GJax Developers Guide

From Intranet

This guide explains how to develop applications using gJax. It assumes knowledge of the following core skills:

- HTML and dynamic html (DHTML)
- JavaScript
- XSL
- M

It also assumes that you have an installed and working copy of gJax.

Table of contents

Introduction

A basic philosophy of gJax is that you can use as much or as little of it's capabilities as you need. The functionality provided by gJax is divided into a number of levels. Each level builds on the previous level by providing additional capabilities. In most cases these additional capabilities provide a trade-off between speed of development and flexibility. You can develop faster at higher levels (and in many cases create more maintainable sources), but you are more constrained about what you can do.

Another key philosopy of gJax is the ability to use whatever level of functionality you need to do a particular job. If you have a form that is developed at level x, but need to add some functionality to it that can only be done by writing raw html at level 1, then you can drop down to the level 1 style just for the bit of the form that demands it.

The following levels exist:

- **Level 0**. Anonymous access. This provides a session layer with unauthenticated access to hand crafted web-pages. Only pages that are explicitly defined as having ANONYMOUS access can be run in this way.
- **Level 1**. Authenticated access. A session is authenticated by a choice of authentication method (eg IIS integrated authentication, a standard pop-up login dialog, your own login dialog, some other mechanism such as recognising

an IP address or cookie). An authenticated session can access any page subject to the access rights of that user.

- **Level 2**. XML data and XSL Stylesheets. Level 2 introduces XSL stylesheets which separate the data from the htlm markup. This provides provides two major benefits:
  - Data is automatically escaped, preventing cross-site scripting (XSS) vulnerabilities.
  - The html markup is served in the XSL file which can be cached on the browser. Subsequently, the only bandwidth use is for XML data which is just the dynamic parts of the page. This provides better performance for frequently used pages that have a high html to data ratio.
  - Maintainablity is improved by separating the static html from they dynamic parts of the application. The html can be maintained by any html aware editor. The server side just needs to be able to generate an xml dataset (and this doesn't have to be M! A wrapper routine could be written that fetched the xml dataset from an Oracle or SQL Server database if that was where the data is stored).
- **Level 3**. ServerLink. This provides a mechanism for communicating (asychronously) with the server without the need to perform a page refresh. Most messages are exchanged using level 2 (XML/XSL) but level 1 or level 0 messages can be processed (using raw html invoking anonymous or authenticated server side methods). The message responses are typically processed using dynamic HTLM techniques (DHTML).
- **Level 4**. XForm Controls. This provides a library of XForms compliant controls that can be included on any web-page that requires user interaction. The following controls are currently supported:
  - Caption control (gjs:caption)
  - Input (gjs:input)
  - Date and calendar control (gjs:calendar)
  - Text area (gjs:textarea)
  - Button (gjs:trigger)
  - Checkbox (gjs:checkbox)
  - Radio button (gjs:radio)
  - Combo box (gjs:select1)
  - Select multi (gjs:select)
  - List box (gjs:list1)
  - Grid for repeating fields (gjs:grid)
    - Row (gjs:gridRow)
    - Column (gjs:gridColumn)
    - Cell (gjs:gridCell)
  - Date (output), sensitive to user's locale (requires authenticated session)
  - Date/Time (output), sensitive to user's locale (requires authenticated session)
  - Tree control A multi-level dynamic tree control
    - Generates tree markup (gjs:treeMore)
    - Generates tree markup for column headings (gjs:treeMoresMainHeading)
    - Generate sortable column headings (gjs:treeColumnHeading)
  - Sortable columns. A control that enables output to displayed in sortable columns
    - Define column width (gjs:column)
    - Column heading (gjs:columnHeading)
  - Menu control. Dynamic multi-level menus, both in a tool-bar and as right-click context sensitive popup menus. Automatic sensitivity to user access rights.

These controls are designed to emulate the Windows standard UI and inherit font styles, sizes and colour schemes from the user's individual desktop settings and preferences. The css stylesheets can be overridden (?) if alternative colour schemes or fonts are desired.

- **Level 5**. Form Control. The form control provides a forms manager that provides a stardardised tabular layout for a form. It also includes support for tabbed panes and standard buttons for OK, Cancel, Save, Print, Delete, Close and Help.
  - Form control A simple form layout manager
    - Form control (gjs:form)
    - Information Caption (gjs:information)
    - Tab Strip (gjs:tabStrip)
    - Tab Strip Element (gjs:tabStripTab)
    - Frames (gjs:frame)

    - Column Widths (gjs:columnWidths)
    - Set Column Width (gjs:column)
    - Fields (gjs:fields)

- **Level 6**. Main Screen Layout Manager. This provides a standard layout for the main screen of an application. It comprises of a menu bar, a tool bar, a folder panel on the left hand side and a message panel at the bottom. It also senses whether a user is has been authenticated using integrated authentication and provides a login dialog if they are not.

Architecture

gJax is a framework for developing rich web applications using AJAX principles. It incorporates a good security and authentication model and uses best practice techniques to enable robust, scalable and maintainable applications to be developed quickly and economically with the potential for a long lifespan.

Security and Authentication

Best Practices

Robust

Scalable

Maintainable

Rapid Development

Rapid application development is facilitated through the emphasis on component and template based techniques that encourages reuse. A library of rich and well crafted UI components provides a head start in contructing new applications.

Economical Development

Long Lifespan

All of the technologies used by gJax have been carefully chosen with a view to their longevity in the market place. Where possible open and accepted standards are used and proprietary and single-source solutions avoided.

- Operating System. gJax is largely independent of any operating system and is currently capable of being deployed on Linux, OS-X, various Unix platforms, Windows and OpenVMS.
- MySQL. The MySQL gateway provides support for the current industry leading open source DBMS. The architecture does not constrain or prohibity gateways to other DBMSs and database products. SQL is an ANSI standard.
- XML. The server acts as an XML data source. XML is non-proprietary and as the foundation for XHTML is likely to have a common use lifespan of at least several decades.
- XSL. Used to transform XML to HTML on the client. XSL is a W3C standard and is fully supported in both IE and Firefox. It is hard to tell at this time whether this component will become estabished enough to have a significantly long lifespan, however support seems assured in the medium term and there is likely to be a migration path to any technology that supersedes it.
- XHTML. The current best practice standard (in conjunction with CSS and JavaScript) for rendering browser content. It is likely to have a common use lifespan of many decades. XHTML is a W3C standard.
- M. M is used for the application server middleware. Two separate implementations of M are supported although other implementations exist that should be able to run gJax with little difficulty. GT.M is an open source implementation of M for Linux. Cache is a proprietary implementation of M for Windows. Both GT.M and Cache are available on various Unix platforms (Solaris, AIX, HPUX, etc) and OpenVMS subject to proprietary licenses.

M is an ANSI and ISO standard.
- JavaScript. JavaScript is the only non-proprietary browser scripting language that exists on all leading web-browsers. JavaScript is an ECMA standard.

Hello World

This is an example of how write a Hello World page using level 0 functionality in three simple steps:

Write a server-side method

Using your favourite Cache/M editor create a routine called `helloWorld` consisting of the following lines:

```
helloWorld()    ; Public ; Hello world example ; VCm.Client.cls?page=HelloWorld
  w "Hello world"
  q
```

Add a helloWorld function

Open the Setup/Functions folder. From the menu select File/New and then enter the following:

```
Function: helloWorld
Description: Hello world example
Server-side method: ^helloWorld
Access codes: ANONYMOUS
```

Run it

Now point a web browser at http://localhost/gJax/VCm.Client.cls?page=helloWorld

If you don't get a page containing the text *Hello World* then one of the following things may be wrong:

1. Check the root part of the url *http://localhost/gJax/VCm.Client.cls* is correct for your installation of gJax. The server name or path might need to be different.
2. Check that *VCm.Client.cls* is typed correctly. It is case sensitive.
3. Check that the page name *page=helloWorld* is typed correctly. Particularly, the h is lower case.
4. Check that the function helloWorld was defined and saved correctly. Particularly, the h lower case and should match the page argument on the browser url.
5. Check that the access codes is set to ANONYMOUS and is spelt correctly. Attempting to access a page that is not set up as ANONYMOUS will result in an error, not a message saying access denied. (This is an important and deliberate security feature - hackers are not able to distingush between pages that do not exist and those that do exist but to which they do not have access).

Level 6 - Workspace

- gJax Workspace Workspace container

Level 5 - Form Control

- gJax gjs:calendar Calendar control

Level 4 - XForm Controls

- gJax gjs:calendar Calendar control
- gJax gjs:treeColumnHeading Sortable and resizable column headings

Level 3 - ServerLink

- This page was last modified 19:29, 27 Jan 2006.