OTHER SERVER-SIDE WEB SCRIPTING TECHNOLOGIES

1. Active Server Pages (ASP)

Definition

ASP stands for ‘active server pages’ and it was first developed in the mid 90’s (version 1.0) to create web content which could change upon interacting with the user, as its name suggests. When an ASP page is requested within HTML code it is sent from the server which hosts the site (a.k.a. server-side scripting); for instance the server would store information on a particular user by way of ‘cookies’, and ensure that content is specific to that particular user. This is useful when displaying current time or weekday for instance (i.e. setting your own time-zone), or when accessing a database which has stored exclusive information.

Early Origins of ASP

With the advent of the Internet Information Server 3.0 (IIS 3.0), or a series of applications for hosting windows based web servers, there came the need to generate faster dynamic content. This was achieved with the development of ASP, a framework which included the Virtual Basic programming language as its core language and various others.

The very first version of ASP, version 1.0, was first developed in December 1996 by Microsoft. Before its full implementation, developers used a combination of programming languages (such as Perl and C++) or scripts in order to create applications which could then be loaded to create dynamic web sites. However, this proved to work very slowly and cause a strain on the servers. Therefore there was a need for applications to be executed directly into the server without loading external programs, and this is where ASP proved to be handy. ASP's strength lies in using ActiveX: this is technology used to build single components which are then executed directly into a website.

Growing into ASP.NET

With the development of the .NET framework for Windows, a more modern programming environment with easier linkable libraries and optimised code, the original version of ASP grew into ASP.NET. Its new version came as IIS 4.0 was developed in 1997, a year or so after IIS 3.0.

The original developers for the next version of ASP were Mark Anders and Scott Guthrie (source http://en.wikipedia.org/wiki/ASP.NET), who developed a prototype called XSP (cross-site printing) in Christmas of 1997, which was then substituted to ASP. The new project was codenamed ‘cool’ by the Microsoft team and used Common Language Runtime, which resulted in being simpler and ‘cleaner’ than the previous ActiveX and COM based versions. CLR is still an integral part of the .NET framework and allows for cross-compatibility, by being able to import and run applications to be built in the many supported programming languages. This allows the web developer to program server applications in different languages depending on

what is considered most effective (i.e. Visual Basic for a graphic user interface), and integrate all the components with ease within the server software.

Some Simple Uses

Aside from the above-mentioned uses for your site, there are many more things a developer can do with ASP.NET. Some include:

The use of 'Event Handlers'

With static web it becomes impossible to write code which executes when something specific occurs. Using CGI scripts this is possible but is lengthy and slow to build. However with event handlers in ASP it can work flawlessly, such as the 'Page_Load' event, which is triggered once a page is loaded, or 'Unload' which works the opposite way or when the page is closed.

Complete compatibility (the 'runat' command)

By inserting this command, the ASP.NET framework ensures complete compatibility with many programming languages and makes pages with older versions of ASP also compatible.

It would be too long to list every single advantage. There are some good free tutorials for using ASP.NET, such as www.asp.net/learn by Microsoft. The library is also freely available to anyone who wishes to use it.

In conclusion, by using ASP.NET you are able to integrate the essential aspects of a business website seamlessly and with simple code.

2. Common Gateway Interface (CGI)

It is a strategy utilized by web server to run outer projects, frequently to create web content powerfully. At whatever point a site page inquiries a database, or a client presents a frame, a CGI content is typically called upon to do work. Characterizes standard path for web servers to run CGI contents and for those projects to send their outcomes back to the server. The activity of the CGI content is to peruse data that the program has sent through server and to produce some type of substantial reaction generally unmistakable substance. when it has finished its undertaking, the CGI content completes and exits.

CGI PROCESSING:

At the point when a HTTP server gets a demand for a CGI content, the server conveys to the content the subtle elements of the demand. The HTTP server and a CGI content convey in four noteworthy ways:

1. **Environment Variables:** The HTTP server utilizes condition factors to pass data about the demand to the CGI content. Contingent upon the kind of demand, nature factors could possibly contain all the data required by the content to work legitimately.

2. **Command Line:** The charge line is utilized for isindex questions. For the most part, isindex inquiries ought not be utilized; since the summon line is utilized straightforwardly, they show numerous potential security dangers.
3. **Standard Input:** For HTTP POST or PUT inquiries, the HTTP server imparts data to the CGI content through standard info. The measure of data kept in touch with standard information is put away in the CONTENT_LENGTH condition variable.
4. **Standard Output:** A content profits its yield for standard yield. The yield can be a report produced by the content, or guidelines to the server for recovering the coveted yield.

USING CGI SCRIPTS:

- A web server enables its proprietor to design which URLs should be taken care of by which CGI contents. This is typically done by denoting a catalog inside the record gathering as containing CGI contents — its name is frequently cgi-container.

For instance,/usr/local/apache/htdocs/cgi-bin could be assigned as a CGI registry on the web server. At the point when a Web program asks for a URL that focuses to a document inside the CGI directory

(e.g., http://example.com/cgi-bin/printenv.pl/with/extra/path?and=a&query=string), at that point, rather than basically sending that record (/usr/nearby/apache/htdocs/cgi-bin/printenv.pl) to the Web program, the HTTP server runs the predefined content and passes the yield of the content to the Web program.

3. Ruby on Rails

- Server-side web application framework written in Ruby
- Model-view-controller (MVC) framework
- Providing default structures for a database, we pages and services
- Encourages and facilitates the use of web standards such:
    a. JSON or XML for data transfer
    b. HTML, CSS and JavaScript for display and user interfacing
- Emphasizes the use of other well-known software engineering patterns and paradigms

History:

- David Heinemeier Hansson removed Ruby on Rails from his work on the project management tool Basecamp at the web application

JULY 2004 and FEBRUARY 2005

- He releases Rails as an open source but did not share commit rights to the project until the year of 2005 in February.

AUGUST 2006 and OCTOBER 2007

- Ruby on Rails reached a milestone when Apple announced that it would ship this framework with Mac OS X v10.5 “Leopard”, which was in October 2007.

MARCH 15, 2009

- They've released a new version of Rails which is Rails version 2.3 with major new developments in templates, engines, Rack and nested model forms.

DECEMBER 23, 2008

- Merb is another web application framework and announced it would work with the Merb project to bring the best ideas of it into Rails 3 ending the unnecessary duplication across both communities and this framework was merged with Rails.

Rails 3.1: Released on August 31, 2011, featuring Reversible Database Migrations, Asset Pipeline, Streaming, jQuery as default JavaScript library and newly introduced CoffeeScript and Sass into the stack.

Rails 3.2: Released on January 20, 2012 with a faster development mode and routing engine (also known as Journey engine), Automatic Query Explain and Tagged Logging. Rails 3.2.x is the last version that supports Ruby 1.8.7. Rails 3.2.12 supports Ruby 2.0.

Rails 4.0: Released on June 25, 2013, introducing Russian Doll Caching, Turbolinks, Live Streaming as well as making Active Resource, Active Record Observer and other components optional by splitting them as gems.

Rails 4.1: Released on April 8, 2014, introducing Spring, Variants, Enums, Mailer previews, and secrets.yml.

Rails 4.2: Released on December 19, 2014, introducing Active Job, asynchronous emails, Adequate Record, Web Console, and foreign keys.

Rails 5.0: Released on June 30, 2016, introducing Action Cable, API mode, and Turbolinks 5.

Rails 5.0.0.1: Released on August 10, 2016, with Exclusive use of rails CLI over Rake and supports Ruby 2.2.2+ versions

Rails 5.1: Released on April 27, 2017, introducing JavaScript integration changes (management of JavaScript dependencies from NPM via Yarn, optional compilation of JavaScript using Webpack, and a rewrite of Rails UJS to use vanilla JavaScript instead of depending on jQuery), system tests using Capybara, encrypted secrets, parameterized mailers, direct & resolved routes, and a unified form_with helper replacing the form_tag/form_for helpers.