The Framework's Basic Life cycle

The Framework's Basic Life cycle

The framework is built upon a concept of a page life cycle, which is a general concept and not related to ApplinX host applications directly. The life cycle of a page is declared by the top context class:

JSP: com.sabratec.j2ee.framework.web.GXWebPageContext.

.NET: System.Web.UI.Page

Each page decides which is its class by declaring the root node: `<gx:html gx_context="<CONTEXT_CLASS>">` and closing it with `</gx:html>` as the end tag.

Each dynamic tag should be set with a prefix of "gx:" for example: `<gx:input id="CustomerId"/>.` The `<CONTEXT_CLASS>` is initialized and starts the page Life cycle:

- **gx_onInit (JSP) /OnInit (.NET) :**

  Used to initialize page and/or project settings, and register to events. For example setting the ApplinX configuration and registering classes to user exits.

- **gx_onLoad (JSP) /OnLoad (.NET) :**

  Used to fill the tags with run-time data from data sources. Host data will be used in lower inheritance level to fill the tags with host data. Page.IsPostBack (.NET)/ gx_isPostBack (JSP) can determine if the page is first called, or submitted back to itself.

- **Post back:**

  Performed automatically by the framework. Each page is submitted to itself using server side buttons/links:

  JSP:

  ```
  <gx:input id= myBtn  onserverclick= <FUNC_NAME> />
  <gx:a id= myLinkBtn onserverclick= <FUNC_NAME> >Send</a>
  ```

  .NET:

  ```
  <input id="myBtn" runat="server" onserverclick="< FUNC_NAME>"/>
  <a id="myLinkBtn" runat="server" onserverclick="<FUNC_NAME>">Send</a>
  ```

  The `<FUNC_NAME>` will be activated automatically upon a user click, and it will be used to perform update actions, re-queries or jumping to another page using redirect. In the context class the developer should hold a function in the format:

  ```
  Public void <SERVER_FUNC>(){
        // code in response to the        server click event.
  }
  ```

  Can be also fired using JavaScript using: `gx_postBack("<SERVER_FUNC>");`

  ```
  public void <SERVER_FUNC>(Object sender,EventArgs e){
        // code in response to the server click event.
  }
  ```

The Framework's Basic Life cycle

In JSP the event can also be fired using JavaScript: `gx_postBack("<SERVER_FUNC>");`

- **gx_preRender (JSP) / PreRender event (.NET):**

  Used to add logic after any post back event for example: refilling the page.

- **Controls/Tags rendering:**

  The dynamic tags are rendered with runtime content and attributes manipulation performed by the tagsAccessor in the previous stages of the life cycle.

- **gx_onUnload(JSP)/OnUnload (.NET):**

  Occurs when the page ends. Used to release any relevant resources.

- **gx_onError (JSP)/Error event (.NET):**

  Used for capturing errors. Any known thrown or runtime error is captured and can be analyzed by the developer.