

# THE DEVELOPMENT OF WEB BASED APPLICATIONS CREATING USER SESSIONS WITH ISAPI

Erdem Uçar<sup>1</sup>  
[erdemu@trakya.edu.tr](mailto:erdemu@trakya.edu.tr)

Erdinç Uzun<sup>2</sup>  
[erdinc@trakya.edu.tr](mailto:erdinc@trakya.edu.tr)

Veysel Karslı<sup>3</sup>  
[veyselkarsli@trakya.edu.tr](mailto:veyselkarsli@trakya.edu.tr)

Trakya Üniversitesi, Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü, 22030, Edirne

Key Words: ISAPI, Session, Cookie

## 1. INTRODUCTION

Internet server programs were first developed for Unix systems. These programs were standard until Microsoft introduced the IIS( Internet Information Server ). The base part of HTTP the CGI standard was developed to allow communication between the browser and server-side separately running programs or scripts.

Microsoft designed IIS without changing the HTTP/CGI architecture, allowing the browser to load and run a server DLL. DLL is a part of IIS process and it's faster than the code needed to load separately running programs. In addition writing an ISAPI DLL in C++ is much easier than writing a code in PERL.

ISAPI server extensions provide an alternative for CGI applications used in internet servers. ISAs( Internet Server Applications ) like HTTP server run in the same address part and can access all available sources of HTTP server. ISAs cost less than CGI applications because they don't need additional processing and they do not execute long processes within the processing time-limit. If another process needs memory space then both extensions and filter DLLs are unloaded from the memory.

An ISA is called by a browser application. An internet server calls an ISA in the same way it calls a CGI application. E.g., a server call for a CGI is as follows:

<http://sample/example.exe?Param1&Param2>

An ISA executing the same function can be called as follows:

<http://samole/example.dll?Param1&Param2>

Every CGI request initiates a process. Every different request runs in its own executable file which is loaded and unloaded in every request. That is why it is costly.

## 2. SESSION

One of the rules for a successful web application development is to protect user data while the user is surfing the applications. HTTP is an independent protocol because every HTTP request coming from a page behaves as an independent request. If the requests come before available requests then the server won't hold any information of the preceding requests. Server's inability to memorize preceding requests makes difficult writing on-line applications which follow the catalog products the user chooses passing through different pages of the catalog. In order to overcome this difficulty a session should be developed.

Variables hidden in session do not leave the server memory while the user surfs the application. In this way surfing the application doesn't cause the loss of information.

Cookies are related to sessions. Cookie is a sign passed by server to the user browser to identify the user. Next time when the same browser requests a page, it sends that cookie to the Web server. Every browser uses different cookies. But the same cookies operate while surfing the application in the same browser. When the request occurs the server generates an id number and passes it to the browser, and when the browser makes the next request, the server takes the id and identifies the browser.

## 3. APPLICATION

When we decided to develop a Web application using ISAPI extension DLLs we noticed that in Web-based programming languages like *asp*, *php* and *jsp* there wasn't any session object appropriate to our objectives.

So we had to create one ourselves. The created object allocates a memory space in server when a user requests to see his pages. Since the memory address location changes each time, it can be used as a session id. This session id is sent to the browser's cookie. While the user surfs the pages, the browser each time sends the session id which is inside the browser's cookie to the server, then the server checks it to make sure if there is a memory space assigned to it. This process makes users differ from each other. When the user wants to exit his page, server understands it and closes the memory location assigned to him. This way the user session ends up. If the user doesn't make any request for a period of time or he exits

the pages without performing the exit operation the server will not close the user session. In this case server's memory accumulates unnecessary information.

For this reason each session has its own time slice and if the user doesn't make any request within this period the user's memory space will be closed. This is realized by storing the session id and the last request time in a database. The database is checked every 60 sec. to check if session time is over. In case it is over, the memory location will be closed, information will be deleted from the database and the session will be closed. (Figure 1)

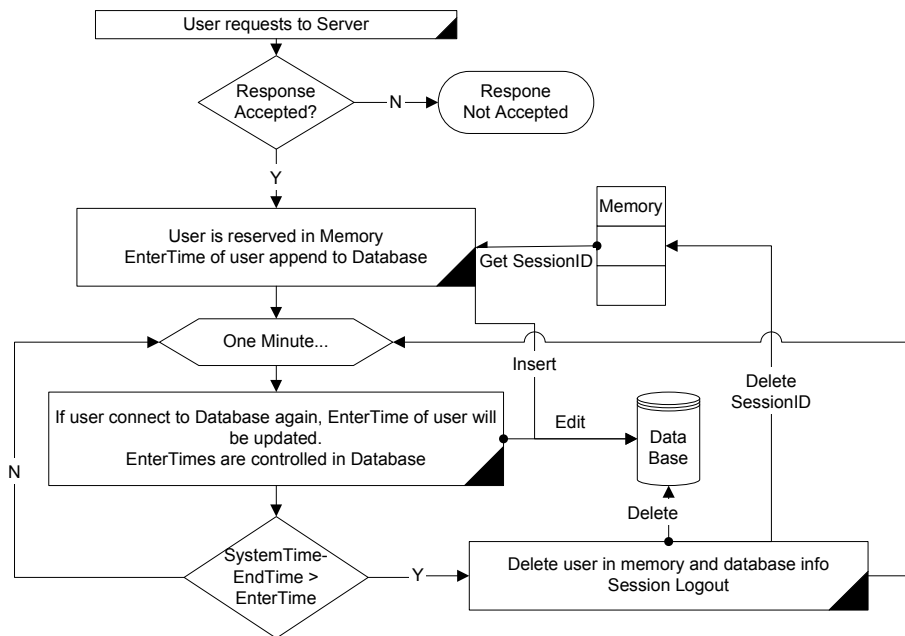


Figure 1 : User Session Process Algorithm

#### **4. STRENGTHS OF ISAPI**

The set of interfaces that make up ISAPI were designed to provide application developers with a powerful way to extend the functionality of IIS. A properly planned, developed, and used ISAPI—particularly ISAPI extensions—can outperform any other Web application technology available currently for IIS. From an ISAPI extension or filter, you have access to the whole array of Win32 API functions. For instance, you can create a customized worker-thread pool for your ISAPI extension to speed processing, using the Win32 thread functions to access the native thread functionality of Windows 2000.

#### **6. REFERENCES**

1. MSDN Library Visual Studio 6.0
2. Programming Microsoft Visual C++ by David J. Kruglinski 1998
3. Inside ISAPI Inside ISAPI - Kevin Clements, Chris Wuestefeld, Jeffrey Trent
4. Professional C++ Isapi - Michael Tracy

#### **5. CONCLUSION**

The application we developed can be used in many different areas.

1. The application we developed will allow the programmers using ISAPI applications creating their own session dependent on their objectives.
2. In addition the advantage of the DLL's performance can provide a wide scope of user session management.
3. With some additions, this application can be used as a Web Server;
4. Due to its performance it can be used for search engine development;
5. Embedding necessary encryption algorithms, the security of the application may be increased;