Java Servlets

William DeRaad & Russ Mehring

Table of Contents

Java Servlets

Introduction

(Servlets and Their History)

What is a Servlet?

- Middle layer between Web browser and other http clients (such as databases)
- Simply put, they are Java objects that respond to http requests
- Servlets must run inside a servlet container (Tomcat or Jetty)
- Servlets allow for extended capabilities of applications that are hosted by web servers by querying input from users or databases to build content dynamically (dynamic web pages need a support server like servlets)

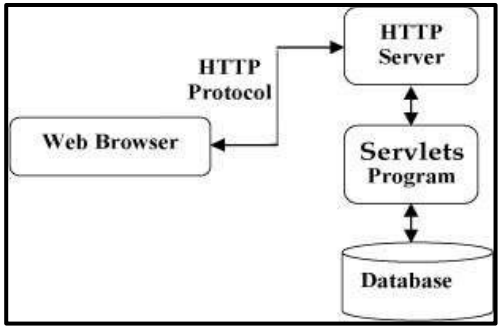


Figure A

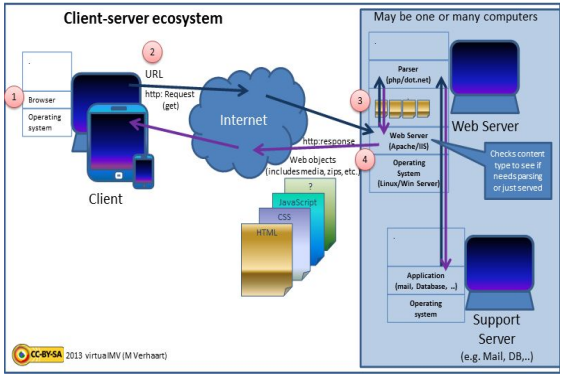
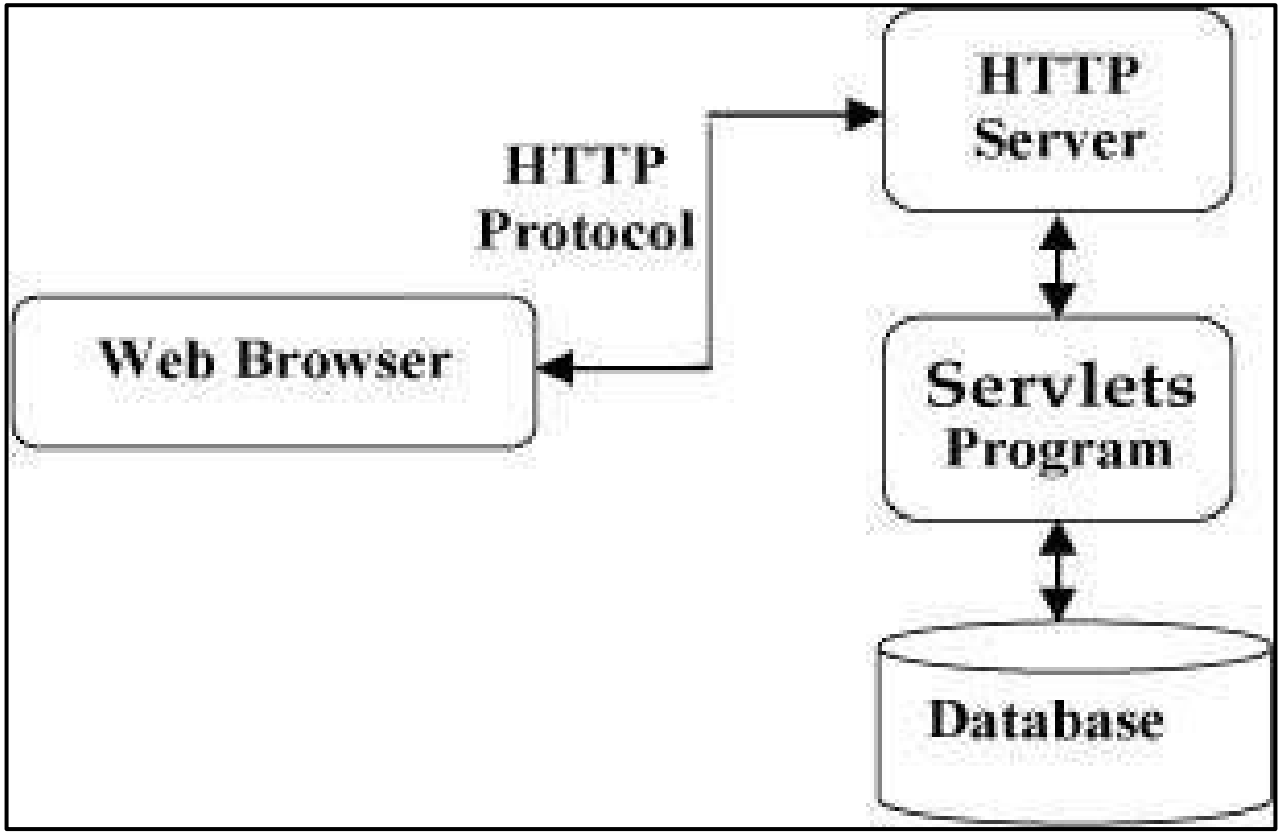


Figure F


HTTP
Server
HTTP
Protocol
Web Browser
Servlets
Program
Database

Why use Servlets?

Servlets offer a myriad of benefits, to name a few:

- Servlets are FAST--outperforming most early similar technology
- Servlets are highly scalable--a Servlet is part of a Java web app. and a servlet container can contain multiple servlets to handle requests
- Servlets offer Java benefits: Platform independence, security, object orientedness, etc.
- Servlets execute within address space of Web server so no need to keep creating them!

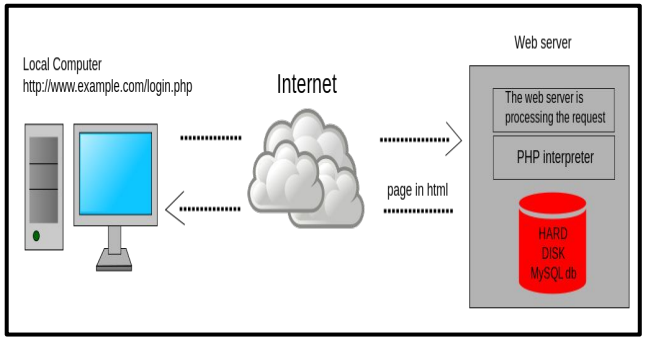


Figure B:
Dynamically built web page overview--scripting occurs server side

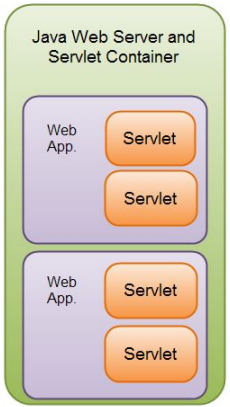


Figure C: Multiple servlets in a single web app container

Common Servlet Tasks

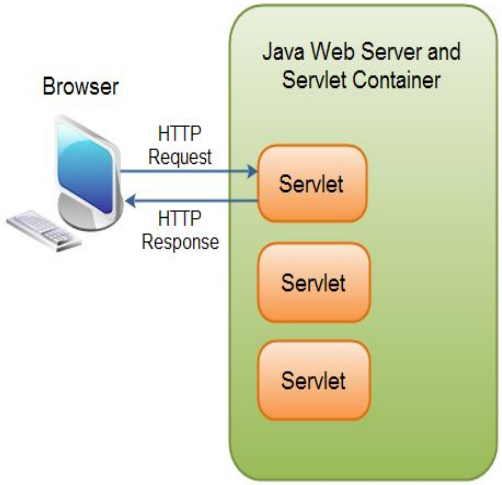


Figure D: Servlet Task

The most common tasks a Java Servlet performs are:

- Read data sent by clients (sent by browsers)
- Read http requests from browsers
- Process data and generate results--meaning connecting to database and processing results or even talking to other servers
- Send explicit data back to browsers (in the form of an html or xml document as well as things like images)
- Send implicit data to browser such as caching user, storing cookies, informing browser that document being generated, etc.

Java Servlet's History

The history of how and when exactly servlets were first conceived and eventually deployed is somewhat muddy, but the history goes roughly as follows:

- Servlets conceived in 1995 by James Gosling (Creator of Java)
- Pavani Diwanji takes servlet concept to develop part of Jeeves
- Jeeves productized into Java Web Server (Sun Microsystems), initial specification 1997
- Server-side Java container developed by Netscape a few years later
- Oracle deploys Tomcat along with the server specification written by James Davidson

Java Servlets

Servlet Lifecycles and OO Design

Servlet Lifecycle

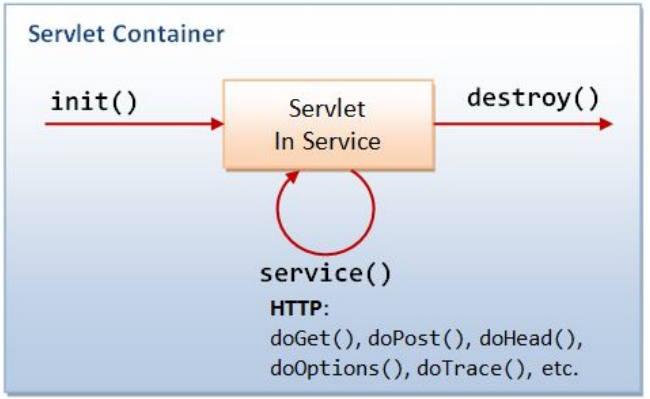


Figure G: Servlet Lifecycle

Every Servlet implements the following methods:

init(), service(), destroy()

- **init()** - This function initializes the servlet instance during initialization phase. This function passes an object that implements the javax.servlet.ServletConfig interface, called exactly once
- **service()** - After initialization the servlet can service client requests(one per thread), service() determines the request type and calls appropriate methods on request type (developers must provide these methods for example doGet() (http get request), doPost() (after http get request handled) etc.)
- **destroy()** - Kills the Servlet service, called exactly once

Java Servlet's Object Oriented Design

Java Servlets are considered to be pure Java objects as they are for Java web based applications

- Servlets provide abstraction through use of interfaces (they implement the Servlet interface, providing an abstraction)
- Servlets abstract away the Hypertext Transfer Protocol response-request paradigm--primarily offering three abstractions:
  - HTTP requests - encapsulates HTTP requests from clients
  - Processing requests for specific application logic
  - HTTP responses - encapsulates http responses from clients
- They are themselves objects thereby allowing them to inherit from parent classes if lots of code is being repeated thus they are also polymorphic

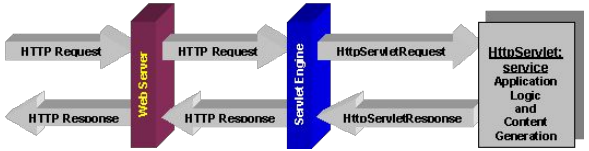


figure H: servlet abstraction

Package Details

Required Packages

- Java Servlets can be created using the **javax.servlet** and **javax.servlet.http** packages

- These packages implement some interfaces: **Java Servlet** and **JSP (Javaserver Pages)** specifications

- Servlets are then created and compiled like normal Java classes, however you will need a Tomcat server up and running to use your servlets

Package details for javax.servlet.*

Java Servlets

The servlet package contains the classes and interfaces which define a contract between the servlet class and the servlet container (i.e Tomcat, Jetty).

- All servlets must implement the methods defined by the **Servlet Interface**
  - **init()**
  - **service()**
  - **destroy()**
  - **getServletConfig()**
  - **getServletInfo()**
- The **GenericServlet** class provides a base protocol-independent servlet.
  - To write a generic servlet you need only to override the abstract **service** method

Package details for javax.servlet.http.*

Java Servlets

The servlet.http package extends the functionality within the servlet package in order to adhere to the http protocol.

- Provides the **HttpServlet** abstract class which can be subclassed to create servlets intended for the web.
- One or more of the following methods must be overridden in your subclass to provide a unique service:
  - **doGet**
  - **doPost**
  - **doPut**
  - **doDelete**

Serving Resources

Service Methods

Java Servlets

The service methods are called by the servlet container to allow the servlet to respond to a request.

- Protocol independent service method (javax.servlet):
  - void **service(ServletRequest** request, **ServletResponse** response**)**
- Http handler methods (javax.servlet.http):
  - void **doGet(HttpServletRequest** request**, HttpServletResponse** response**)**
  - void **doPost(HttpServletRequest** request**, HttpServletResponse** response**)**
  - void **doPut(HttpServletRequest** request**, HttpServletResponse** response**)**
  - void **doDelete(HttpServletRequest** request**, HttpServletResponse** response**)**

Parsing Requests

Java Servlets

The interfaces **ServletRequest** and **HttpServletRequest** are used to format request information for servlets

- Either a **ServletRequest** or a **HttpServletRequest** object is provided as a parameter for all service and handler methods
  - These objects provide the client request information to the servlet
- Client request information can be retrieved using:
  - **getParameter**
  - **getContentType**
  - **getReader**
  - **getServerName**
  - **getProtocol**
  - There are many more and can be protocol dependant for **HttpServletRequest** objects

Constructing Responses

Java Servlets

The interfaces **ServletResponse** and **HttpServletResponse** are used to format response information for clients

- Either a **ServletResponse** or a **HttpServletResponse** object is provided as a parameter for all service and handler methods.
  - These objects are used to construct the response information for the client
- Client response information can be created using:
  - **setContentType**
  - **setBufferSize**
  - **getWriter**
  - **flushBuffer**
  - There are many more and can be protocol dependant for **HttpServletResponse** objects

Sharing Information

Sharing Information

Java Servlets

Servlets can use a variety of techniques to share information, since they have access to anything that any other java class would.

- Private helper objects
  - You can add any required functionality that you need to your servlet class
- Objects which are attributes of a public scope
  - You can use package and class level **scoping** to make objects available to your servlets, as you would any other java class
- Using a database
  - Standard database operations Create, Read, Update, and Delete (CRUD) can be easily accomplished using the **HttpServlet** handler methods **doGet, doPut, doPost,** and **doDelete**

Servlet Container Information

Every servlet must live within a servlet container (e.g. Tomcat) and they provide the following much needed tasks:

- Communication Support
- Lifecycle and Resource Management (i.e. garbage collection)
- Multithreading Support
- JSP Support
- Miscellaneous Task

Java Servlet Example

Conclusion

Java Servlets

Java Servlets are an effective way to abstract away the HTTP Request and Response protocol. They greatly extend Web based apps features by allowing the creation of dynamically designed web pages.

Java servlets are platform independent since they are written in Java, and because of this fact they also take advantage of the many benefits of the Java programming language such as security, full access to Java class libraries, garbage collection etc.

They also have a relatively small learning curve and one can learn to implement them in an afternoon for small projects.

Sources & Additional Resources

Java Servlets

1) http://sqltech.cl/doc/oas10gR31/web.1013/b28959/overview.htm -Used for summary section and for benefits section
2) http://www.tutorialspoint.com/servlets/servlets_overview.htm - Used for summary section, general servlet knowledge, also for Figure A.
3) https://en.wikipedia.org/wiki/Dynamic_web_page - Used to understand dynamic web pages, also for Figure B/Figure F
4) http://tutorials.jenkov.com/java-servlets/overview.html#what-is-a-servlet - Used for benefits section and description of Servlet Scalability, also FigureC/Figure D
5) https://weblogs.java.net/blog/driscoll/archive/2005/12/servlet_history_1.html- For Servlet history

Sources & Additional Resources

6) https://en.wikipedia.org/wiki/Java_servlet - for history, lifecycle and general servlet info
7) http://www3.ntu.edu.sg/home/ehchua/programming/java/JavaServlets.html - for figure G servlet lifecycle
8) https://www.subbu.org/articles/servlets/ServletIssues.html - for OO and servlets, figure H
9) http://www3.ntu.edu.sg/home/ehchua/programming/java/JavaServlets.html - for OO and servlets
10)http://surfpk.com/setting-up-environment-for-java-servlet-programming-java-servlet-tutorial/ - Servlets logo

Sources & Additional Resources

Java Servlets

11)https://tomcat.apache.org/tomcat-5.5-doc/servletapi/javax/servlet/http/package-summary.html - servlet.http package details
12)https://tomcat.apache.org/tomcat-5.5-doc/servletapi/javax/servlet/package-summary.html - servlet package details
13)http://www.tutorialspoint.com/servlets/index.htm -Excellent tutorial on getting servlets up and running with Tomcat
14)http://www.journaldev.com/2015/servlet-interview-questions-and-answers#what-is-servlet - For servlet container (tomcat) information