EQUIBRAND CORPORATIONS
5/18/21

# Component v2

This document contains the design and architecture as implemented for Version2 of page content components written in c# for asp.net. The following diagrams and descriptions will explain the page content components and architecture in detail.

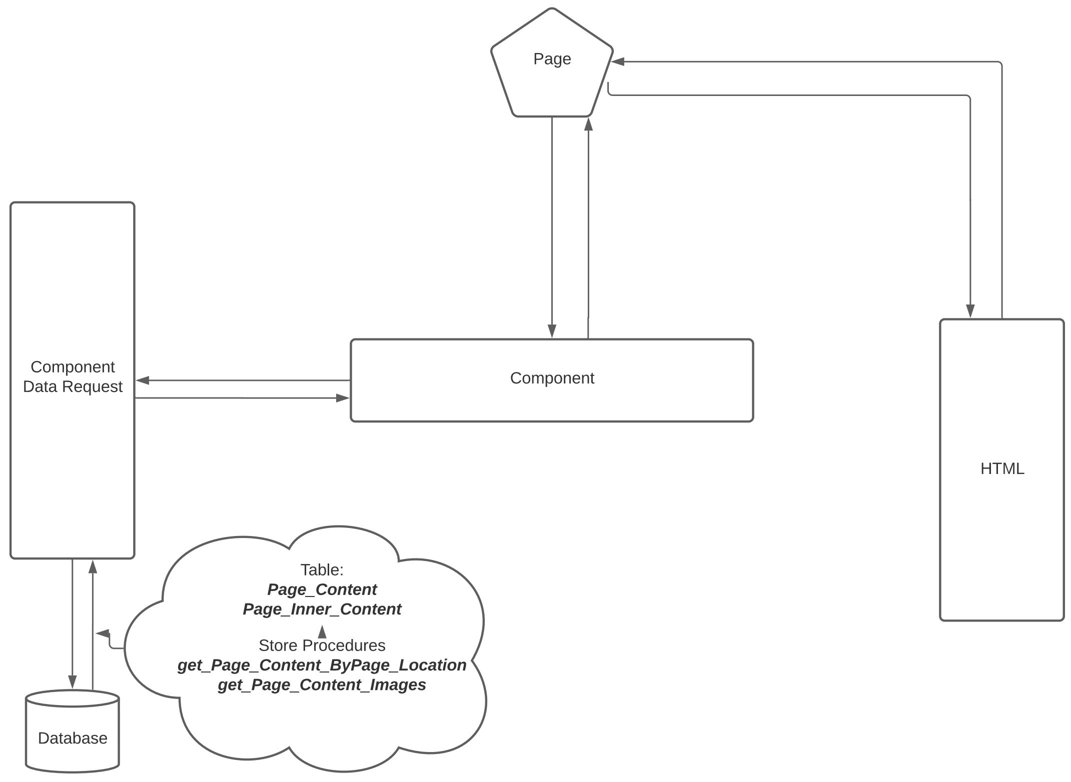


*Figure 1:* Architecture diagram for the page content component
P -> C -> CDR(data) -> C -> P -> HTML -> P

The architecture of this program is laid out such that a user can interface solely with the page portion while utilizing each piece of the components effectively. It has been architected such that the program should be able to run numerous html components, all abstracted from the user.

The Component block handles component creation and bookkeeping (TODOS: status, error handling), then the Component Data Request block simply fetches the requested data received from component providing the status of each request as well as the desired data that is retrieved by SQL Server procedures.

Finally, once the request has been completed, the component then calls HTML component. An independent class that will allow any front-end developer easily to navigate and create their desire html. (When all initialized together, the component class can pass the data received into html component to have a fully dynamic snippet).

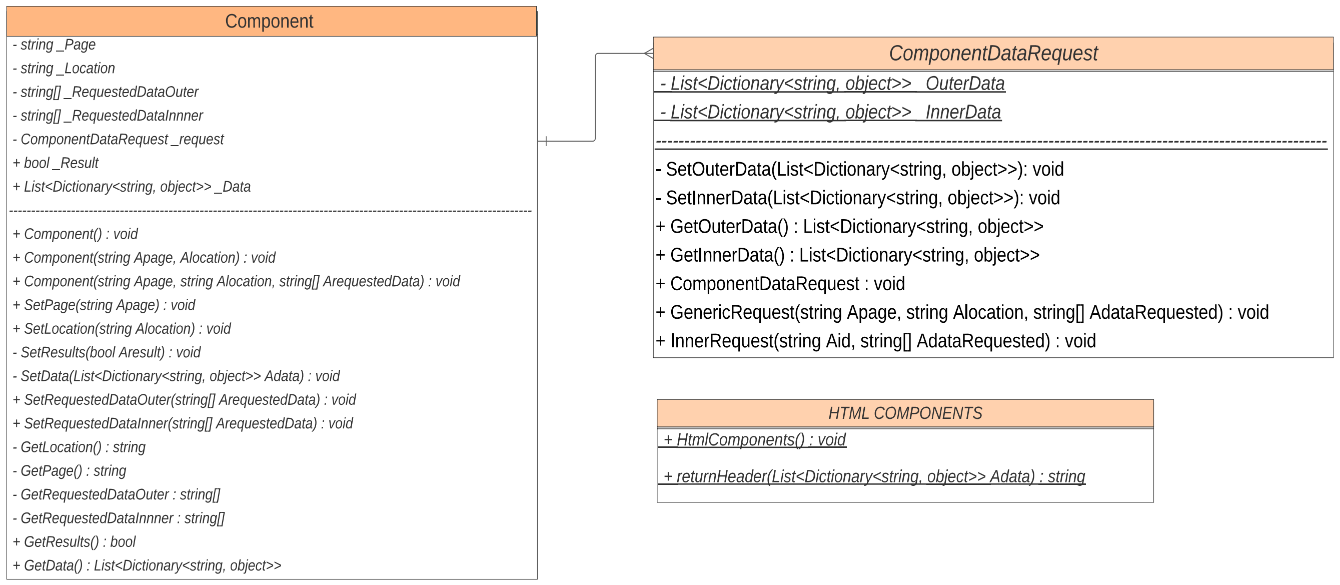

Component
- string _Page
- string _Location
- string[] _RequestedDataOuter
- string[] _RequestedDataInnner
- ComponentDataRequest _request
+ bool _Result
+ List<Dictionary<string, object>> _Data
+ Component() : void
+ Component(string Apage, Alocation) : void
+ Component(string Apage, string Alocation, string[] ArequestedData) : void
+ SetPage(string Apage) : void
+ SetLocation(string Alocation) : void
- SetResults(bool Aresult) : void
- SetData(List<Dictionary<string, object>> Adata) : void
+ SetRequestedDataOuter(string[] ArequestedData) : void
+ SetRequestedDataInner(string[] ArequestedData) : void
- GetLocation() : string
- GetPage() : string
- GetRequestedDataOuter : string[]
- GetRequestedDataInnner : string[]
+ GetResults() : bool
+ GetData() : List<Dictionary<string, object>>
ComponentDataRequest
- List<Dictionary<string, object>> OuterData
- List<Dictionary<string, object>> InnerData
- SetOuterData(List<Dictionary<string, object>>): void
- SetInnerData(List<Dictionary<string, object>>): void
+ GetOuterData() : List<Dictionary<string, object>>
+ GetInnerData() : List<Dictionary<string, object>>
+ ComponentDataRequest : void
+ GenericRequest(string Apage, string Alocation, string[] AdataRequested) : void
+ InnerRequest(string Aid, string[] AdataRequested) : void
HTML COMPONENTS
+ HtmlComponents() : void
+ returnHeader(List<Dictionary<string, object>> Adata) : string

*Figure 2:* UML Diagram of classes in the page content component

Component():

This class is designed to capture content information such as the current page and the content's location. In addition, the user will be able to establish a list of requested fields for which data from our database is needed. It was also be used to generate a componentDataRequest object and submit our desired fields to their methods, which will populated them with data.

- sendData()

  This method is responsible to send our data to our componentDataRequest object function “genericRequest” or “multipleRequest” to get our desire fields initialized with data.

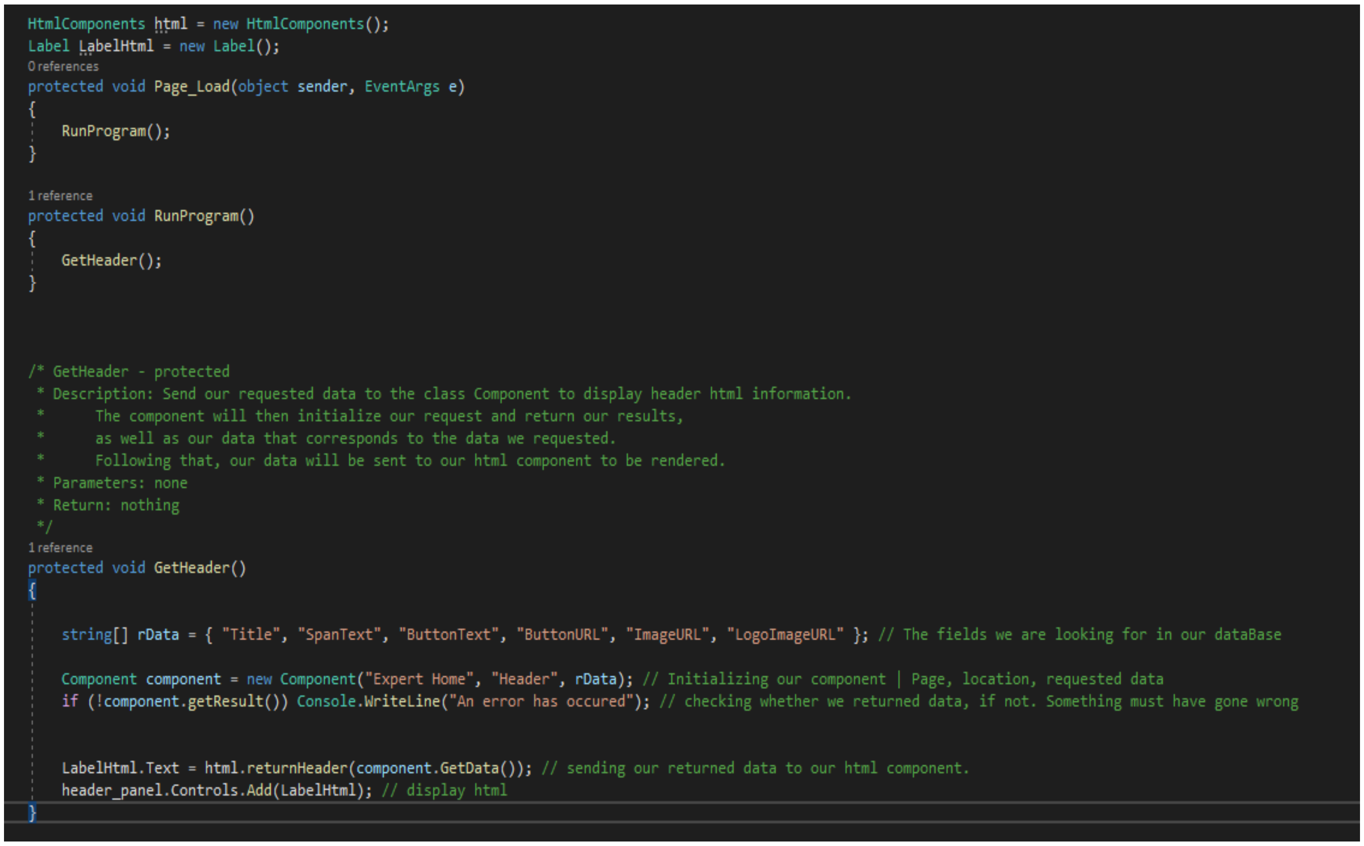
ComponentDataRequest():

This class is intended to be used to fetch and receive data from our database executing store procedures that have been created in sql server. At this moment we are using Web5 server and XpertEquine database. In addition, once the data has been fetched, we initialize it to our List<Dictionary> variable that will then be called in our component class whenever it is convenient to be used.

- GenericRequest()
  - This function calls a data table ‘Access’ and runs store procedure that will initialize our OuterData variable and uses table Page_Content.
- InnerRequest ()
  - This function calls a data table ‘Access’ and runs store procedure that will initialize our InnerData variable and uses table Page_Inner_Content.

HTMLComponent():

This class Implements functions that returns a html snippet. Keeping it abstracted from the intended page.



```
HtmlComponents html = new HtmlComponents();
Label LabelHtml = new Label();
0 references
protected void Page_Load(object sender, EventArgs e)
{
    RunProgram();
}

1 reference
protected void RunProgram()
{
    GetHeader();
}



/* GetHeader - protected
 * Description: Send our requested data to the class Component to display header html information.
 *      The component will then initialize our request and return our results,
 *      as well as our data that corresponds to the data we requested.
 *      Following that, our data will be sent to our html component to be rendered.
 * Parameters: none
 * Return: nothing
 */
1 reference
protected void GetHeader()
{

    string[] rData = { "Title", "SpanText", "ButtonText", "ButtonURL", "ImageURL", "LogoImageURL" }; // The fields we are looking for in our dataBase

    Component component = new Component("Expert Home", "Header", rData); // Initializing our component | Page, location, requested data
    if (!component.getResult()) Console.WriteLine("An error has occured"); // checking whether we returned data, if not. Something must have gone wrong


    LabelHtml.Text = html.returnHeader(component.GetData()); // sending our returned data to our html component.
    header_panel.Controls.Add(LabelHtml); // display html
}
```

*Figure 3:* Page implementation of class Component and html snippet