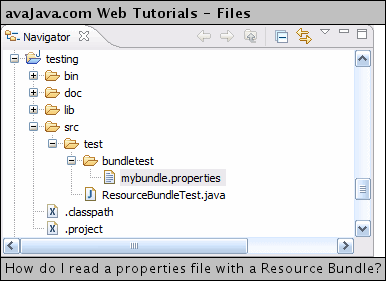
*[Splitting a Java String by the pipe symbol using split(“|”)](http://stackoverflow.com/questions/10796160/splitting-a-java-string-by-the-pipe-symbol-using-split)*

String test = “amit|kumar|rajeev|mahankudo|bangalore”;

test.split("\\|");

test.split(Pattern.quote("|"));

*How do I read a properties file with a Resource Bundle?*



public static void main(String[] args) {

ResourceBundle rb = ResourceBundle.getBundle("test.bundletest.mybundle");

Enumeration <String> keys = rb.getKeys();

while (keys.hasMoreElements()) {

String key = keys.nextElement();

String value = rb.getString(key);

System.out.println(key + ": " + value);

}

}

# *Convert ResourceBundle to Map : ResourceBundle « I18N « Java Tutorial*

Map<String, String> map = **new** HashMap<String, String>();

Enumeration<String> keys = resource.getKeys();

**while** (keys.hasMoreElements()) {

String key = keys.nextElement();

map.put(key, resource.getString(key));

}

Iterator iterator = map.keySet().iterator();

**while** (iterator.hasNext()) {

String key = (String) iterator.next();

String value = map.get(key);

System.out.println(key + " = " + value);

}

*Check for server is running or not*

public class ServerRunningStatus{

private static final String SERVER\_ADDRESS = "192.144.10.10"/localhost;

    private static final int TCP\_SERVER\_PORT = 8890;

    private static boolean connected = false;

    static Socket s;

    public static void main(String[] args) {

    Timer timer = new Timer();

    timer.schedule(task, 01, 5001);

}

static TimerTask task = new TimerTask() {

        @Override

        public void run() {

            // TODO Auto-generated method stub

            Boolean serverRunningStatus = hostAvailabilityCheck();

if(serverRunningStatus)

System.out.println(“Server is running”);

else

System.out.println(“Server is not running”);

        }

    };

    public static boolean hostAvailabilityCheck()

    {

        boolean available = true;

        try {

             if (connected == false)

             {

(s = new Socket(SERVER\_ADDRESS, TCP\_SERVER\_PORT)).close();

             }

            }

        catch (UnknownHostException e)

            { // unknown host

             available = false;

            s = null;

            }

        catch (IOException e) {

// io exception, service probably not running

             available = false;

            s = null;

            }

        catch (NullPointerException e) {

            available = false;

            s=null;

        }

      return available;

}

*Prevent the Saturday and Sunday*

public static void main(String[] args) {

Calendar startingTime = Calendar.*getInstance*();

if (startingTime.get(Calendar.*DAY\_OF\_WEEK*) != Calendar.*SATURDAY*

&& startingTime.get(Calendar.*DAY\_OF\_WEEK*) != Calendar.*SUNDAY*) {

System.*out*.println("\*\*\*\*\*\*\*other day\*\*\*\*\*\*\*\*\*\*\*");

}

System.*out*.println("\*\*\*\*\*\*\*\*\*done done\*\*\*\*\*\*\*\*\*\*");

}

*Generating java object to csv file*

Link for javascvjar api

<http://www.java2s.com/Code/Jar/j/Downloadjavacsvjar.htm>

private void CsvWriterAppendExample(Employee emp) {

Date today = new Date();

SimpleDateFormat dateFormat = new SimpleDateFormat("ddMMyyyy");// ddMMyyyy\_HHmmSS

String date = dateFormat.format(today);

String outputFileName = "src/com/dom/parse/employee\_" + date + ".csv";

CsvWriter csvOutput = null;

final String NEW\_LINE\_SEPARATOR = "\n";

// before we open the file check to see if it already exists

boolean alreadyExists = new File(outputFileName).exists();

try {

csvOutput = new CsvWriter(new FileWriter(outputFileName, true), ',');

// if the file didn't already exist then we need to write out the

// header line

if (!alreadyExists) {

csvOutput.write("Name");

csvOutput.write("Id");

csvOutput.write("Age");

csvOutput.write("Type");

csvOutput.write("Address");

csvOutput.endRecord();

}

csvOutput.write(emp.getName());

csvOutput.write(String.valueOf(emp.getId()));

csvOutput.write(String.valueOf(emp.getAge()));

csvOutput.write(emp.getType());

csvOutput.write(emp.getAddress());

//csvOutput.write(NEW\_LINE\_SEPARATOR);

csvOutput.endRecord();

} catch (IOException e) {

e.printStackTrace();

}

finally {

csvOutput.flush(); csvOutput.close();

}

}

*Getting the serverInforation*

**public** **static** List<String> getEndPoints()

**throws** MalformedObjectNameException, NullPointerException, UnknownHostException, AttributeNotFoundException,

InstanceNotFoundException, MBeanException, ReflectionException {

MBeanServer mbs = ManagementFactory.*getPlatformMBeanServer*();

Set<ObjectName> objs = mbs.queryNames(**new** ObjectName("\*:type=Connector,\*"),

Query.*match*(Query.*attr*("protocol"), Query.*value*("HTTP/1.1")));

String hostname = InetAddress.*getLocalHost*().getHostName();

InetAddress[] addresses = InetAddress.*getAllByName*(hostname);

ArrayList<String> endPoints = **new** ArrayList<String>();

**for** (Iterator<ObjectName> i = objs.iterator(); i.hasNext();) {

ObjectName obj = i.next();

String scheme = mbs.getAttribute(obj, "scheme").toString();

String port = obj.getKeyProperty("port");

System.***out***.println("\*\*\*\*\*\*\*server port number\*\*\*\*\*\*\*>" + port);

**for** (InetAddress addr : addresses) {

String host = addr.getHostAddress();

String ep = scheme + "://" + host + ":" + port;

endPoints.add(ep);

}

}

**return** endPoints;

}

***JAXB – Marshalling and UnMarshalling***

**JAXB** stands for **Java Architecture for XML Binding.**

JAXB parses the annotations defined on the classes to gather information and then converts XML to Java Objects / Java Object to XML document.

In the DOM Approach to XML parsing the DOM creates an in memory tree structure of the entire XML document and the programmer needs to traverse these Nodes to extract the data related to the XML.

* Marshalling –  process of the converting a Java Object to XML
* UnMarshalling – process of converting XML document to Java Objects.

<https://dzone.com/articles/parsing-xml-using-dom-sax-and>

<http://www.java67.com/2012/09/dom-vs-sax-parser-in-java-xml-parsing.html>

<https://www.javacodegeeks.com/2013/05/parsing-xml-using-dom-sax-and-stax-parser-in-java.html>

<http://javahash.com/jaxb-example-marshalling-and-unmarshalling/>