

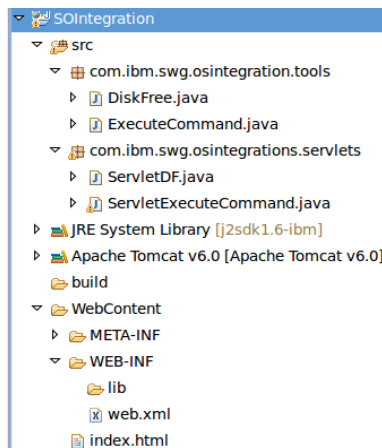
Executando comandos do Sistema Operacional em uma aplicação WEB Java

Criei uma aplicação muito simples que demonstra como um Servlet pode invocar uma classe que efetua um comando qualquer no sistema operacional.

O projeto foi testado em Linux, Unix e Solaris. Caso testem em Windows e puderem me dar um feedback, fico grato.

O mesmo foi criado no Eclipse e utiliza o Tomcat 6 como servidor de aplicação, foi utilizado o Java 6. Deve funcionar com outras versões.

A estrutura do projeto é:



onde:

DiskFree: classe java que executa o comando df no sistema operacional que está rodando o tomcat e retorna um BufferedReader.

ExecuteCommand: classe java que executa qualquer comando que for enviada para ela no host do tomcat.

ServletDF: É o servlet que invoca a classe DiskFree.

ServletExecuteCommand: É o servlet que invoca a classe Executar comando.

index.html: Página principal da aplicação que invoca os servlets e permite que o usuário entre com um comando a ser executado.

Vale lembrar que a aplicação tem finalidade didática, portanto, não foram observados padrões de projeto, boas práticas, etc.

Seguem as classes:

DiskFree.java

```
package com.ibm.swg.osintegration.tools;
```

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
/**
```

```
 * Perform a df command in the host
```

```
 * @author julianom
```

```
 *
```

```
 */
```

```
public class DiskFree {
```

```

    public BufferedReader doWork() {
        BufferedReader Resultset = null;
        String [] Command = null;

        if (System.getProperty("os.name").equals("Linux")) {
            Command = new String[1];
            Command[0] = "df";
        }
        if (System.getProperty("os.name").equals("Solaris")) {
            Command = new String[2];
            Command[0] = "df";
            Command[1] = "-k";
        }
        if (Command == null) {
            return Resultset;
        }

        Process Findspace = null;
        try {
            Findspace = Runtime.getRuntime().exec(Command);
        } catch (IOException e) {
            e.printStackTrace();
            return Resultset;
        }

        Resultset = new BufferedReader(
            new InputStreamReader (
                Findspace.getInputStream()));

        return Resultset;
    }
}

```

ExecuteCommand.java

```

package com.ibm.swg.osintegration.tools;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/**
 * Perform a command in the host
 *
 * @author julianom
 */
public class ExecuteCommand {

    public BufferedReader doWork(String command) {
        BufferedReader Resultset = null;

        if (command == null) {
            return Resultset;
        }

        Process Findspace = null;
        try {
            Findspace = Runtime.getRuntime().exec(command);
        } catch (IOException e) {
            e.printStackTrace();
            return Resultset;
        }
    }
}

```

```

    }

    Resultset = new BufferedReader(new InputStreamReader(Findspace
        .getInputStream()));

    return Resultset;

}
}

```

ServletDF.java

```

package com.ibm.swg.osintegrations.servlets;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ibm.swg.osintegration.tools.DiskFree;
/**
 * Call the DiskFree class
 * @author julianom
 *
 */
public class ServletDF extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><title>Disk Free Monitor</title></head><body>");

        DiskFree df = new DiskFree();
        BufferedReader Resultset = df.doWork();
        if (null != Resultset) {
            String line;
            out.println("<h4>df output</h4>");

            try {
                while ((line = Resultset.readLine()) != null) {
                    out.println(line + "<br>");
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        out.println("<p>Finished</p>");
    }
    else
    {
        out.print("<p>Unable to perform the df command</p>");
    }

    out.println("</body></html>");
    out.close();
}
}
}

```

ServletExecuteCommand.java

```

package com.ibm.swg.osintegrations.servlets;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ibm.swg.osintegration.tools.DiskFree;
import com.ibm.swg.osintegration.tools.ExecuteCommand;

/**
 * Servlet implementation class ServletExecuteCommand Call ExecuteCommand class
 * This servlet call the Execute command class and show its results in the screen
 */
public class ServletExecuteCommand extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        String command = request.getParameter("command");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><title>Execute command</title></head><body>");

        ExecuteCommand ec = new ExecuteCommand();
        BufferedReader Resultset = ec.doWork(command);

        if (null != Resultset) {
            String line;
            out.println("<h4>" + command + " output</h4>");

            try {
                while ((line = Resultset.readLine()) != null) {
                    out.println(line + "<br>");
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        }
        out.println("<p>Finished</p>");
    } else {
        out.print("<p>Command " + command + " failed!!!</p>");
    }

    out.println("</body></html>");
    out.close();
}
}

```

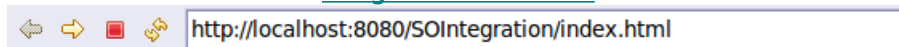
index.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>OS integration</title>
</head>
<body>
<p>This is an example of how a web application written in Java can perform OS
commands. (There are a lot of another ways to do this)</p>
<p>Este é um exemplo de como uma aplicação web Java pode executar comandos do
sistema operacional. (Existem outras formas de se fazer isso)</p>
<p>Tested/Testado = Linux/Unix/Solaris</p>
<hr>
<a href="ServletDF">Call df / Chamar df</a><br>
<form action="ServletExecuteCommand">
Call command/Chamar comando: <input type="text" name="command" value="ls -la"/>
<input type="submit" value="Submit"/>
</form>
<p>Fonte: <a href="http://jmmwrite.wordpress.com">Juliano Martins</a></p>
</body>
</html>

```

Programa rodando:



This is an example of how a web application written in Java can perfo
another ways to do this)

Este é um exemplo de como uma aplicação web Java pode executar c
(Existem outras formas de se fazer isso)

Tested/Testado = Linux/Unix/Solaris

[Call df / Chamar df](#)

Call command/Chamar comando:

Fonte: [Juliano Martins](#)

ls -la output

```
total 161188
drwxr-xr-x 70 julianom julianom 4096 2010-01-12 18:15 .
drwxr-xr-x  5 root root 4096 2009-11-12 15:37 ..
drwx----- 3 julianom julianom 4096 2009-09-15 15:09 .adobe
drwx----- 2 julianom julianom 4096 2010-01-11 17:36 .agnc
drwx----- 10 julianom julianom 4096 2010-01-11 09:04 .amsn
drwx----- 2 julianom julianom 4096 2009-12-07 16:15 amsn_received
-rw----- 1 julianom julianom 7649 2010-01-12 18:08 .bash_history
-rw-r--r-- 1 julianom julianom 220 2009-09-02 16:29 .bash_logout
-rw-r--r-- 1 julianom julianom 3318 2009-09-24 14:35 .bashrc
-rw-r--r-- 1 julianom julianom 3266 2009-09-17 13:50 .bashrc~
drwxr-xr-x  2 julianom julianom 4096 2009-11-26 10:37 bin
drwx----- 2 julianom julianom 4096 2009-11-10 16:41 .bogofilter
drwxr-xr-x  9 julianom julianom 4096 2010-01-11 18:56 .cache
drwx----- 2 julianom julianom 4096 2009-11-27 11:36 .camel_certs
drwxr-xr-x  3 julianom julianom 4096 2009-11-03 17:47 ccrc_linux
-rw-r--r-- 1 julianom julianom 163801975 2009-10-14 17:41 ccrc_linux.zip
drwx----- 3 julianom julianom 4096 2009-09-15 15:42 .compiz
```

Fonte: <http://jmmwrite.wordpress.com>