Introduction to DordStream

Dordstream is a free and open source content management system software written in C# and paired with MSSQL server.

DordSream is a content management system software that:

(1) Allow to perform database operation through Javascript (Introduced SQL Query).

(2) Allow to develop blog and website with Html Document (Introduced Dshtml Tag).

(3) Allow to develop web software.

DordStream is a Template based CMS software I.e. The website or blog or web software has to be developed in template design structure and installed for usage.

DordStream has the following features

1. It allows multi user website or web application or blog.
2. It is template-based.
3. It allows adding and removing of widget.
4. Its supported one website or web application per installation.
5. Its support other web content types including mailing, media galleries and membership site.
6. It runs on any computer platform that supports both web server capable of running .Net Core and a database (MSSQL Server) to the content and configuration (Internet Information Services IIS).
7. It allows to watermark uploaded images.
8. It allows direct publication to facebook page.
9. It allows setting of role in database operation.

10. It allows setting of role of accessing any page name contain either dord or custom (Template pages).
11. It allows creation of posts.
12. It allows creation of pages.
13. It allows changing of Title (site title).
14. It allows changing of the favicon (site logo on browser).
15. It allows changing of Keywords.
16. It allows changing of Description.
17. It allows to enable anonymous user.
18. It allows uploading of template.
19. It allows to enable usage of cache.
20. It allows to enable ip address verification.

Introduction to DordStream Html (Dshtml)

What is Html Tag?

Html tags are like keywords that defines how web browser will format and display the content. With the help of the tags a web browser can distinguish between a html content and a simple content. For more details about Html, visit https://www.w3school.com.

The Html-like tag was named Dshtml Tag.

What is Dshtml Tag?

Dshtml tag is a set of custom tag that extend the function of html through the CMS.

Dshtml is also html-like that is used to perform some tasks within the CMS and web browser will display and format the same as html content.

Similarities between html tag and dshtml tag are

1. All html tags and dshtml tags must be enclosed within angle bracket (< >) while dshtml tags.
2. Every tag in html and dshtml perform different tasks.
3. Html tags are, <p>, <h1>, <strong>, <li>, <a> …etc. for more details visit https://www.w3school.com. While dshtml tags are, <audiolist>, <videolist>, <blogbody>, <featureposts>, <recentposts>, <posttypes>, <multiplecomments>, <multiplelabels>, <multiplepages>, <pagination>, <logout>, <singlepage>, <singlepost>, <sitename>, <widget>, <layout>, <top>, <left>, <right>, <bottom>.
4. Html Tag can be use with html attributes and events while dshtml can be use with dshtml attribute only.
5. Html attributes are id, class, title, src, name, alt, …etc. for more details visit https://www.w3school.com. While Dshtml attributes are ds-artist, ds-album, ds-count, ds-widget, ds-index, ds-name, ds-description, ds-title-length, ds-content-length, ds-label-exclude, ds-text, ds-type, ds-title, ds-src, ds-link, ds-releasedate, ds-date, ds-size, ds-firstletter, ds-time, ds-username, ds-link-type, ds-content, ds-image, ds-video, ds-sociallink, ds-author, ds-commentcount, ds-error, ds-email, ds-subject, ds-message, ds-password, ds-pageindex, ds-phonenumber, ds-firstname, ds-lastname.

DordStream Html Attributes (dshtml attributes)

Dshtml attributes are special words used inside the opening tag to control the element behaviors. Dshtml attributes are modifier of a Dshtml element type. An attribute either modifies the default functionality to certain element types unable to function correctly without them. In Dshtml syntax, an attribute is added to a Dshtml start tag.

Several basic attributes types have been recognized, including (1) required attributes, needed by a particular element type for that element type to function correctly. (2) optional attributes, used to modify the default functionality of an element type. (3) standard attributes, supported by many element types.

Dshtml attributes generally appear as name-value pairs, separated by = and are written within the start tag of an element, after the element name;

<element attribute=”value”> element</element>

Where element name is the Dshtml element type, and attribute is the name of the attribute, set to the provided value. The value may be enclosed in double quotes.

Required Attributes

These are the attributes that is required when using dshtml element.

1. Name (ds-name): This attribute provides a unique identifier for an element i.e. unique name.
2. Description (ds-description): This attribute provides full description for an element.
3. Artist (ds-artist): This attribute provides the artist name for an element. It can only be used with audiolist and videolist dshtml element.
4. Album (ds-album): This attribute provides the album name for an element. It can only be used with audiolist and videolist dshtml element.

5. Type (ds-type): This attribute provides the label name for an element. It can only be used with posttypes element.

Optional Attributes

These are the attributes that is optional when using dshtml element.

1. Count (ds-count): This attribute provides the number of children for an element.
2. Widget (ds-widget): This attribute identifies an element as widget on the CMS. It takes widget as value.
3. Exclude (ds-label-exclude): This attribute provides the label name(s) to be excluded from the CMS to an element.
4. Index (ds-index): This attribute provides the position (index) for accessing an element.
5. Title Length (ds-title-length): This attribute provides the title length of the children element that make use of either title as parameter or ds-title as attribute. It can be used with dshtml element and div, ul html element.
6. Content Length (ds-content-length): This attribute provides the content length of the children element that make use of either content as parameter or ds-content as attribute. It can be used with dshtml element and div, ul html element.
7. Parent (ds-parent): This attribute provides the dshtml element name. It can only be when div or ul is the parent element and it cannot be use with dshtml element. Its values are label, page, comment, pagination, body, recent, feature and label name.
8. Text (ds-text): This attribute provides the text for auth element. e.g. welcome or greetings text.

Standard Attributes

These are the attributes used with sibling’s element, when div or ul is use as the parent tag. It takes 1 as value. These attributes can also be used as parameter when using the attribute as value to id html attribute. It can be use as parameter when removing the prefix (ds-). These attributes have the same function as the parameter and different usage. The similarity between this attribute and the parameter is that , The attribute take 1 as value and it is only use as children attribute when div or ul is used as parent tag while The parameter is use as value to id attribute and it is use as children attribute when dshtml tag is used as parent element.

The attribute that is also used as the parameter change or replace the content or link of the assign element.

Example of standard attributes

```
<div ds-parent=”label”>

<img ds-image=”1” alt=”Image”/>

</div>
```

Example of the parameter

```
<multiplelabels>

<img id=”image” alt=”Image” />

</multiplelabels>
```

Example 1 make use of div tag as parent element (html tag) and the attribute ds-image as sibling or children element attribute which take 1 as value.

Example 2 make use of multiplelabels tag (dshtml tag) as parent tag and use id attribute with image as value.

The two examples have different syntax but produce the same result.

These are the attributes that is also used as parameter (removing the prefix ds-).

1. Name (ds-name): This is the label name. i.e. the label name creating when publishing post on the CMS.
2. Title (ds-title): This is the title of the post. i.e. the title of the post published on the CMS.
3. Content (ds-content): This is content or body of the post. i.e. the body post published on the CMS.
4. Image (ds-image): This is the first image contain in the body of the post. It can only be use with img element.
5. Video (ds-video): This is the first video contain in the body of the post. It can only be use with video element.
6. Audio (ds-audio): This is the first image contain in the body of the post. It can only be use with audio element.
7. Link (ds-link): This is the link to the post. i.e. the link to access the particular post. It can only be use with a element. In a scenario where multiple links are needed ds-link1, ds-link2, ds-link3, ds-link4 can be used.
8. Author (ds-author): This is the author that make publication. i.e. author that create the post on the CMS.
9. Date (ds-date): This is the publication date. i.e. the date the post was published or created.
10. Time (ds-time): This is the publication time. i.e. the time the post was published or created.
11. CommentCount (ds-commentcount): This is the total number of users commented on a post. i.e. the number of users that make a comment on the particular post.
12. Id (ds-id): This is the post unique Identifier. i.e. the postId.
13. FirstLetter (ds-firstletter): This is first letter of the post title.
14. SocialLink (ds-sociallink): This is used when social media is required, the link to the post will be append to the provided social link. In a scenario where multiple social links are needed ds-sociallink1, ds-sociallink2, ds-sociallink3, ds-sociallink4 can be used.
15. #Name or ##Name: this cannot be use as attribute or parameter. It can only be use as value when label name is needed for accessing through javascript. It can only be use as attribute value. e.g. <a data-target="#name">Click me</a> #name will change to the label name when access through browser.
16. Src (ds-src): This is link to the audio file which can be used with source html element.

DordStream Html Tag (dshtml tag)

Dshtml tag is a set of custom tag that extend the function of html through the CMS. It is also html-like that is used to perform some tasks within the CMS and displayed, formatted the same as html content on web browser. It has the same syntax as html

Dshtml element can only be written as a child tag to html. i.e. it must be written inside html element.

Dshtml attributes can be written with dshtml element and html element.

Dshtml parameters can only be written inside dshtml element.

NOTE, Dshtml tag depend on dshtml attribute or dshtml parameter to perform the task.

These are the dshtml tags

1. **DordBody(dordbody)**

DordBody is a dshtml tag which is used for rendering of body.

It is case sensitive(lower case).

It has no attributes and parameters.

It can be writing as <dordbody/>.

2. AudioList (audiolist)

Audio List is a dshtml tag which is used with audio html tag in order to get or display the audio files from the CMS. Audio files can be retrieved with the dshtml parameters.

It is also case sensitive (lower case).

It can be written as <audiolist> children element </audiolist>

Attributes: Artist(ds-artist), Album(ds-album), Count(ds-count), Widget(widget), Index(ds-index), Name(ds-name), Description(ds-description).

Parameters: Title, Name, Src, Link, Artist, Album, ReleaseDate, Date, Size, FirstLetter, Time

Code example

The snippet code retrieved 30 davido’s audio files from the album named One mill and display on web browser.

```
<audiolist ds-artist="davido" ds-album="One Mill" ds-count="30" ds-name="Video List" ds-description="Audio List">

<audio controls id="video">

<source src="#" id="src">

</audio>

</audiolist>
```

3. *VideoList Tag (videolist)*

Video List is a dshtml tag which is used with a video html tag in order to get or display the video files from the CMS.

Video files can be retrieved with the dshtml parameters.

It is also case sensitive (lower case).

It can be written as <videolist> children element </videolist>

Attributes: Artist(ds-artist), Album(ds-album), Count(ds-count), Widget(widget), Index(ds-index), Name(ds-name), Description(ds-description).

Parameters: Title, Name, Src, Link, Artist, Album, ReleaseDate, Date, Size, FirstLetter, Time

Code example

The snippet code retrieved 30 davido’s video files from the album named One mill and display on web browser.

```
<videolist ds-artist="davido" ds-album="One Mill" ds-count="30" ds-name="Video List" ds-description="Video List">

<video controls id="video">

<source src="#" id="src">

</video>

</videolist>
```

4. *Authorization Tag(auth)*

Authorization Tag is a dshtml tag which is used for Authentication and Authorization of users. It is used when the registration and login are required.

It is also case sensitive (lower case).

It can be written as <auth> children element </auth>

Attributes: Text(ds-text), Name(ds-name), Description(ds-description).

Parameters: Username, Login, Register

Code example

The snippet code display “welcome!” + username of the authenticated user, and register and login link if it was not authenticated on web browser.

```
 <auth ds-text="Welcome!">

  <p id="username" style="text-align:center; color: blue; text-decoration:underline; margin-left: 30px; margin-top: 15px;">Bellie_Coin</p>
```

```
    <li><a href="#" data-hover="Registration" id="register">Registration</a></li>

     <li><a href="#" data-hover="Login" id="login">Login</a></li>

      </auth>
```

5. *Blog Body Tag (blogbody)*

Blog Body Tag is a dshtml tag that retrieves or displays the categorize contents based on the label name from the post section of the CMS. It can only be written inside category page of the template.

It is also case sensitive (lower case).

It can be written using two methods (1) <blogbody> children element </blogbody> (2) either <div> or <ul> tag. (ds-parent must be set to body).

Attributes: Name(ds-name), Description(ds-description), Widget, Index(ds-index), Title Length(ds-title-length), Content Length(ds-title-length), Count(ds-count)

Parameters: Title, Name, Content, Image, Video, Audio, Link, Link1, Link2, Link3, Link4, Author, Date, CommentCount, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code Example

The snippet code below displays 30 posts based on the specified label name through the url with title length of 30 and content length of 40 on web browser. Using the two methods.

First Method

```
<blogbody ds-name="Post Body" ds-description="Display Post Body" ds-count="30" ds-title-length="30" ds-content-length="40" >

                <div class="card">
```

```
<p class="title" id="title>Post Title </p>

  <p class="name" id="name"> Label Name</p>

 <img src="#" id="image" class="card-img" />

  <i id="time"> Last seen</i>

  <i id="commentcount">Total Comments<i/>

   <span id="author">Author Name </span>

    <audio controls><source src="#" id="audio" type="mp3" ></audio>

     <video controls><source src="#" id="video" type="mp3" ></video>

    <a href="#" id="link">Post Link </a>

    <a href="#" id="link1">Post Link1 </a>

     <a href="#" id="link2">Post Link2 </a>

      <a href="#" id="link3">Post Link3 </a>

       <a href="#" id="link4">Post Link4 </a>

      <a href="www.facebook.com?" id="sociallink">Post Social Link </a>

        <a href="www.facebook.com?" id="sociallink1">Post Social Link1 </a>

         <a href="www.facebook.com?" id="sociallink2">Post Social Link2 </a>

        <a href="www.facebook.com?" id="sociallink3">Post Social Link3 </a>

      <a href="www.facebook.com?" id="sociallink4">Post Social Link4 </a>

       <p id="firstletter"> Post First Letter </p>

      <i id="date">Post Date </i>

      <i id="id"> Post Id</i>

      <input type="hidden" id="id"/>

       <a data-target="#name" href="#">Label Name </a>

       <a data-target="##name" href="#">Label Name </a>
```

```
                </div>
              </blogbody>
```

Second Method

```
<div ds-parent=”body” ds-name="Post Body" ds-description="Display Post Body" ds-count="30" ds-title-
length="30" ds-content-length="40">
                <div class="card">
                <p class="title" ds-title=”1”>Post Title </p>
<p class="name" ds-name=”1"> Label Name</p>
                 <img src="#" ds-image=”1" class="card-img" />
                  <i ds-time=”1"> Last seen</i>
                  <i ds-commentcount=”1">Total Comments<i/>
                   <span ds-author=”1">Author Name </span>
                    <audio controls><source src="#" ds-audio=”1" type="mp3" ></audio>
                     <video controls><source src="#" ds-video=”1" type="mp3" ></video>
                    <a href="#" ds-link=”1">Post Link </a>
                    <a href="#" ds-link1=”1">Post Link1 </a>
                     <a href="#" ds-link2=”1">Post Link2 </a>
                      <a href="#" ds-link3=”1">Post Link3 </a>
                       <a href="#" ds-link4=”1">Post Link4 </a>
                      <a href="www.facebook.com?" ds-sociallink=”1">Post Social Link </a>
                        <a href="www.facebook.com?" ds-sociallink1=”1">Post Social Link1 </a>
                         <a href="www.facebook.com?" ds-sociallink2=”1">Post Social Link2 </a>
                        <a href="www.facebook.com?" ds-sociallink3=”1">Post Social Link3 </a>
                      <a href="www.facebook.com?" ds-sociallink4=”1">Post Social Link4 </a>
```

```
            <p ds-firstletter=”1"> Post First Letter </p>

           <i ds-date=”1">Post Date </i>

           <i ds-id=”1"> Post Id</i>

           <input type="hidden" ds-id=”1"/>

            <a data-target="#name" href="#">Label Name </a>

            <a data-target="##name" href="#">Label Name </a>

       </div>

     </div>
```

6. RecentPosts Tag (recentposts)

RecentPosts Tag is a dshtml tag that retrieves or displays posts that ordered by posted date from the post section of the CMS. i.e. recent posts

It is also case sensitive (lower case).

It can be written using two methods (1) <recentposts> children element </recentposts> (2) either <div> or <ul> tag. (ds-parent must be set to recent).

Attributes: Name(ds-name), Description(ds-description), Widget, Index(ds-index), Title Length(ds-title-length), Content Length(ds-title-length), Count(ds-count), Exclude (ds-label-exclude).

Parameters: Title, Name, Content, Image, Video, Audio, Link, Link1, Link2, Link3, Link4, Author, Date, CommentCount, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code Example

The snippet code below displays 30 recent posts, excluding news and beauty label name, with title length of 30 and content length of 40 on web browser. Using the two methods.

First Method

```
<recentposts ds-name="Recent Post" ds-description="Display Recent Post" ds-count="30" ds-title-length="30" ds-content-length="40"  ds-label-exclude="news,beauty" >

                <div class="card">

                <p class="title" id="title">Post Title </p>

                  <p class="name" id="name"> Label Name</p>

                 <img src="#" id="image" class="card-img" />

                  <i id="time"> Last seen</i>

                  <i id="commentcount">Total Comments<i/>

                   <span id="author">Author Name </span>

                    <audio controls><source src="#" id="audio" type="mp3" ></audio>

                     <video controls><source src="#" id="video" type="mp3" ></video>

                    <a href="#" id="link">Post Link </a>

                    <a href="#" id="link1">Post Link1 </a>

                     <a href="#" id="link2">Post Link2 </a>

                      <a href="#" id="link3">Post Link3 </a>

                       <a href="#" id="link4">Post Link4 </a>

                      <a href="www.facebook.com?" id="sociallink">Post Social Link </a>

                        <a href="www.facebook.com?" id="sociallink1">Post Social Link1 </a>

                         <a href="www.facebook.com?" id="sociallink2">Post Social Link2 </a>

                        <a href="www.facebook.com?" id="sociallink3">Post Social Link3 </a>

                      <a href="www.facebook.com?" id="sociallink4">Post Social Link4 </a>

                       <p id="firstletter"> Post First Letter </p>
                      <i id="date">Post Date </i>

                      <i id="id"> Post Id</i>

                      <input type="hidden" id="id"/>

                       <a data-target="#name" href="#">Label Name </a>

                       <a data-target="##name" href="#">Label Name </a>

                </div>

              </recentposts>
```

Second Method

```
<div  ds-parent=”recent”  ds-name="Recent  Post"  ds-description="Display  Recent  Post"  ds-count="30"  ds-title-
length="30" ds-content-length="40" ds-label-exclude="news,beauty" >

                <div class="card">

                <p class="title" ds-title=”1”>Post Title </p>

<p class="name" ds-name=”1"> Label Name</p>

                 <img src="#" ds-image=”1" class="card-img" />

                  <i ds-time=”1"> Last seen</i>

                  <i ds-commentcount=”1">Total Comments<i/>

                   <span ds-author=”1">Author Name </span>

                    <audio controls><source src="#" ds-audio=”1" type="mp3" ></audio>

                     <video controls><source src="#" ds-video=”1" type="mp3" ></video>

                    <a href="#" ds-link=”1">Post Link </a>

                    <a href="#" ds-link1=”1">Post Link1 </a>

                     <a href="#" ds-link2=”1">Post Link2 </a>

                      <a href="#" ds-link3=”1">Post Link3 </a>

                       <a href="#" ds-link4=”1">Post Link4 </a>
```

```
        <a href="www.facebook.com?" ds-sociallink=”1">Post Social Link </a>
         <a href="www.facebook.com?" ds-sociallink1=”1">Post Social Link1 </a>
          <a href="www.facebook.com?" ds-sociallink2=”1">Post Social Link2 </a>
         <a href="www.facebook.com?" ds-sociallink3=”1">Post Social Link3 </a>
        <a href="www.facebook.com?" ds-sociallink4=”1">Post Social Link4 </a>
         <p ds-firstletter=”1"> Post First Letter </p>
        <i ds-date=”1">Post Date </i>
        <i ds-id=”1"> Post Id</i>
        <input type="hidden" ds-id=”1"/>
         <a data-target="#name" href="#">Label Name </a>
         <a data-target="##name" href="#">Label Name </a>
    </div>
  </div>
```

7. *FeaturePosts Tag (featureposts)*

FeaturePosts Tag is a dshtml tag that retrieves or displays posts that ordered by label name from the post section of the CMS. i.e. feature posts

It is also case sensitive (lower case).

It can be written using two methods (1) <featureposts> children element </featureposts> (2) either <div> or <ul> tag. (ds-parent must be set to feature).

Attributes: Name(ds-name), Description(ds-description), Widget, Index(ds-index), Title Length(ds-title-length), Content Length(ds-title-length), Count(ds-count), Exclude (ds-label-exclude).

Parameters: Title, Name, Content, Image, Video, Audio, Link, Link1, Link2, Link3, Link4, Author, Date, CommentCount, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code Example

The snippet code below displays 30 feature posts, excluding news and beauty label name, with title length of 30 and content length of 40 on web browser. Using the two methods.

First Method

```
<featureposts ds-name="Feature Post" ds-description="Display Feature Post" ds-count="30" ds-title-length="30"
ds-content-length="40"  ds-label-exclude="news,beauty" >

                <div class="card">

                <p class="title" id="title>Post Title </p>

                  <p class="name" id="name"> Label Name</p>

                 <img src="#" id="image" class="card-img" />

                  <i id="time"> Last seen</i>

                  <i id="commentcount">Total Comments<i/>

                   <span id="author">Author Name </span>

                    <audio controls><source src="#" id="audio" type="mp3" ></audio>

                     <video controls><source src="#" id="video" type="mp3" ></video>

                    <a href="#" id="link">Post Link </a>

                    <a href="#" id="link1">Post Link1 </a>

                     <a href="#" id="link2">Post Link2 </a>

                      <a href="#" id="link3">Post Link3 </a>

                       <a href="#" id="link4">Post Link4 </a>

                      <a href="www.facebook.com?" id="sociallink">Post Social Link </a>

                        <a href="www.facebook.com?" id="sociallink1">Post Social Link1 </a>
                    <a href="www.facebook.com?" id="sociallink2">Post Social Link2 </a>
                   <a href="www.facebook.com?" id="sociallink3">Post Social Link3 </a>
                  <a href="www.facebook.com?" id="sociallink4">Post Social Link4 </a>
                  <p id="firstletter"> Post First Letter </p>
                  <i id="date">Post Date </i>
                  <i id="id"> Post Id</i>
                  <input type="hidden" id="id"/>
                  <a data-target="#name" href="#">Label Name </a>
                  <a data-target="##name" href="#">Label Name </a>
             </div>
          </featureposts>
```

Second Method

```
<div ds-parent="feature" ds-name="Feature Post" ds-description="Display Feature Post" ds-count="30" ds-title-
length="30" ds-content-length="40" ds-label-exclude="news,beauty" >
             <div class="card">
             <p class="title" ds-title="1">Post Title </p>
<p class="name" ds-name="1"> Label Name</p>
              <img src="#" ds-image="1" class="card-img" />
               <i ds-time="1"> Last seen</i>
               <i ds-commentcount="1">Total Comments<i/>
                <span ds-author="1">Author Name </span>
                 <audio controls><source src="#" ds-audio="1" type="mp3" ></audio>
                  <video controls><source src="#" ds-video="1" type="mp3" ></video>
```

```
      <a href="#" ds-link=”1">Post Link </a>

      <a href="#" ds-link1=”1">Post Link1 </a>

       <a href="#" ds-link2=”1">Post Link2 </a>

        <a href="#" ds-link3=”1">Post Link3 </a>

         <a href="#" ds-link4=”1">Post Link4 </a>

        <a href="www.facebook.com?" ds-sociallink=”1">Post Social Link </a>

          <a href="www.facebook.com?" ds-sociallink1=”1">Post Social Link1 </a>

           <a href="www.facebook.com?" ds-sociallink2=”1">Post Social Link2 </a>

          <a href="www.facebook.com?" ds-sociallink3=”1">Post Social Link3 </a>

        <a href="www.facebook.com?" ds-sociallink4=”1">Post Social Link4 </a>

         <p ds-firstletter=”1"> Post First Letter </p>

        <i ds-date=”1">Post Date </i>

        <i ds-id=”1"> Post Id</i>

        <input type="hidden" ds-id=”1"/>

         <a data-target="#name" href="#">Label Name </a>

         <a data-target="##name" href="#">Label Name </a>

  </div>

</div>
```

8. *Post Types Tag (posttypes)*

PostTypes Tag is a dshtml tag that retrieves or displays the posts based on the specified label name from the post section of the CMS.

It is also case sensitive (lower case).

It can be written using two methods (1) <posttypes> children element </posttypes> (2) either <div> or <ul> tag. (ds-parent must be set to the label name).

Attributes: Name(ds-name), Description(ds-description), Widget, Index(ds-index), Title Length(ds-title-length), Content Length(ds-title-length), Count(ds-count)

Parameters: Title, Name, Content, Image, Video, Audio, Link, Link1, Link2, Link3, Link4, Author, Date, CommentCount, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code Example

The snippet code below displays 30 posts based on the specified label name with title length of 30 and content length of 40 on web browser. Using the two methods.

First Method

```
<posttypes ds-type="news"  ds-name="News Post" ds-description="Display News Posts" ds-count="30" ds-title-length="30" ds-content-length="40" >

               <div class="card">

               <p class="title" id="title>Post Title </p>

                 <p class="name" id="name"> Label Name</p>

                <img src="#" id="image" class="card-img" />

                 <i id="time"> Last seen</i>

                 <i id="commentcount">Total Comments<i/>

                  <span id="author">Author Name </span>

                   <audio controls><source src="#" id="audio" type="mp3" ></audio>
```

```
          <video controls><source src="#" id="video" type="mp3" ></video>
         <a href="#" id="link">Post Link </a>
         <a href="#" id="link1">Post Link1 </a>
          <a href="#" id="link2">Post Link2 </a>
           <a href="#" id="link3">Post Link3 </a>
            <a href="#" id="link4">Post Link4 </a>
           <a href="www.facebook.com?" id="sociallink">Post Social Link </a>
             <a href="www.facebook.com?" id="sociallink1">Post Social Link1 </a>
              <a href="www.facebook.com?" id="sociallink2">Post Social Link2 </a>
             <a href="www.facebook.com?" id="sociallink3">Post Social Link3 </a>
           <a href="www.facebook.com?" id="sociallink4">Post Social Link4 </a>
            <p id="firstletter"> Post First Letter </p>
           <i id="date">Post Date </i>
           <i id="id"> Post Id</i>
           <input type="hidden" id="id"/>
            <a data-target="#name" href="#">Label Name </a>
            <a data-target="##name" href="#">Label Name </a>
     </div>
   </posttypes>
```

Second Method

```
<div ds-parent="news" ds-name="News Post" ds-description="Display News Posts" ds-count="30" ds-title-length="30" ds-content-length="40">
        <div class="card">
        <p class="title" ds-title="1">Post Title </p>
```

```
<p class="name" ds-name=”1"> Label Name</p>

            <img src="#" ds-image=”1" class="card-img" />

             <i ds-time=”1"> Last seen</i>

             <i ds-commentcount=”1">Total Comments<i/>

              <span ds-author=”1">Author Name </span>

               <audio controls><source src="#" ds-audio="1" type="mp3" ></audio>

                <video controls><source src="#" ds-video=”1" type="mp3" ></video>

               <a href="#" ds-link=”1">Post Link </a>

               <a href="#" ds-link1=”1">Post Link1 </a>

                <a href="#" ds-link2=”1">Post Link2 </a>

                 <a href="#" ds-link3=”1">Post Link3 </a>

                  <a href="#" ds-link4=”1">Post Link4 </a>

                 <a href="www.facebook.com?" ds-sociallink=”1">Post Social Link </a>

                   <a href="www.facebook.com?" ds-sociallink1=”1">Post Social Link1 </a>

                    <a href="www.facebook.com?" ds-sociallink2=”1">Post Social Link2 </a>

                   <a href="www.facebook.com?" ds-sociallink3=”1">Post Social Link3 </a>

                 <a href="www.facebook.com?" ds-sociallink4=”1">Post Social Link4 </a>

                  <p ds-firstletter=”1"> Post First Letter </p>

                 <i ds-date=”1">Post Date </i>

                 <i ds-id=”1"> Post Id</i>

                 <input type="hidden" ds-id=”1"/>

                  <a data-target="#name" href="#">Label Name </a>

                  <a data-target="##name" href="#">Label Name </a>

            </div>
```

```
</div>
```

9. Multiplelabels Tag (multiplelabels)

Multiplelabels Tag is a dshtml tag that retrieves or displays posts that grouped by label name from the CMS. It a tag used to retrieve or display distinct label name from the CMS. i.e. no duplicate label name will be displayed.

It is also case sensitive (lower case).

It can be written using two methods (1) <multiplelabels> children element </multiplelabels> (2) either <div> or <ul> tag. (ds-parent must be set to label).

Attributes: Name(ds-name), Description(ds-description), Widget, Index(ds-index), Title Length(ds-title-length), Content Length(ds-title-length), Count(ds-count), Exclude (ds-label-exclude).

Parameters: Title, Name, Content, Image, Video, Audio, Link, Link1, Link2, Link3, Link4, Author, Date, CommentCount, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code Example

The snippet code below displays 30 posts with distinct label name, excluding news and beauty label name, with title length of 30 and content length of 40 on web browser. Using the two methods.

First Method

```
<multiplelabels ds-name="Multiple Labels" ds-description="Display Multiple Labels" ds-count="30" ds-title-length="30" ds-content-length="40" ds-label-exclude="news,beauty" >

        <div class="card">

        <p class="title" id="title>Post Title </p>

          <p class="name" id="name"> Label Name</p>

         <img src="#" id="image" class="card-img" />
```

```
  <i id="time"> Last seen</i>

  <i id="commentcount">Total Comments<i/>

   <span id="author">Author Name </span>

    <audio controls><source src="#" id="audio" type="mp3" ></audio>

     <video controls><source src="#" id="video" type="mp3" ></video>

    <a href="#" id="link">Post Link </a>

    <a href="#" id="link1">Post Link1 </a>

     <a href="#" id="link2">Post Link2 </a>

      <a href="#" id="link3">Post Link3 </a>

       <a href="#" id="link4">Post Link4 </a>

      <a href="www.facebook.com?" id="sociallink">Post Social Link </a>

        <a href="www.facebook.com?" id="sociallink1">Post Social Link1 </a>

         <a href="www.facebook.com?" id="sociallink2">Post Social Link2 </a>

        <a href="www.facebook.com?" id="sociallink3">Post Social Link3 </a>

      <a href="www.facebook.com?" id="sociallink4">Post Social Link4 </a>

       <p id="firstletter"> Post First Letter </p>

      <i id="date">Post Date </i>

      <i id="id"> Post Id</i>

      <input type="hidden" id="id"/>

       <a data-target="#name" href="#">Label Name </a>

       <a data-target="##name" href="#">Label Name </a>

 </div>

</multiplelabels>
```

Second Method

```
<div ds-parent=”label” ds-name="Multiple Label" ds-description="Display Multiple labels" ds-count="30" ds-title-length="30" ds-content-length="40" ds-label-exclude="news,beauty" >

                <div class="card">

                <p class="title" ds-title=”1”>Post Title </p>

<p class="name" ds-name=”1”> Label Name</p>

                 <img src="#" ds-image=”1" class="card-img" />

                  <i ds-time=”1"> Last seen</i>

                  <i ds-commentcount=”1”>Total Comments<i/>

                   <span ds-author=”1">Author Name </span>

                    <audio controls><source src="#" ds-audio="1" type="mp3" ></audio>

                     <video controls><source src="#" ds-video=”1" type="mp3" ></video>

                    <a href="#" ds-link=”1”>Post Link </a>

                    <a href="#" ds-link1=”1”>Post Link1 </a>

                     <a href="#" ds-link2="1">Post Link2 </a>

                      <a href="#" ds-link3=”1">Post Link3 </a>

                       <a href="#" ds-link4=”1”>Post Link4 </a>

                      <a href="www.facebook.com?" ds-sociallink=”1”>Post Social Link </a>

                        <a href="www.facebook.com?" ds-sociallink1=”1”>Post Social Link1 </a>

                         <a href="www.facebook.com?" ds-sociallink2=”1”>Post Social Link2 </a>

                        <a href="www.facebook.com?" ds-sociallink3=”1”>Post Social Link3 </a>

                      <a href="www.facebook.com?" ds-sociallink4=”1”>Post Social Link4 </a>

                       <p ds-firstletter=”1”> Post First Letter </p>

                      <i ds-date=”1”>Post Date </i>

                      <i ds-id=”1"> Post Id</i>
```

```
            <input type="hidden" ds-id="1"/>

             <a data-target="#name" href="#">Label Name </a>

             <a data-target="##name" href="#">Label Name </a>

         </div>

        </div>
```

10. Multiple Comments Tag(multiplecomments)

Multiple Comments Tag is a dshtml tag that retrieves the all comments from comment section of the CMS.

It is also case sensitive (lower case).

It can be written using two methods (1) <multiplecomments> children element </multiplecomments> (2) either <div> or <ul> tag. (ds-parent must be set to comment).

Attributes: Name(ds-name), Description(ds-description), Index(ds-index) Count(ds-count),

Parameters: Name, Content, Image, Date, FirstLetter, Id, Time

Code example

The snippet code below displays all comments made by users on web browser. Using the two methods.

First Method

```
<div class="response">

             <h4>Responses</h4>

            <multiplecomments>
            <div class="media response-info mb-3">

              <div class="media-left response-text-left">

                <a href="#">

                  <img  class="media-object"  src="images/icon1.png"  alt=""  id="image"  style="width:  50px;
height: 50px;">

                </a>

                <h5><a href="#" id="name">Admin</a></h5>

              </div>

              <div class="media-body response-text-right">

                <p  id="content">Lorem  ipsum  dolor  sit  amet,  consectetur  adipisicing  elit,There  are  many
variations of passages of Lorem Ipsum available,

                  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

                <ul>

                  <li id="date">October 25, 2016</li>



                </ul>



              </div>

              <div class="clearfix"> </div>

            </div>




           </multiplecomments>

          </div>
```

Second Method

```
<div class="response">

                  <h4>Responses</h4>

                 <div class=”card-group” ds-parent=”comment”>

                  <div class="media response-info mb-3">

                     <div class="media-left response-text-left">

                        <a href="#">

                           <img class="media-object" src="images/icon1.png" alt="" ds-image=”1" style="width: 50px;
height: 50px;">

                        </a>

                        <h5><a href="#" ds-name=”1">Admin</a></h5>

                     </div>

                     <div class="media-body response-text-right">

                        <p ds-content=”1">Lorem ipsum dolor sit amet, consectetur adipisicing elit,There are many
variations of passages of Lorem Ipsum available,

                           sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

                        <ul>

                           <li ds-date=”1">October 25, 2016</li>

                        </ul>

                     </div>

                     <div class="clearfix"> </div>

                  </div>

                 </div>

               </div>
```

11. Multiple Pages Tag(multiplepages)

Multiple Pages Tag is a dshtml tag that retrieves or displays all the pages from the page section of the CMS

It is also case sensitive (lower case).

It can be written using two methods (1) <multiplepages> children element </multiplepages> (2) either <div> or <ul> tag. (ds-parent must be set to page).

Attributes: Name(ds-name), Description(ds-description), Widget, Index(ds-index), Count(ds-count)

Parameters: Name, Image, Audio, Video, Content Link, Link1, Link2, Link3, Link4, Date, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code example

The snippet code below displays all pages from the CMS, on web browser. Using the two methods.

First Method

```
<div class=”container”>
 <multiplepages ds-name="Multiple Pages" ds-description="Display Multiple Pages">
            <li class="treeview">
              <a href="#" id="link">
                <span id="name"></span>
              </a>
            </li>
          </multiplepages>
</div>
```

Second Method

```
<div class=”container” ds-parent=”page” ds-name=”Multiple Pages” ds-description=”Display Multiple pages”>

                <li class="treeview">

                    <a href="#" ds-link=”1">

                        <span ds-name=”1"></span>

                    </a>

                 </li>

</div>
```

12. Pagination Tag(pagination)

Pagination Tag is a dshtml tag that retrieve or display the total number of posts divided by the total number of displayed posts from the post section of the CMS. i.e. page index.

It is also case sensitive (lower case).

It can be written using two methods (1) <pagination> children element </pagination> (2) either <div> or <ul> tag. (ds-parent must be set to pagination).

Attributes: Name(ds-name), Description(ds-description)

Parameters: PageIndex, Link

Code example

The snippet code below displays all the number pages from the post section of the CMS, on web browser. Using the two methods.

First Method

```
<ul class="pagination mt-5">
```

```
        <pagination>
            <li class="page-item">
                <a class="page-link" id="link"><span id="pageindex"></span></a>
            </li>

        </pagination>
    </ul>
```

Second Method

```
<ul class="pagination mt-5" ds-parent=”pagination”>
            <li class="page-item">
                <a class="page-link" ds-link=”1”><span ds-pageindex=”1”></span></a>
            </li>
    </ul>
```

13. Logout Tag (logout)

Logout Tag is a dshtml tag use for unauthorizing user from the current context.

It is also case sensitive (lower case).

It can be written using two methods (1) <logout> children element </logout> (2) using ds-link-type attribute to a tag and logout as value.

It has no attribute.

Parameter: Link.

Code example

The snippet code below unauthorize user from the current context. Using the two methods.

First Method

```
    <logout>
         <a href="#" id="link"><h1>Logout</h1></a>
       </logout>
```

Second Method

```
<a href="#" ds-link-type="logout"><h1>Logout</h1></a>
```

14. Single Post Tag (singlepost)

Single Post Tag is a dshtml tag that retrieves or displays single post from the post section of the CMS.

It can only be used in post.html file.

It is also used when reading post content through the url on a web browser.

It is also case sensitive (lower case).

It can be written using two methods (1) <singlepost> children element </singlepost> (2) either <div> or <ul> tag. (ds-single-parent must be set to post).

Attributes: Name(ds-name), Description(ds-description)

Parameters: Title, Name, Image, Content, Link, Link1, Link2, Link3, Link4, Author, Date, CommentCount, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code example

The snippet code below displays a post content for reading on web browser. Using the two methods.

First Method

```
<singlepost ds-name="View Post Tag" ds-description="Shows the view post" >

          <div class="single-left1">

            <img src="/images/blog.jpg" id="image" alt=" " class="img-fluid" style="width: 80%; height: 400px;
margin-bottom: 15px" />

            <h6 class="blog-first text-dark my-4">

              <i class="far fa-user mr-2"></i>

              <i id="author"></i>

            </h6>

            <ul class="blog_list my-3">

              <li id="date"></li>



              <li>

                <a href="#">

                  <i class="far fa-comments mr-1"></i>

                  <i id="commentcount"></i></a>

              </li>

            </ul>

            <h5 class="card-title">
```

```
                    <a href="#" class="text-dark" id="link">
                      <span id="title"></span> </a>
                  </h5>
                  <p id="content"></p>
               </div>
               </singlepost>
```

Second Method

```
<div ds-single-parent="post" ds-name="View Post Tag" ds-description="Shows the view post" >
               <div class="single-left1">
                  <img src="/images/blog.jpg" ds-mage="1" alt=" " class="img-fluid" style="width: 80%; height: 400px;
margin-bottom: 15px" />
                  <h6 class="blog-first text-dark my-4">
                    <i class="far fa-user mr-2"></i>
                    <i ds-author="1"></i>
                  </h6>
                  <ul class="blog_list my-3">
                    <li ds-date="1"></li>


                    <li>
                       <a href="#">
                          <i class="far fa-comments mr-1"></i>
                          <i ds-commentcount="1"></i></a>
                    </li>
```

```
      </ul>

      <h5 class="card-title">

        <a href="#" class="text-dark" ds-link="1">

          <span ds-title="1"></span> </a>

      </h5>

      <p ds-content="1"></p>

    </div>

    </div>
```

15. Single Page Tag (singlepage)

Single Page Tag is a dshtml tag that retrieves or displays single page from the CMS

It is also case sensitive (lower case).

It can only be used in page.html file.

It can be written using two methods (1) <singlepage> children element </singlepage> (2) either <div> or <ul> tag. (ds-single-parent must be set to page).

Attributes: Name(ds-name), Description(ds-description).

Parameters: Name, Image, Content, Link, Link1, Link2, Link3, Link4, Date, FirstLetter, Id, SocialLink, SocialLink1, SocialLink2, SocialLink3, SocialLink4, Time, #Name, ##Name.

Code example

The snippet code below displays a page content for reading on web browser. Using the two methods.

First Method

```
<singlepage ds-name="View Page Tag" ds-description="Shows the view page">
          <div class="container-fluid" style="min-height: 1500px">
            <h5 class="text-center" id="name"></h5>

            <span id="content" class="mt-3"></span>

          </div>
        </singlepage>
```

Second Method

```
<div ds-single-parent=”page” ds-name="View Page Tag" ds-description="Shows the view page">
          <div class="container-fluid" style="min-height: 1500px">
            <h5 class="text-center" ds-name=”1"></h5>

            <span ds-content=”1" class="mt-3"></span>

          </div>
        </div>
```

16. Site Name Tag (sitename) and Username(username)

Site Name Tag is a dshtml tag used when displaying Title from the Setting section of the CMS.

It is also case sensitive (lower case).

It can be written in two ways (1) <sitename> </sitename> (2) using I, span, h1..h6 and p with ds-username or ds-sitename as attribute

It has no attribute.

It has no parameter.

Code example

The snippet code below displays the site title and authorized username.

```
  <div class="logo">

           <a href="/"><h1><sitename></sitename></h1></a>

 <span ds-sitename=" sitename"></span>

<h1 ds-username=" username"></h1>

      </div>
```

17. Widget Tag (widget)

Widget Tag is a dshtml tag which instruct the CMS to interpret the children element as widget.

It Is a tag use when indicating the children element as widget and show on adding widget section of the CMS.

It has no attribute and parameters

It is also case sensitive (lower case).

It can be written as <widget> Children Element</widget>.

Code example

The snippet code below instructs the CMS to interpret posttypes tag as widget.

```
       <widget>

    <posttypes ds-type="trending" ds-name="Toggle Recent Posts" ds-description="Toggle Recent Posts" ds-label-
exclude="event,best" ds-count="50" widget="widget">

        <div class="media">

            <a href="#">

                <img class="d-flex mr-3 img-fluid" src="#" id="image" alt="Generic placeholder image" />

            </a>

            <div class="media-body">

                <div class="news-title">

                    <h2 class="title-small">

                        <a href="#" id="link" class="text-black">

                            <span id="title"></span></a>

                    </h2>

                </div>

                <div class="news-auther">

                    <small class="text-time">

                        <em id="time">10 mins ago</em>

                    </small>

                </div>

            </div>

        </div>

       </posttypes>

  </widget>
```

18. Layout Tag (layout)

Layout Tag is a dshtml tag used as design framework for a web page.

It is a tag that gives a web page layout.

It is written inside body element

It is used immediately after body html tag.

It encapsulates the structure of the web page.

It is also case sensitive (lower case).

It has no parameter.

Attribute: Class (html attribute), Id (html attribute), TagName (html tag), Style (html attribute).

It can be written as < layout > Children Tag</ layout >.

Code example

```
<layout>

<top>

<!— Top Content -->

</top>

<left>

<!— Left Content -->

</left>

 <right>

<!— Right Content -->

</right>
```

```
<bottom>

<!— Bottom Content -->

</bottom>

  </layout>
```

19. Top Tag (top)

Top Tag is a dshtml tag which is used for positioning the children elements to the top part of the layout.

It is written inside layout element.

It is also case sensitive (lower case).

It has no parameter.

Attribute: Class (html attribute), Id (html attribute), TagName (html tag), Style (html attribute).

It can be written as < top> Children Element</top>.

Code example

```
<top>

<!— Content -->

</top>
```

20. Left Tag(left)

Left Tag is a dshtml tag which is used for positioning the children elements to the left part of the layout.

It is written inside layout element.

It is also case sensitive (lower case).

It has no parameter.

Attribute: Class (html attribute), Id (html attribute), TagName (html tag), Style (html attribute).

It can be written as <left> Children Element</left>.

Code example

```
<left>

<!— Content -->

</left>
```

21. ***Right Tag (right)***

Right Tag is a dshtml tag which is used for positioning the children elements to the right part of the layout.

It is written inside layout element.

It is also case sensitive (lower case).

It has no parameter.

Attribute: Class (html attribute), Id (html attribute), TagName (html tag), Style (html attribute).

It can be written as <right> Children Tag</right>.

Code example

```
<right>

<!— Content -->

</right>
```

22. Bottom Tag(bottom)

Bottom Tag is a dshtml tag which is used for positioning the children elements to the bottom part of the layout.

It is written inside layout element.

It is also case sensitive (lower case).

It has no parameter.

Attribute: Class (html attribute), Id (html attribute), TagName (html tag), Style (html attribute).

It can be written as <bottom> Children Tag</bottom>.

Code example

```
<bottom>

<!— Content -->

</bottom>
```

23. Form Tag (form)

Form tag is a html tag which perform a lot of operations or actions in dshtml

It is the most powerful tag in dshtml which perform server actions (Method: Get or Post).

I. Search i.e. searching for a content through its title, directly from the CMS.
II. Comment i.e. Commenting on a post.
III. Contact i.e. Sending message to the administration.
IV. Login i.e. Form for Authorize user.
V. Register i.e. Form for Registration of user.

VI. Subscribe i.e. Registering email for Newsletter update.

VII. Profile i.e. Updating User Profile.

It takes ds-action attribute to perform those operates.

Code example 1

The snippet code below creates a search form and search content by the title from the CMS and display on web browser.

```
<form ds-action=”search” ds-name=”Search” ds-description=”Search Operation”>

    <p ds-error=”1”>  </p>

<input type=”text” ds-title=”1”>

<input type=”submit” value=”Search”>

</form>
```

Code example 2

The snippet code below creates a subscribe form and subscribe user by the email to the newsletter or update.

```
<form ds-action=”subscribe” ds-name=”Subscribe” ds-description=”Subscribe Operation”>

    <p ds-error=”1”> </p>

<input type=”text” ds-email=”1”>

<input type=”submit” value=”Subscribe”>

</form>
```

Code example 3

The snippet code below creates a comment form and allow user to make comment with the name and content (comment).

```
<form ds-action="comment" ds-name="Comment" ds-description="Comment Operation">

<input type="text" ds-name="1">

<input type="text" ds-content="1">

<input type="submit" value="Comment">

</form>
```

Code example 4

The snippet code below creates a contact form and to contact the administration with the email, subject and message.

```
<form ds-action="contact" ds-name="Contact" ds-description="Contact Operation">

   <p ds-error="1"> </p>

<input type="text" ds-email="1">

<input type="text" ds-subject="1">

<input type="text" ds-message="1">

<input type="submit" value="Contact Us">

</form>
```

Code example 5

The snippet code below creates a register form and allow user registration.

```
<form ds-action=”register” ds-name="Register" ds-description="Register Page">

  <p id=”error”>  </p>

                    <h5>User Name:</h5>

                         <input type="text" ds-username="1">

                         <h5>Email Address:</h5>

                         <input type="email"   ds-email=”1">

                         <h5>Phone Number:</h5>

                         <input type="text"   ds-phonenumber=”1">

                         <h5>First Name:</h5>

                         <input type="text"   ds-firstname=”1">

                         <h5>Last Name:</h5>

                         <input type="text"   ds-lastname=”1">

                         <h5>Password:</h5>

                         <input type="password"   ds-password=”1">

                         <h5>User Image:</h5>

                         <input type="file"   ds-userimage=”1">

                         <input type="submit" value="Register">



                      </form>
```

Code example 6

The snippet code below creates a login form and allow user authorization.

```
<form ds-action=”login” ds-name="Login" ds-description="Login Page">
```

```
  <p id=”error”>  </p>

                  <h5>User Name:</h5>

                       <input type="text" ds-username="1">

                       <h5>Password:</h5>

                       <input type="password"   ds-password=”1">

                      <a href=”#” ds-forget=”1”>Forget Password</a>

                       <input type="submit" value="Login">



                    </form>
```

Code example 7

The snippet code below creates a profile form and allow user to update their profile.

```
<form ds-action=”profile” ds-name="Update Profile" ds-description="Update Profile">

    <p id=”error”>  </p>

                  <h5>User Name:</h5>

                       <input type="text" ds-username="1">

                       <h5>Email Address:</h5>

                       <input type="email"   ds-email=”1">

                       <h5>Phone Number:</h5>

                       <input type="text"   ds-phonenumber=”1">

                       <h5>First Name:</h5>

                       <input type="text"   ds-firstname=”1">

                       <h5>Last Name:</h5>

                       <input type="text"   ds-lastname=”1">
```

```
        <h5>Password:</h5>

        <input type="password"   ds-password=”1">

        <h5>User Image:</h5>

        <input type="file"   ds-userimage=”1">

        <input type="submit" value="Register">


      </form>
```

24. Anchor Tag (a)

Anchor tag is a html tag which handle links and replace the href value to the specified ds-link-type.

Ds-link-type has the following value

I. Logout I.e. It will change the href value to Logout link.
II. Login I.e. It will change the href value to Login link.
III. Register I.e. It will change the href value to Register link.
IV. Forget I.e. It will change the href value to Forget Password link.

It takes ds-link-type attribute to handle the link.

Code example 1

The snippet code below changes the href value to logout link

```
<a href="#" ds-link-type=”logout”><h1>Logout</h1></a>
```

Code example 2

The snippet code below changes the href value to login link

```
<a href="#" ds-link-type=”login”><h1>Already own account</h1></a>
```

Code example 3

The snippet code below changes the href value to register link

```
<a href="#" ds-link-type=”register”><h1>Create Account</h1></a>
```

Code example 4

The snippet code below changes the href value to Forgot Password link

```
<a href="#" ds-link-type=”forget”><h1>Forgot password</h1></a>
```

25. Profile Tag (profile)

i. Profile Tag is a dshtml tag which is used to retrieved user details.

ii. It is case sensitive (lower case).

iii. It has no attributes.

iv. Parameters: UserName, Email, UserImage, Role, Phone Number, FirstName, LastName

Code example

The snippet code below show users details

```
<profile>
```

```
<p id="username"></p>

<img src="#" id="userimage" style="width: 50px; height: 50px"/>

<p id="firstname"></p>

<p id="email"></p>

<p id="phonenumber"></p>

<p id="role"></p>

<p id="lastname"></p>

</profile>
```

Introduction to Structured Query Language (SQL)

What is Structured Query Language (SQL)?

Structured Query Language (SQL) is a standardize programming language that is used to manage relational database and perform various operations on the data in them.

Structured Query language (SQL) is used for the following

I. Modifying database table and index structure.
II. Adding, Updating and Deleting rows of a data
III. Retrieving subsets of information from within relational database management systems.

SQL queries and other operations take the form of commands written as statements and are aggregated into programs that enable users to add, modify, or retrieve data from the database table.

A table is the must basic unit of a database and consist of rows and columns of the data. A single table holds records and each record is stored in a row of the table. Table are the most used type of database objects, or structure that hold on reference data in a relational database. Each column in a table corresponds to category of data. For example, Customer name or address, while each row contains a data value for the intersecting column.

Relational database is relational because they are composed of table that relate to each other. For example, a SQL database use for customer service can have one table for customer name or address and other table can hold information about the specific purchases, product codes and customer contact. A table used to track customer contact usually uses a unique customer identifier called key or primary key to reference the customer's record in a separate table used to store customer data, such as name and contact information.

The Database Management Systems software used on the CMS is called DbConnection.

DbConnection

DbConnection is a Database Management Systems Software for relational database which create connection to the database for storing, retrieving and running of queries on data. It serves as an interface between an end-user and a database, allowing users to create, read, update and delete data in the database.

DbConnection return DbContext which is used to store, modify, delete and retrieve data from the database through queries.

DbConnection is a Software developed under javascript. i.e. it can be access through javascript.

Syntax

The syntax for creating or initializing DbConnection

```
let dbConnection = new DbConnection();
```

it is parameter less function.

Instance Properties

DbConnection.state.

Returns the current state of the connection i.e. Connected, Connecting or Disconnected.

DbConnection.context.

Returns the DbContext which provide an interface for running queries, adding, modifying and deleting from database.

Instance Methods

DbConnection.open(callback)

This method open connection for database to accept queries. it's a void function, Callback is the function to be invoked when the connection is ready to accept the incoming queries.

DbConnection.close()

This method close connection to the database. it's a void function,

DbContext

DbContext is a software that manipulate data from the database. I.e. Add, Modify, Delete and Retrieve through queries.

Syntax

The syntax for creating or initializing DbContext

```
let dbContext = dbConnection.context;
```

Instance Methods

DbContext.ExecuteQuery(query,..values, callback)

Execute the query with the values and return the callback function.

ExecuteQuery can only execute add, modify, drop, delete query,

Callback return the sate of the operation, its return function with two parameters,

done (Boolean) and error (string).

DbContext.ExecuteQueryReader(query,..values, callback)

Execute the query with the values and return the callback function.

ExecuteQueryReader can only execute select query,

Callback return the values in array, its return function with a parameter,

Reader (Array).

SQL Queries and Examples

Creating of table, Columns and Adding values Query

The query below creates a table with the columns and insert the rows/values, if the table is not existing in the database otherwise, it will insert the rows/values.

Add Column1=Value1, Column2=Value2, Column3=Value3 Into TableName.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below creates student table with their name, age, address and insert their values if student table is not existing in the database otherwise, it will insert their values.

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let addQuery=”add Name=@1, Age=@2, Address=@3 into Student”;

context.executeQuery(addQuery,”SomeoneName”,15,”no 12 someone
address”,(done,error)=>{ alert(error); });

});}
```

Where the name, age, address are the columns and @number is the value identifier.

Creating of table, Columns and Adding values with Foreign Key Query

The query below creates a table with the columns and insert the values, and create relationship with an existing table, if the table is not existing in the database otherwise, it will insert the values.

Add Column1=Value1, Column2=Value2, Column3=Value3 Into TableName Join ForeignTable Where ForeignColumn=ForeignValue

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below creates student table with their name, age, address and insert their values and create a relationship with Course table, if student table is not existing in the database otherwise, it will insert their values.

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let addQuery=”add Name=@1, Title=@2 into Course Join Student where ID=@3”;

context.executeQuery(addQuery,”SomeoneName”,”Introduction to Course”,
604155018 ,(done,error)=>{ alert(error); });

});}
```

Where the name, age, address are the columns and @number is the value identifier.

Adding Unique Key to a Column of an Existing Table Query

The query below add unique key to a particular or specified column name of an existing table in the database.

Add Unique(ColumnName) Into TableName

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below add unique key to name in student table.

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let addQuery="add unique(Name) into Student";

context.executeQuery(addQuery,(done,error)=>{ alert(error); });

});}
```

Where the name is the column and @number is the value identifier.

Adding Column of an Existing Table Query

The query below add new column of an existing table in the database.

Add Column(ColumnName) Into TableName

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below add RegistrationDate to student table.

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let addQuery=”add Column(RegistrationDate) into Student”;

context.executeQuery(addQuery,(done,error)=>{ alert(error); });

});}
```

Where the Registration is the column and @number is the value identifier.

Modifying Rows of an Existing Table Query

The query below modifies the row of the specified column name and condition from an existing Table in a database

Modify Column1=Value1, Column2=Value2, Column3=Value3 From TableName Where Column3=Value3.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize. And where is a conditional keyword which can be used with =, !=, >, <, AND, OR operators.

Example

The snippet code below modify name, address where the age is 12 from student table

```
window.onload = function() {
```

```
let dbConnection= new DbConnection();

let context = dbConnection.context;


dbConnection.open(function() {

let modifyQuery=”modify Name=@1, Address=@2 from Student where Age=@3”;

context.executeQuery(modifyQuery,”NewName”,”no 12 someone
address”,12,(done,error)=>{ alert(error); });

});}
```

Where the name, age, address are the columns and @number is the value identifier.

Modifying Primary Key to a Column of an Existing Table Query

The query below modifies the primary key to the specified column name from an existing Table in a database

Modify Primary(ColumnName) from TableName.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below change primary key to Name from Course table

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let modifyQuery=”modify primary(Name) from Course”;

context.executeQuery(modifyQuery,(done,error)=>{ alert(error); });
});}
```

Note: Primary key won't be modify if has been used for foreign Key.

Where the Age is the column and @number is the value identifier.

Modifying Unique Key to a Column of an Existing Table Query

The query below modifies the unique key to the specified column name from an existing Table in a database

Modify Unique(ColumnName) from TableName.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below change unique key to age from student table

```
window.onload = function() {
let dbConnection= new DbConnection();
let context = dbConnection.context;
dbConnection.open(function() {
let modifyQuery="modify unique(Age) from Student";
context.executeQuery(modifyQuery,(done,error)=>{ alert(error); });
});}
```

Where the Age is the column and @number is the value identifier.

Deleting Row of an Existing Table Query

The query below delete row with given condition from an existing Table in a database

Delete From TableName Where Column1=Value1.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize. And where is a conditional keyword which can be used with =, !=, >, <, AND, OR operators.

Example

The snippet code below delete row where the age is 12 from student table

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let deleteQuery=”delete from Student where Age=@1”;

context.executeQuery(deleteQuery,12,(done,error)=>{ alert(error); });

});}
```

Where age is the column and @number is the row/value identifier.

Drop Unique Key from a Column of an Existing Table Query

The query below removes unique key from an existing column with unique key, from an existing Table in a database

Drop Unique(ColumnName) from TableName.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below removes unique key from Age in student table

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let deleteQuery=”drop unique(Age) from Student”;

context.executeQuery(deleteQuery,(done,error)=>{ alert(error); });

});}
```

Where the Age is the column and @number is the row/value identifier.

Drop Column from an Existing Table Query

The query below deletes specified column from an existing Table in a database

Drop Column(ColumnName) from TableName.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below remove RegistrationDate from student table

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let deleteQuery=”drop Column(RegistrationDate) from Student”;

context.executeQuery(deleteQuery,(done,error)=>{ alert(error); });

});}
```

Where the RegistrationDate is the column and @number is the row/value identifier.

Selecting Rows from an Existing Table Query

The query below selects rows from an existing Table in a database.

Select all from TableName. OR

Select 20 from TableName Where Column1=Value1 And Column2=Value2.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize. And where is a conditional keyword which can be used with =, !=, >, <, AND, OR operators.

Example

The snippet code below selects all rows from student table

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let selectQuery=”select all from Student”;

context.executeQueryReader(selectQuery,(reader,error)=>{
reader.forEach((value,index)=>alert(value.name))});

});}
```

Drop an Existing Table Query

The query below drops an existing Table in a database

Drop TableName.

Note: the keywords are not case sensitive. i.e. it can be written Lowercase, Uppercase and Capitalize.

Example

The snippet code below drop student table

```
window.onload = function() {

let dbConnection= new DbConnection();

let context = dbConnection.context;

dbConnection.open(function() {

let dropQuery="drop  Student";

context.executeQuery(dropQuery,(done,error)=>{ alert(error); });

});}
```

Note: Any Column name used for Primary Key, Foreign Key or Unique key always has small storage or size.

DordStream Template

A template is a set of files commonly HTML, CSS and JAVASCRIPT which then install for usage.

The files have the unique ways of structuring in order for the CMS to interpret the files as template.

Note: Template and Theme are the same on the CMS.

This section expounds the procedures to follow in order to developed DordStream Template.

Therefore, it must contain some of the files listed below

- BANNER (banner.png or banner.jpg)
    - Banner is an image file that show or display the themes.
    - This is also giving the clear picture of the design.
    - The file extension is either .png or .jpg.
    - It is required.
    - It is case sensitive (lower case).
    - The file can be name as banner.png or banner.jpg

- DESCRIPTION (readme.txt)
    - Description is a text file that contain the full description of the application.
    - The file extension is .txt.
    - It is case sensitive (lower case).
    - It is required.
    - The file can name as readme.txt

- INDEX PAGE (index.html)
    - Index Page is a html file that contain the web Content.
    - The file extension is .html
    - It is required
    - It is case sensitive (lower case).
    - The file can name index.html

- POST PAGE (post.html)

    - This is the page that contain the single post (view post) content.
    - The file extension is .html
    - The file is required.
    - It is case sensitive (lower case).
    - The file can be name as post.html

- PAGE (page.html)
    - This is the page that contain the single page (view page) content.
    - The file extension is .html
    - The file is required.
    - It is case sensitive (lower case).
    - The file can be name as page .html

- CATEGORY PAGE (category.html)
    - This is the page that contain the post category content.
    - The file extension is .html
    - The file is required.
    - It is case sensitive (lower case).
    - The file can be name as category.html

- Error Page (error.html)
    - This is the error page which will load when error occur on the application.
    - The file extension is .html
    - It is required.
    - It is case sensitive (lower case).
    - The file can be name error.html

- Login Page(login.html)
    - This is the login page of the application.
    - The file extension is .html
    - The file is Optional.
    - It is case sensitive (lower case).
    - The file can be name as login.html.

- Register Page(register.html)
    - This is the register page of the application.
    - The file extension is .html
    - The file is Optional.
    - It is case sensitive (lower case).
    - The file can be name as register.html.

- CUSTOM PAGE (mypagecustom.html)
    - This is the page use when additional page is needed.
    - The file extension is .html
    - The file is Optional.
    - It is case sensitive (lower case).
    - The file name must contain custom.

- DORD PAGE (mypagedord.html)

- This is the page use when additional page is needed.
- The file extension is .html
- The file is Optional.
- It is case sensitive (lower case).
- The file name must contain dord.

References

- Froala Editor
- Microsoft Cooperation
- Agility Html Pack