

GABRIELE ROMANATO

Menu

Come creare un server SMTP di base con Java

Il Simple Mail Transfer Protocol (SMTP) è uno dei protocolli più utilizzati per l'invio di email. In questo articolo vedremo come implementare un server SMTP di base in Java. Questa guida è pensata per chi vuole comprendere il funzionamento di un server SMTP e sperimentare con Java.

Un server SMTP di base ascolta le connessioni in ingresso su una porta specifica (generalmente la porta 25). La struttura di base include:

- Apertura di un socket server.
- Gestione delle connessioni dei client.
- Risposte ai comandi SMTP.

```
import java.io.*;
import java.net.*;

public class SMTPServer {
    private static final int PORT = 2525; // Porta di esempio

    public static void main(String[] args) {
        System.out.println("Avviando il server SMTP sulla
        porta " + PORT);

        try (ServerSocket serverSocket = new
        ServerSocket(PORT)) {
            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Connessione accettata da
                " + clientSocket.getInetAddress());
                new Thread(new
                ClientHandler(clientSocket)).start();
            }
        } catch (IOException e) {
            System.err.println("Errore nel server: " +
```

```

e.getMessage());
    }
}

private static class ClientHandler implements Runnable {
    private Socket clientSocket;

    public ClientHandler(Socket clientSocket) {
        this.clientSocket = clientSocket;
    }

    @Override
    public void run() {
        try (BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
            PrintWriter out = new
PrintWriter(clientSocket.getOutputStream(), true)) {

            // Saluto iniziale del server SMTP
            out.println("220 Simple SMTP Server");

            String line;
            while ((line = in.readLine()) != null) {
                System.out.println("Ricevuto: " + line);

                if (line.startsWith("HELO")) {
                    out.println("250 Hello " +
line.substring(5));
                } else if (line.startsWith("MAIL FROM:"))
{
                    out.println("250 OK");
                } else if (line.startsWith("RCPT TO:")) {
                    out.println("250 OK");
                } else if (line.equals("DATA")) {
                    out.println("354 End data with <CR>
<LF>.<CR><LF>");

                    StringBuilder data = new
StringBuilder();

                    while ((line = in.readLine()) !=
null) {

                        if (line.equals(".")) {
                            break;

```

```

        }
        data.append(line).append("\n");
    }

    System.out.println("Messaggio
ricevuto:\n" + data);
    out.println("250 OK");
} else if (line.equals("QUIT")) {
    out.println("221 Bye");
    break;
} else {
    out.println("502 Command not
implemented");
}
}

} catch (IOException e) {
    System.err.println("Errore durante la
comunicazione con il client: " + e.getMessage());
} finally {
    try {
        clientSocket.close();
    } catch (IOException e) {
        System.err.println("Errore durante la
chiusura del socket: " + e.getMessage());
    }
}
}
}
}
}

```

Conclusione

Creare un server SMTP di base in Java è un ottimo modo per apprendere i fondamenti del protocollo e della programmazione di rete. Questo esempio può essere esteso per creare un server SMTP completo o integrato in progetti più complessi.