ITL 3 PRACTICAL - 04

**AIM**

**Write a servlet to remove spam**

**Servlets**

**Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).**

**Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.**

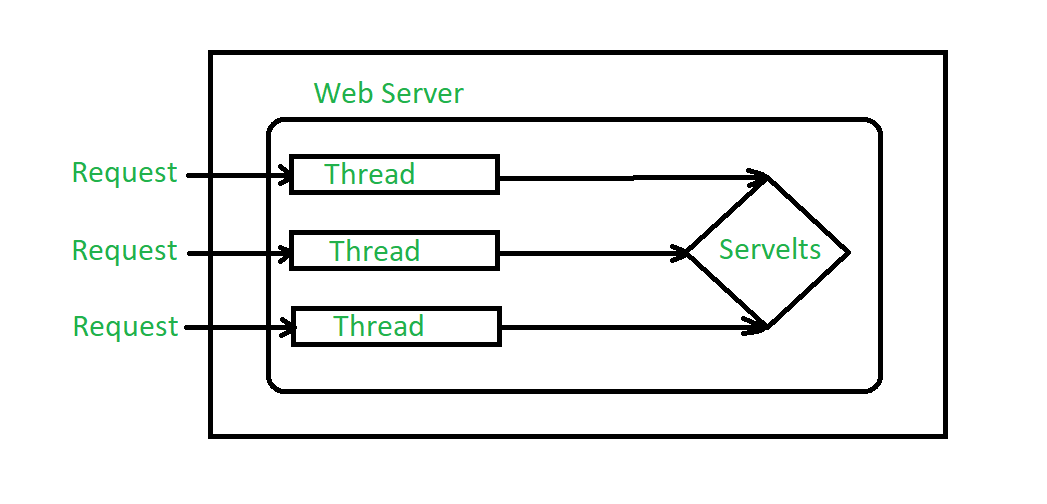
**There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.**

**What is a Servlet?**

**Servlet can be described in many ways, depending on the context.**

* **Servlet is a technology which is used to create a web application.**
* **Servlet is an API that provides many interfaces and classes including documentation.**
* **Servlet is an interface that must be implemented for creating any Servlet.**
* **Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.**
* **Servlet is a web component that is deployed on the server to create a dynamic web page.**

### **Advantages of Servlet**



**There are many advantages of Servlet over CGI. The web container creates threads for handling the multiple requests to the Servlet. Threads have many benefits over the Processes such as they share a common memory area, lightweight, cost of communication between the threads are low. The advantages of Servlet are as follows:**

1. **Better performance: because it creates a thread for each request, not process.**
2. **Portability: because it uses Java language.**
3. **Robust:**[**JVM**](https://www.javatpoint.com/jvm-java-virtual-machine)**manages Servlets, so we don't need to worry about the memory leak,**[**garbage collection**](https://www.javatpoint.com/Garbage-Collection)**, etc.**
4. **Secure: because it uses java language.**

**CODE**

**import com.sun.mail.imap.protocol.FLAGS;**

**import java.io.\*;**

**import java.util.\*;**

**import javax.mail.\*;**

**import javax.mail.internet.\*;**

**public class DeleteMail {**

**public static void main(String args[]) throws Exception {**

**String user= "sonoojaiswal@javatpoint.com";//change accordingly**

**String password="xxxxx";//change accordingly**

**//1) get the session object**

**Properties properties = System.getProperties();**

**Session session = Session.getDefaultInstance(properties);**

**//2) create the store object and connect to the current host**

**Store store = session.getStore("pop3");**

**store.connect("mail.javatpoint.com",user,password);**

**//3) create the folder object and open it**

**Folder folder = store.getFolder("inbox");**

**if (!folder.exists()) {**

**System.out.println("inbox not found");**

**System.exit(0);**

**}**

**folder.open(Folder.READ\_WRITE);**

**//4) Get all the messages and print it (optional)**

**Message[] msg = folder.getMessages();**

**//System.out.println((messages.length+1)+" message found");**

**for (int i = 0; i < msg.length; i++) {**

**System.out.println("--------- " + (i + 1) + "------------");**

**String from = InternetAddress.toString(msg[i].getFrom());**

**if (from != null) {**

**System.out.println("From: " + from);**

**}**

**String replyTo = InternetAddress.toString(**

**msg[i].getReplyTo());**

**if (replyTo != null) {**

**System.out.println("Reply-to: " + replyTo);**

**}**

**String to = InternetAddress.toString(**

**msg[i].getRecipients(Message.RecipientType.TO));**

**if (to != null) {**

**System.out.println("To: " + to);**

**}**

**String subject = msg[i].getSubject();**

**if (subject != null) {**

**System.out.println("Subject: " + subject);**

**}**

**Date sent = msg[i].getSentDate();**

**if (sent != null) {**

**System.out.println("Sent: " + sent);**

**}**

**System.out.println("Message : ");**

**System.out.println(msg[i].getContent());**

**}//end of for loop**

**//5) get the message number to delete (optional)**

**System.out.println("Enter message number to delete :");**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String no = br.readLine();**

**//6) delete the message using setFlag method**

**msg[Integer.parseInt(no) - 1].setFlag(FLAGS.Flag.DELETED, true);**

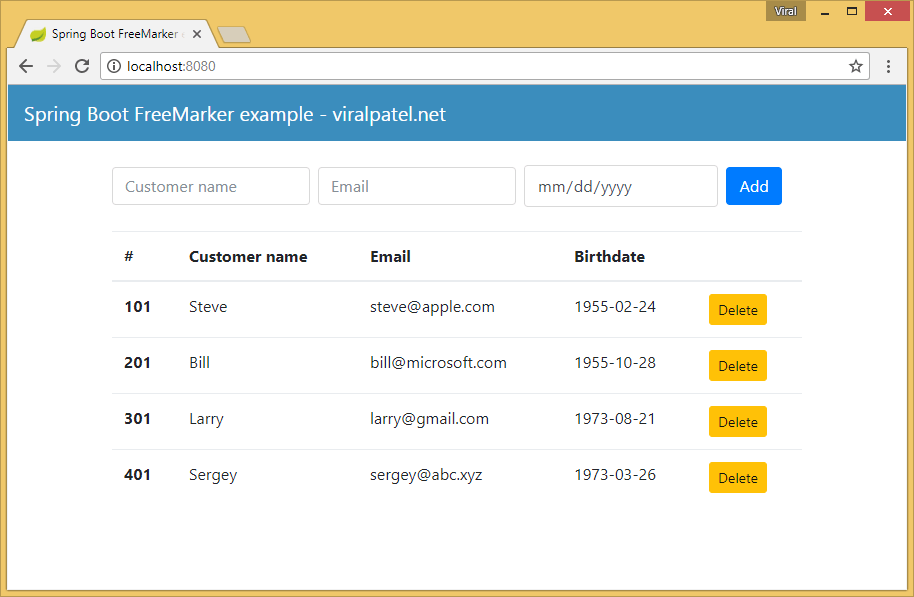
**System.out.println("Msg Delete .....")**

**folder.close(true);**

**store.close();**

**}**

**}**

**OUTPUT**