



Soft Aspects

RIAs & Beyond



New light of Rich Internet Applications

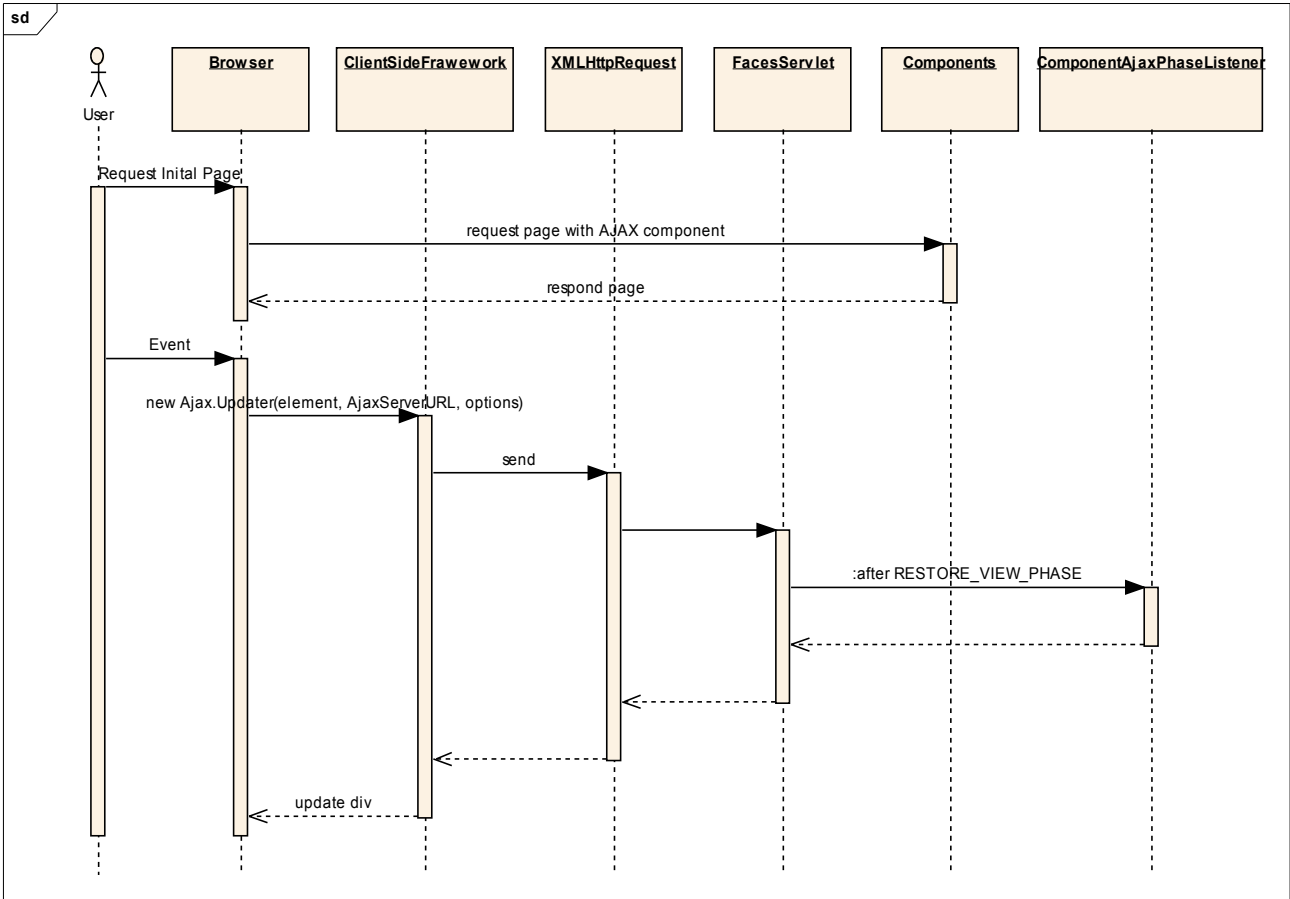
WGF AJAX Support Overview

Table of Contents

| | |
|---------------------------------|---|
| 1. WGF AJAX Implementation..... | 3 |
| 2. Server Side Processing..... | 4 |
| 3. Client Side Processing..... | 5 |

1. WGF AJAX Implementation

Sequence diagram of the AJAX request in the WGF implementation

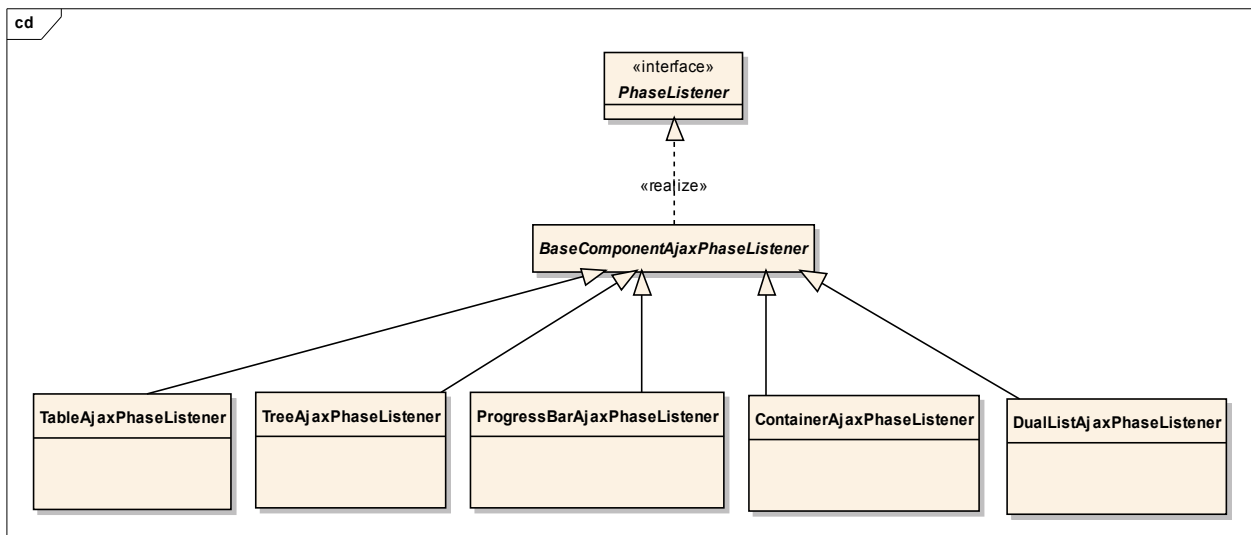


2. Server Side Processing

The server-side life-cycle is based on standard JSF 1.2 PhaseListener approach. There are components' AjaxPhase listeners implementations of JSF PhaseListener interface. To enable AJAX support for a particular WGF AJAX component the below configuration should be added to faces-config.xml file.

```
<lifecycle>
  <phaselistener>
    com.softaspects.faces.galileo.renderers.flowchart.TableAjaxPhaseListener
  </phase-listener>
</lifecycle>
```

Structural diagram of the WGF AJAX implementation classes



3. Client Side Processing

WGF AJAX client-side framework is based on open-source javascript framework Prototype (<http://prototype.conio.net/>).

Ajax.Updater instance is created to send Ajax requests to server and update dynamically added element with server-side data.

Below is the code snippet of AJAX client-side processing that is used in the Table component. The similar approach is used in other WGF components with AJAX support.

table.js

..
..

```
function table_refresh() {

    if (this.refreshStart)
        return;

    $(this.getId() + "_Action").value += "|onTableRefresh";

    this.refreshStart = true;
    if (this.ajaxRefreshEnabled) {
        // lazy load
        var charSetString = "";

        // Detecting current character set
        if (document.characterSet) {
            charSetString = '&charSet=' + document.characterSet;
        } else if (window.document.charset) {
            charSetString = '&charSet=' + window.document.charset;
        }

        this.lazyContentRequestUrl = getParentByTag($(this.getId()), "FORM").action;
        // Preparing server-side URL to be asked for content

        var rmodelId = this.getId() + "_rmodel";
        var rmodelValue = $(rmodelId).value;

        var actionId = this.getId() + "_Action";
        var actionValue = $(actionId).value;

        var actionDataId = this.getId() + "_ActionData";
        var actionDataValue = $(actionDataId).value;

        var cdeleteId = this.getId() + "_cdelete";
        var cdeleteValue = $(cdeleteId).value;
    }
}
```

```

var pageId = this.getId() + "_page";
var pageValue = $(pageId).value;

var cselectId = this.getId() + "_cselect";
var cselectValue = $(cselectId).value;

var changeId = this.getId() + "_change";
var changeValue = $(changeId).value;

var cmodelId = this.getId() + "_cmodel";
var cmodelValue = $(cmodelId).value;

var rselectId = this.getId() + "_rselect";
var rselectValue = $(rselectId).value;

var rdeleteId = this.getId() + "_rdelete";
var rdeleteValue = $(rdeleteId).value;

var sortId = this.getId() + "_sort";
var sortValue = $(sortId).value;

var ctreemodelId = this.getId() + "_ctreemodel";
var ctreemodelValue = null;
if ($(ctreemodelId))
    ctreemodelValue = $(ctreemodelId).value;

var lazyLoadServerURL = this.lazyContentRequestUrl +
    "?table-lazy-content-request=true"

    + "&" + rmodelId + "=" + rmodelValue
    + "&" + cdeleteId + "=" + cdeleteValue
    + "&" + pageId + "=" + pageValue
    + "&" + cselectId + "=" + cselectValue
    + "&" + changeId + "=" + changeValue
    + "&" + cmodelId + "=" + cmodelValue
    + "&" + rselectId + "=" + rselectValue
    + "&" + rdeleteId + "=" + rdeleteValue
    + "&" + sortId + "=" + sortValue
    + "&" + actionId + "=" + actionValue
    + "&" + actionDataId + "=" + actionDataValue
    + "&tableId=" + this.getId()
    + charSetString;

// Setting event handlers
if (this.userFunctionOnLoading) {
    try {
        options.onLoading = eval(this.userFunctionOnLoading);
    } catch (e) {
    }
}

if (this.userFunctionOnComplete) {
    try {
        options.onComplete = eval(this.userFunctionOnComplete);
    } catch (e) {
    }
}

```

```
    }  
  }  
  
  // 2. id top-element  
  var element = $(this.getId() + "_wrapper");  
  // Initiating request to the server and updating the element  
  // Finding the element to hold new content  
  
  var request =  
    new Ajax.Updater(element, lazyLoadServerURL, options);  
  
  }  
  
  this.refreshStart = false;  
  this.headerCellIdValue = null;  
} else {  
  var form = getParentByTag($(this.getId()), "FORM");  
  if (form)  
    form.submit();  
}  
}
```