

[HOME](#) » [SEGUE BLOG](#) » CLIENT-SIDE VS. SERVER-SIDE CODE: WHAT'S THE DIFFERENCE?

Our Staff Experience and Expertise

Client-Side vs. Server-Side Code: What's the Difference?

by **Bill Fote**

May 1st, 2013

Server-Side | Client-Side | Web Development | Ajax | Web Apps | Web Applications

In web applications, there is the client and the server. The "client" is a web browser, like [Internet Explorer](#), [Google Chrome](#), [Firefox](#), etc. The "server" is a web application server at a remote location that will process web requests and send pages to the client. Web applications can contain code that is processed on the client's browser or on the web server. However, web applications have a disconnected architecture, which means that there is never a live, constant connection between the page displayed in the client's browser and a web or database server. The majority of the processing will be done at the server and not on the client's internet browser. When a database needs to be accessed on a server, the web application will post the page back to the web server and server-side code will process the request.

In order to fully understand the web application's architecture, we must understand [postback](#) and the [page life cycle](#), which I have discussed more extensively in previous articles.

Server-Side Code

Connect with Us



Get Weekly Blog Updates!

email address: →

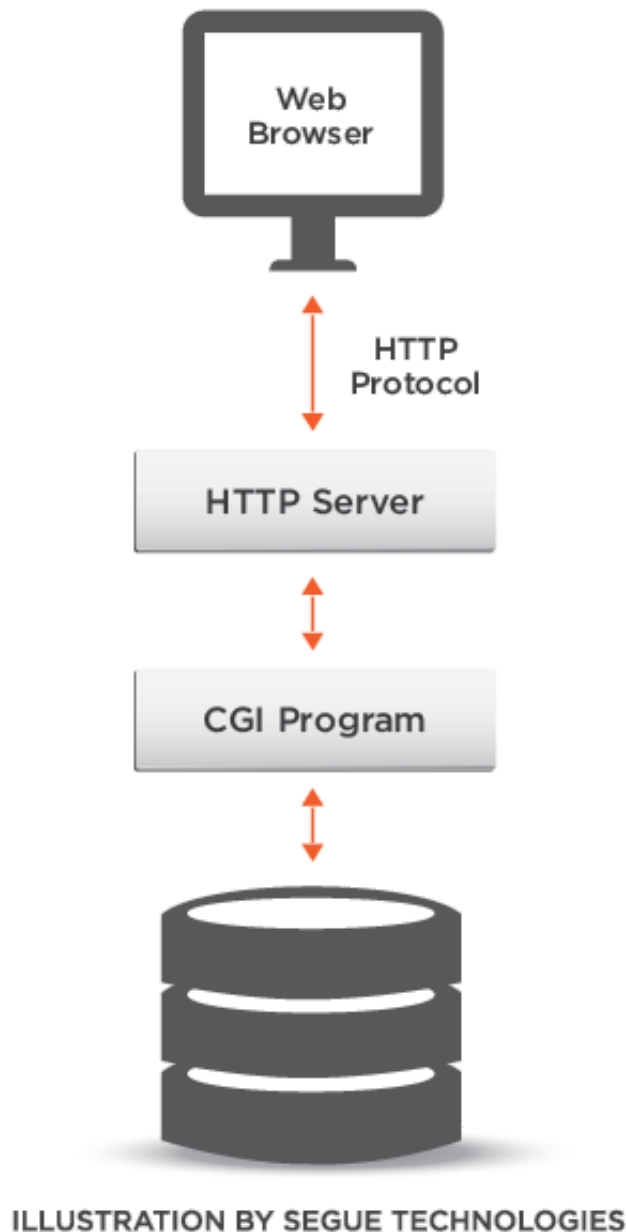


Click Here

for "Hot Tip" emails on Web, Data, & Mobile development

Popular Blog Topics

There are several server-side



technologies that can be used when developing web applications. The most popular is Microsoft's [ASP.NET](#). In ASP.NET, server-side code uses the .NET Framework and is written in languages like C# and VB.NET. Server-side processing is used to interact with permanent storage like databases or files. The server will also render pages to the client and process user input. Server-side processing happens when a page is first requested and when pages are posted back to the server. Examples of server-side processing are user validation, saving and retrieving data, and navigating to other pages.

Mobile Application Development
Mobile
Health IT
Quality Control
Quality Assurance
Software Development
Android

Popular Articles

Waterfall vs. Agile: Which is the Right Development Methodology for Your Project?

8 Benefits of Agile Software Development

.NET vs Java: How to Make Your Pick

[view more](#)

Recent Articles

ColdFusion: 20 Years and Beyond

ColdFusion 11: A Review

What is Assured Compliance Assessment Solution (ACAS)?

[view more](#)

Related Content

Five Guys Website Launch

Segue Awarded

The disadvantage of server-side processing is the page postback: it can introduce processing overhead that can decrease performance and force the user to wait for the page to be processed and recreated. Once the page is posted back to the server, the client must wait for the server to process the request and send the page back to the client.

Client-Side Code

The benefits of client-side processing in an ASP.NET web application are programming languages like C# and VB.NET along with the .NET Framework. Languages like C# and VB.NET sit on top of the .NET framework and have all the benefits of object oriented architectures like inheritance, implementing interfaces and polymorphism.

In contrast to server-side code, client-side scripts are embedded on the client's web page and processed on the client's internet browser. Client-side scripts are written in some type of scripting language like JavaScript and interact directly with the page's HTML elements like text boxes, buttons, list-boxes and tables. HTML and CSS (cascading style sheets) are also used in the client. In order for client-side code to work, the client's internet browser must support these languages.

There are many advantages to client-side scripting including faster response times, a more interactive application, and less overhead on the web server. Client-side code is ideal for when the page elements need to be changed without the need to contact the database. A good example would be to dynamically show and hide elements based on user inputs. One of the most common examples is input validation and Microsoft's Visual Studio includes a set of client-side validation controls.

However, disadvantages of client-side scripting are that scripting languages require more time and effort, while the

USAF Contract for
International
Affairs Link

Segue Designs
Custom
Responsive
Website for the
Foundation for Fair
Contracting

Segue Designs
Responsive
Microsite for
OSEHRA

About Segue

client's browser must support that scripting language.

[SERVICES](#)[OUR WORK](#)Search 

The general rule is to use server-side processing and page postbacks when the client needs to interact with server-side objects like databases, files, etc. However, the concept of [Ajax](#) has changed the rules quite a bit. Ajax is the concept of the client calling the server directly to interact with server objects like a database, without a postback involved.

Ajax is a concept that involves a group of existing technologies such as server-side data, web services and client-side scripting. The client-side scripts will call a web service and the web service processes the database request. The request could be the retrieve and/or save data. Ajax calls are asynchronous, meaning that once the client makes an Ajax call to the web service, the client is not locked and waiting for a response. The web service will send a response back to the client when their task has completed. The client will intercept the response and process the response accordingly.

A significant breakthrough in client-side scripting is [jQuery](#). To [quote](#), jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Starting with Visual Studio 2005, Microsoft has offered their [Ajax Control Toolkit](#). This a set of ASP.NET controls that have plenty of built-in client-side processing. With Visual Studio 2008, they've offered AJAX-enabled WCF Services. These web services are streamlined for asynchronous Ajax callbacks and

require little client-side scripting.

In conclusion, the amount of the client-side scripting used in web applications will continue to increase as its power, flexibility and simplicity continue to increase.



Share This Article

13

1 Comment

by **Jenni** on Thu Apr 3, 2014 at 9:49 am

Thank you for this article! I have to write a discussion board post about the differences between server side and client side scripting and this article gave a nice break down of the two.

- [reply](#)

YOUR NAME

POST A COMMENT *

SUBMIT

SERVICES

Web
Data
Mobile
Supporting Services

OUR WORK

Government
Commercial
Non-Profit

ABOUT US

Management Team
Partners
Community Involvement
Awards and Recognition
Careers
News

CONTACT US

2300 Wilson Boulevard
Suite 420
Arlington, VA 22201
Toll Free: 888.549.8033
Tel: 703.549.8033
Fax: 703.549.8233

CONNECT WITH US



Washington, DC Atlanta, GA Annapolis, MD
Alexandria, VA Quantico, VA Baltimore, MD
Arlington, VA Richmond, VA Dayton, OH

*Building innovative Web, Data, and Mobile
applications
to help your organization succeed*

