public class ImageServlet extends HttpServlet {

private static final String SQL\_FIND = "SELECT content FROM Image WHERE name = ?";

Resource(name="jdbc/yourDB")

private DataSource dataSource;

Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

String imageName = request.getPathInfo().substring(1);

try (Connection connection = dataSource.getConnection();

PreparedStatement statement = connection.prepareStatement(SQL\_FIND))

{

statement.setString(1, imageName);

try (ResultSet resultSet = statement.executeQuery())

{

if (resultSet.next())

{

byte[] content = resultSet.getBytes("content");

response.setContentType(getServletContext().getMimeType(imageName));

response.setContentLength(content.length);

response.getOutputStream().write(content);

}

else {

response.sendError(HttpServletResponse.SC\_NOT\_FOUND);

}

}

}

catch (SQLException e)

{

throw new ServletException("Something failed at SQL/DB level.", e);

}

}

}

InputStream input = new URL("http://WWW.GMAIL.COM").openStream();

Document document = new DOC().parseDOM(input, null);

NodeList imgs = document.getElementsByTagName("img");

List<String> srcs = new ArrayList<String>();

for (int i = 0; i < imgs.getLength(); i++)

{

srcs.add(imgs.item(i).getAttributes().getNamedItem("src").getNodeValue());

}

for (String src: srcs)

{

System.out.println(src);

}

Parser parser = new Parser("http://www.yahoo.com/");

NodeList list = parser.parse(new TagNameFilter("IMG"));

for ( SimpleNodeIterator iterator = list.elements(); iterator.hasMoreNodes(); )

{

Tag tag = (Tag) iterator.nextNode();

System.out.println(tag.getAttribute("src"));

}

Parser parser = new Parser("http://www.yahoo.com/");

NodeList list = parser.parse(new TagNameFilter("IMG"));

for ( SimpleNodeIterator iterator = list.elements(); iterator.hasMoreNodes(); )

{

Tag tag = (Tag) iterator.nextNode();

System.out.println(tag.getAttribute("src"));

}

Parser parser = new Parser(url);

PageMeta pageMeta = new PageMeta();

pageMeta.setUrl(url);

NodeList meta = parser.parse(new TagNameFilter("meta"));

for (SimpleNodeIterator iterator = meta.elements(); iterator.hasMoreNodes(); )

{

Tag tag = (Tag) iterator.nextNode();

if ("og:image".equals(tag.getAttribute("property")))

{

pageMeta.setImageUrl(tag.getAttribute("content"));

}

if ("og:title".equals(tag.getAttribute("property")))

{

pageMeta.setTitle(tag.getAttribute("content"));

}

if ("og:description".equals(tag.getAttribute("property")))

{

pageMeta.setDescription(tag.getAttribute("content"));

}

}