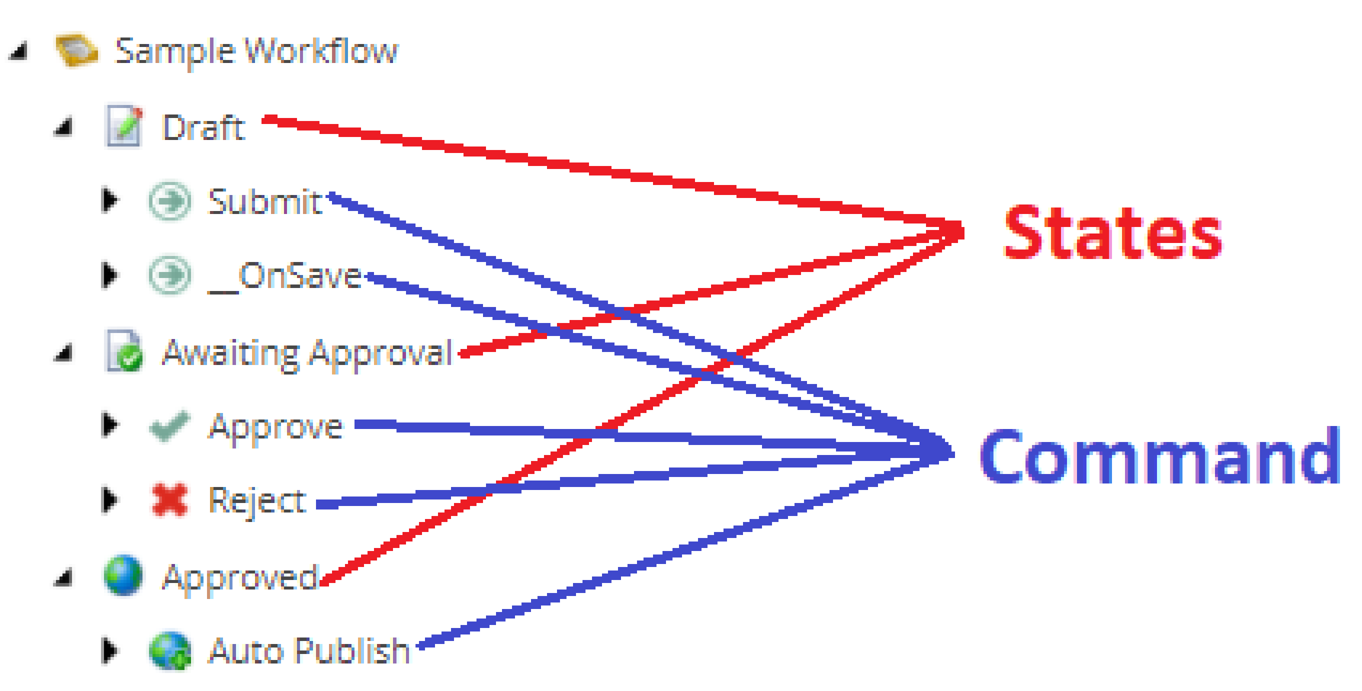


Create a user

- Log in to Sitecore and, on the Launchpad, click **User Manager**.
- In the **User Manager** dialog box, in the **Users** group, click **New**.
- In the **Create a New User** dialog box, enter the relevant information about the new user.

Workflow

- According to Sitecore "Workflows ensure that items move through a predefined set of states before they become publishable, usually intended to ensure that content receives the appropriate reviews and approvals before publication to the live Web site."
- All workflow are resides in "/sitecore/system/Workflows/" folder.
- A workflow have states, commands and actions.
- Every workflow start with draft state and end with publish or final state.
- Most of the workflow looks indicated in below diagram
- Assign your workflow to "Default workflow" field of Standard values.



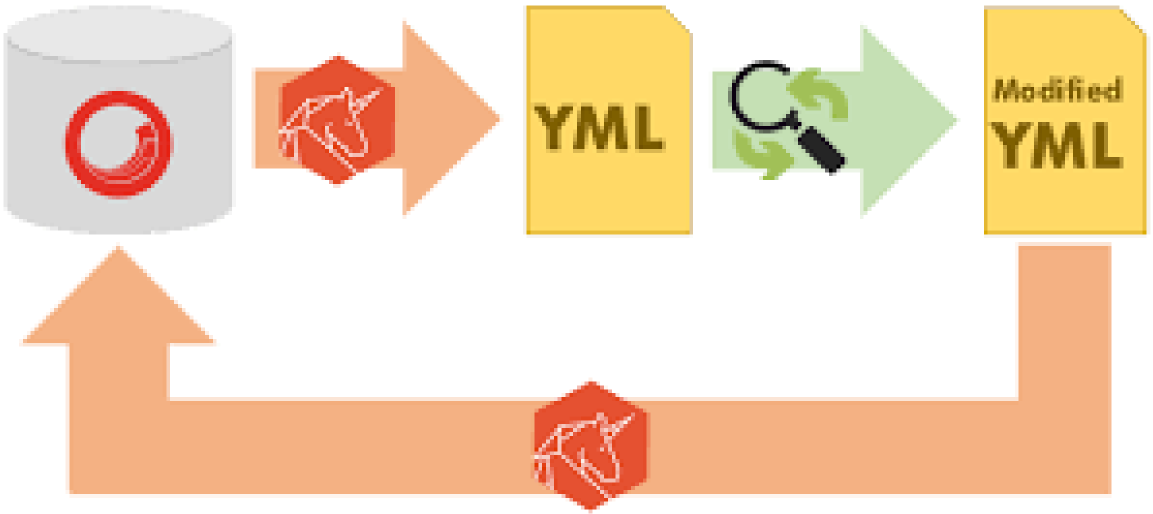
Data Template vs. Standard Values

- Data Template
  1. Field Sections
  2. Fields
  3. Template Icon
  4. Base template Inheritance

- Standard Values
  1. Default Field Values
  2. Default Insert Options
  3. Default Presentation Details*
  4. Default Workflows*

Serialization

- Serialization allows you to serialize an entire Sitecore database or a series of items in a database to text files. You can then use these text files to transfer this database or series of items to another database or Sitecore solution.
- The Sitecore serialization functionality is designed to help teams of developers that work on the same Sitecore solution to synchronize database changes between their individual development environments, but is also valuable when a single developer works on a solution.
- Sitecore serialization is typically used in combination with a source control system to make it easy for developers to synchronize database changes between their local databases that they use for development — with the added benefit that the source control system will keep track of database changes and allow you to compare database changes.



Serialization – Ways to do (Unicorn)

Check document

Buckets

- What is an Item Bucket?
  - A bucket is a special type of item in Sitecore that acts as a container for other items.
  - Most importantly, the underlying structure of items inside a bucket does not matter to the user. Items in a bucket are not displayed in the content tree - instead, users find these items by using Sitecore search.
- When to Use a Bucket
  - Tree structure isn't important.
  - There are more than 100 items under one node.
  - Searching for content is preferred.

- Bucketing done on the basis of YYYY\MM\dd\HH\mm format by default.
  - Change Bucketing structure from "\App_Config\Sitecore\Buckets\Sitecore.Buckets.config" file or make your patch file having entry

```
<setting name="BucketConfiguration.BucketFolderPath" value="yyyy\/MM\/dd\/HH\/mm"/>
```

Demo

Buckets

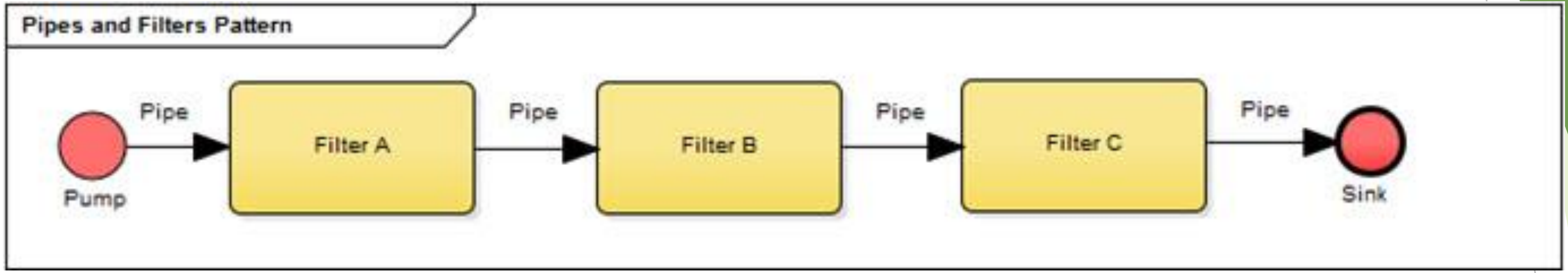
Check document

Demo

Implementing Search-Driven Components

Check document and Sample Application

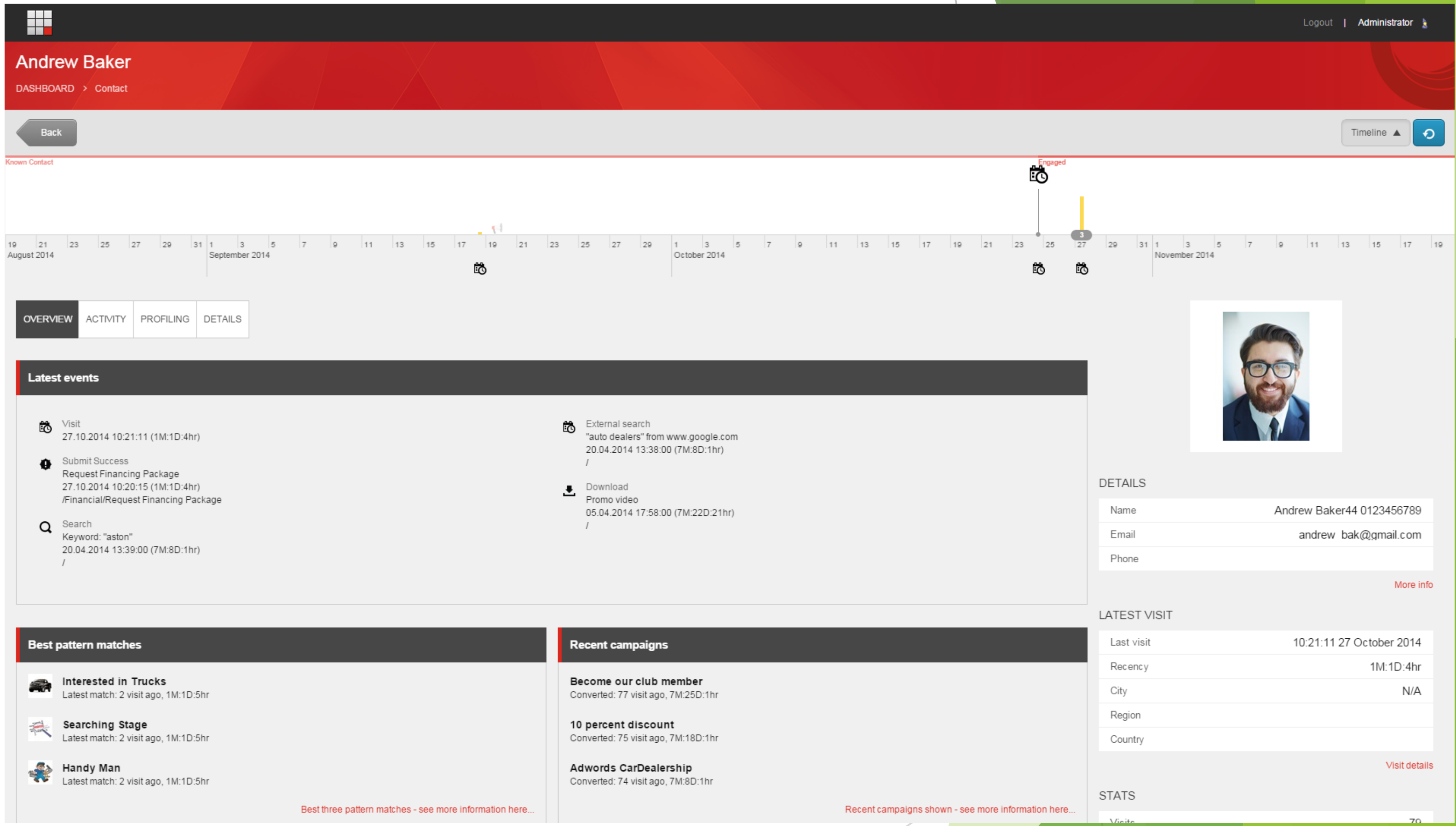
Pipeline and Processor



- Pipelines define a sequence of processors that implement a function, such as defining the Sitecore context for an HTTP request or generating a list of messages in the Content Editor.
- Each processor in a pipeline contains a method named Process() that accepts a single argument and returns void. A processor can abort the pipeline, preventing Sitecore from invoking subsequent processors.
- The argument that is passed to the Process() method must be of a type that is specific to the pipeline or be the default argument – Sitecore.Pipelines.PipelineArgs. To create a pipeline processor, create a class that implements a method named Process() with the same signature as the other processors in the pipeline. This processor can inherit from an existing processor, and you can add, remove, replace, and rearrange processors in the pipelines to suit your requirements.

Experience Profile

- The Experience Profile lets you monitor the behavior of contacts that have interacted with your company or with your website.
- The Experience Profile enables you and your sales teams to monitor the key areas of customer experience and interaction, such as visits, campaigns, goals, profiles, automations, outcomes, and keywords. For example, for each contact, you can see at a glance which events and goals they have triggered as well as how many engagement value points they have accumulated on your website.
- The Home page of the Experience Profile is a dashboard that lets you find specific contacts. When your website has been running for some time, you may have to search through many contacts.



Sitecore xConnect

- xConnect is the service layer that sits in between the xDB Collection database and xDB index, and any trusted client or device that wants to read, write, or search contact and interaction data.
- All clients must use xConnect to read, write, and search xDB data. Access to xConnect is secure by default and all clients must supply a valid certificate thumbprint. You can use the xConnect Client API (C#) or interact with the web API directly.
- Create contact through xConnect console application
- Check the visitor record in Sitecore Experience Profile
- Also check contact details in SQL server table `[xpten_Xdb.Collection.Shard0].[xdb_collection].[ContactFacets]`
- To enable xConnect Anonymous data collection, make below entry "***true***" in "*\YourxConnectApp\App_Data\Config\Sitecore\SearchIndexer\sc.Xdb.Collection.IndexerSettings.xml*" and "*YourxConnectApp \App_Data\jobs\continuous\IndexWorker\App_Data\config\sitecore\SearchIndexer\sc.Xdb.Collection.IndexerSettings.xml*"
  - `<IndexPIISensitiveData>true</IndexPIISensitiveData>`
  - `<IndexAnonymousContactData>true</IndexAnonymousContactData>`
- xConnect Issue : The request was aborted: Could not create SSL/TLS secure channel.
  - Open "***Manage computer certificates***" in a start menu search
  - Find the client certificate (normally in Personal\Certificates) with name "<sitecore>.xconnect". Right click it, choose ***All Tasks***, then ***Manage Private Keys***.
  - Assign "***Everyone***" permission to this.

Check document and Sample Application for more details