Java Servlets

Nancy McCracken
NPAC
Syracuse University

November 1999

Overview of Java Servlets

- A servlet is a Java program that is run by a Web server.
- Servlets follow a standard servlet API defining the interface between the server and the servlet, and are designed to work within a request/ response model. The servlet API is part of the standard extensions of JDK.
- Servlets can provide all the services of standard CGI, but are platform-independent and can have lifetimes spanning many transactions.
- More generally, servlets play the role of providing middle-tier services between clients and back-end applications.
- References:
  - http:/ / www.javasoft.com/ products/ servlet
  - "Fundamentals of Java Servlets", a tutorial from the MageLang Institute
  - Java Servlet Programming, Jason Hunter, O'Reilly, 1998.

Architectural Roles for Servlets

- Middle-tier process:
  - Connection management to multiple clients
  - Transaction management
  - Business Rule Enforcement
  - Supporting different types of clients, such as HTML and Java applet
- CGI scripts
- Proxy server for applets to connect to other hosts.
- Protocol support other protocols: SMTP, POP, FTP, . . .
- Some servers allow the <servlet> tag in HTML, and the server will call the servlet to insert text into the outgoing HTML stream (server side includes - SSI).
- Some servers support Java Server Pages (JSP) which allow servlet scripts to be embedded into HTML pages and executed when the page is requested.

Temporary and Permanent Servlets

- A temporary servlet is started when a request arrives and shut down after the response is generated.
- A permanent servlet is loaded when the server is started and lives until the server is shut down.
  - This is useful when startup costs are high, such as a servlet that establishes a connection to a database.
  - Also useful for permanent server-side service, such as an RMI server.
  - Provides faster response to client requests when this is crucial.
- Being temporary or permanent is part of the server configuration.

Using Servlets

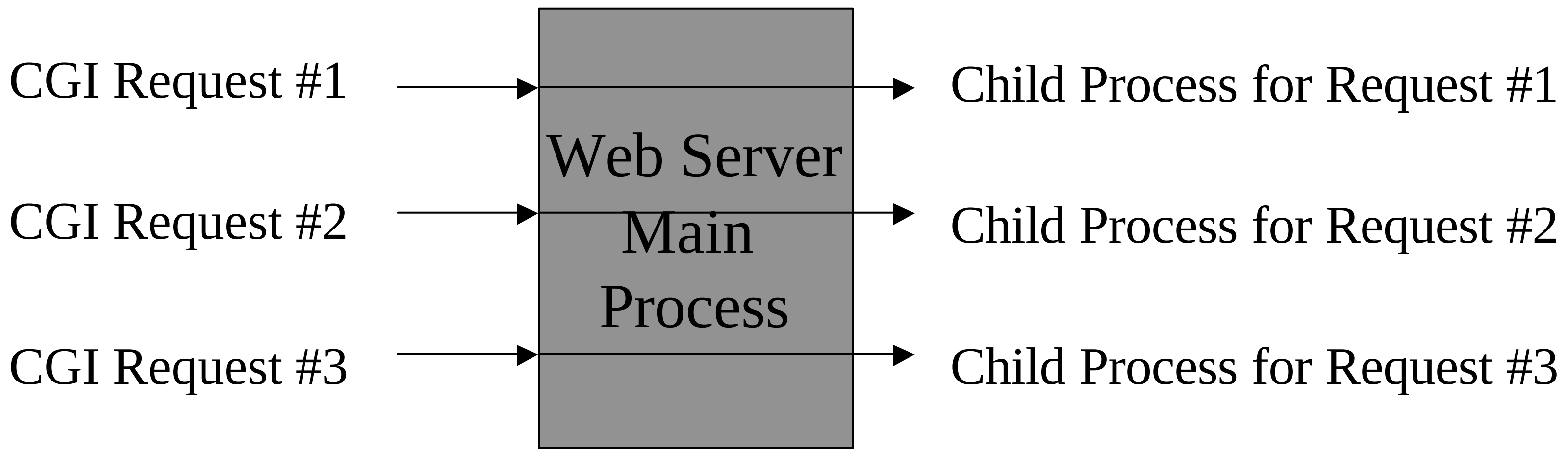
- Install Java Servlet Development Kit (JSDK). This works with both JDK1.1 and JDK1.2 platforms.
- Servlets can be tested with the utility program servletrunner.
- Servlet API is a standard extension to the JDK under javax:
  - Package javax.servlet
  - Package javax.servlet.http
- The servlet API provides support in 4 categories:
  - Servlet life cycle
  - Access to servlet context
  - Utility classes
  - HTTP-specific support classes

The Servlet Life Cycle

- On the web server host machine, servlets run under the same process as the web server.
- The web server is responsible for creating an instance of the servlet and invoking standard methods from the interface (in a similar way to a browser invoking methods of an applet). There are three main methods:
  - init ( )
  - service ( )
  - destroy ( )
- and two helper methods:
  - getServletConfig ( )
  - getServletInfo ( )

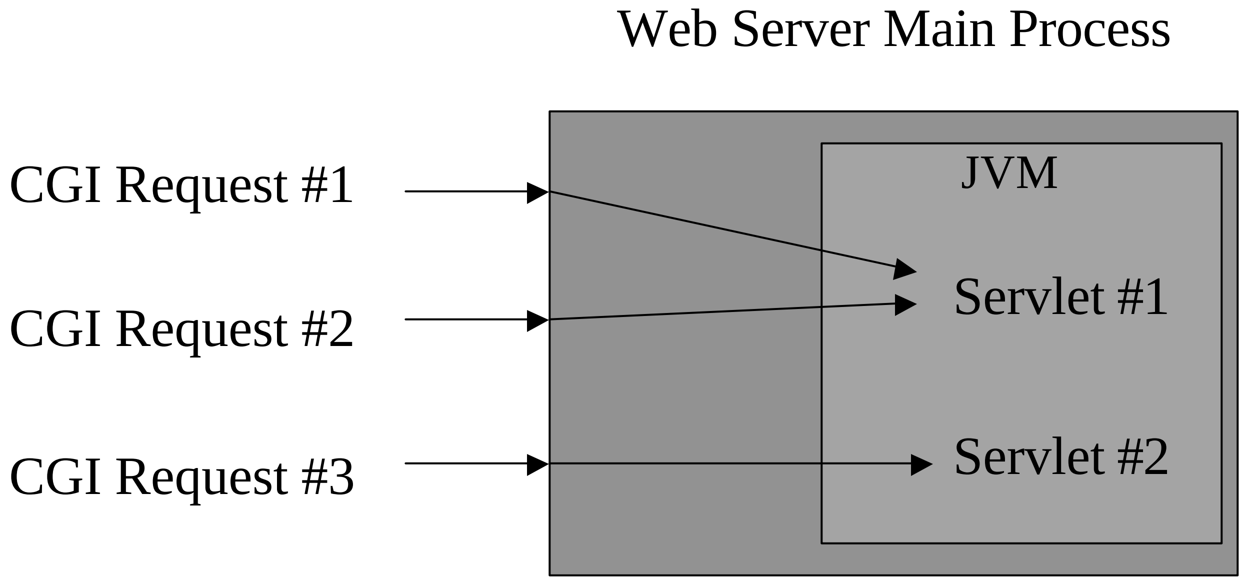
Standard CGI Architecture

- Standard web server with CGI spawns a process for each CGI request and terminates it when the request is completed. This takes significant time and server resources.


CGI Request #1
CGI Request #2
CGI Request #3
Web Server Main Process
Child Process for Request #1
Child Process for Request #2
Child Process for Request #3

Servlet Architecture

- In a Java servlet-based web server, one instance of the servlet object is run in the Java Virtual Machine of the web server. For each request, the web server spawns a thread that runs that instance.



The init method

- public void init (ServletConfig config)
- The init method is invoked when the servlet is first started. For permanent servlets, this is when the server is started.
- The init method is called only once and is guaranteed to finish before any calls to the service method.
- The ServletConfig object passed as the parameter has a method

      getServletContext ( )

  that returns a ServletContext with information about the servlet environment.

The service method

- public void service (ServletRequest req, ServletResponse res)
- Each request message from the client results in invoking the service method.
- There are two ways the client can send information:
  - parameter name/ value pairs
  - streams - the servlet can invoke getInputStream(), which returns a ServletInputStream.
- Similarly, the servlet can invoke getOutputStream () to generate the response. Other methods, such as setContentType can set information that the web server will use in generating the output to the client.
- More than one instance of the service method can be invoked at one time to respond to multiple requests.

The destroy method

- Called when the servlet in unloaded to clean up any open resources.
- Although the server normally waits until all service calls are terminated to invoke destroy, it may not be possible, and your destroy method should make sure that resources are not being used.

Example Servlet (from MageLang)

- This servlet returns a page to the web browser:

```
import java.io.*;
import javax.servlet.*;

public SampleServlet implements Servlet
{  private ServletConfig config;

    public void init (ServletConfig config) throws ServletException
      {  this.config = config;  }

    public void destroy ( )  {  }  // do nothing

    public ServletConfig getServletConfig ( )
      {  return config;  }

    public String getServletInfo()
       {  return “A simple servlet”; }
```

Service method of the Example

```
public void service (ServletRequest req, ServletResponse res)
                    throws ServletException, IOException
    {     res.setContentType ( “text/html” );
          PrintWriter out = res.getWriter ( );
          out.println ( “<html>”);
          out.println ( “<head>”);
          out.println ( “<title> A Sample Servlet </title>”);
          out.println ( “</head>”);
          out.println ( “<body>”);
          out.println ( “<h1> A Sample Servlet </h1>”);
          out.println ( “</body>”);
          out.println ( “</html>”);
     }
}
```

Servlet Initialization Information

- This information is passed to the servlet via the ServletConfig object parameter of the init method.
- Each web server has its own information tags defined.
- The getInitParameterNames ( ) returns an Enumeration of all the parameters available to the servlet on that server.
- A particular parameter is obtained by

  config.getInitParameter (“timezone”);

Servlet Context Information

- This information is available at any time through the ServerConfig object. (This is why you should save the ServerConfig parameter from the init method.):

  Servlet Context c = config.getServletContext ( )
- The ServletContext object has methods:
  - getAttribute
  - getMimeType
  - getRealPath
  - getServerInfo
  - getServlet (String name)
  - getServletNames
  - log (String message)

The Servlet Request parameter

- The servlet gets information from the ServletRequest parameter either by a Stream:
  - req.getInputStream (or req.getReader for ASCII chars)
- or by getting individual parameter names:
  - req.getParameter (“firstname”);
  - req.getParameter (“secondname”);
- There are additional methods for getting request information, such as:
  - getRemoteAddr
  - getProtocol
  - getServerName
  - . . .

Utility classes

- There are two exception classes in the servlet API.
  - javax.servlet.ServletException, the servlet should throw this exception to the server in the event of general failure.
  - javax.servlet.UnavailableException, servlets can report this at any time to the server. The servlet may write a log entry so that an administrator can take some action.

Generic Servlets

- This subclass of Servlet provides an init method that takes care of saving the ServletConfig parameter.
- It provides several methods that use the ServletConfig object to return information.
- It also provides a dummy destroy method.
- In general, the class Servlet and GenericServlet provide methods to deal with request/ response architectures in a protocol-independent way.

HTTPServlet

- This subclass of GenericServlet provides additional services based on the HTTP protocol.
- It separates requests into the different types of request methods.
  - GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT, OPTIONS
- The service method of HTTPLservlet checks the type of request and calls one of several special methods if they are present:
  - doGet, doHead, doDelete, doOptions, doPost, doTrace
  - Note that if you supply a doGet method, the service method can use if for doHead, doTrace and doOptions as well by extracting header information.
- doGet and doPost methods should read request data, set response headers and write response data.

Processing HTTP requests

- The doPost method is required to open an InputStream to get the parameters, however, it can call a method parsePostData, that returns a Hashtable with the names and values of the parameters.
- There are servlet methods of the form req.getXXX for every CGI environment variable XXX.
- To create a cookie on the browser, you create an object of type Cookie and use res.addCookie(c) to send it to the browser. Cookies are retrieved by using methods getCookies, and getName and getValue for individual cookies.
- There is also a special HPPTSession class that implicitly uses cookies to allow you to save session information as name/ value pairs in between requests.

Security Issues

- Most servers allow the server administrator to associate different levels of "trust" to servlets, usually depending on where they came from. The levels usually control the access to the file system and networking.
  - Note that a fully trusted servlet can even issue System.exit(), stopping the web server.
- Servers may also allow Access Control Lists to be created which can allow different users of groups of users access to some web pages (or not).

Threading Issues

- A web server may call the applet's service method for more than one request at a time by creating multiple threads.
  - You can disallow this by using javax.servlet.SingleThreadModel, which makes it possible to write simple servlets where only one instance of the service method is ever running at one time.
- If your service method uses outside resources, such as instance data from the servlet object, files, or databases, you must carefully consider what might happen under multiple calls to the service method, and use synchronization where appropriate.

Additional Resources

- Main servlet support page at Sun:
  http:/ / java.sun.com/ products/ servlet
- List of environments supporting servlets at
  http:/ / java.sun.com/ products/ servlet/ runners.html
- Information on the HTTP protocol at the World Wide Web Consortium’s web site
  http:/ / www.w3.org/ Protocols
- Servlet Central web magazine
  http:/ / www.servletcentral.com
- Jrun Magazine http:/ / www.jrunmag.com
- Books:
  - Java Servlets, Karl Moss, McGraw-Hill.
  - Java Servlet Programming, Jason Hunter and William Crawford, O’Reilly.