The Java™ Tutorials

Trail: Custom Networking
Lesson: Working with URLs

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases and might use technology no longer available.
See JDK Release Notes for information about new features, enhancements, and removed or deprecated options for all JDK releases.

Reading Directly from a URL

After you've successfully created a `URL`, you can call the `URL`'s `openStream()` method to get a stream from which you can read the contents of the URL. The `openStream()` method returns a `java.io.InputStream` object, so reading from a URL is as easy as reading from an input stream.

The following small Java program uses `openStream()` to get an input stream on the URL `http://www.oracle.com/`. It then opens a `BufferedReader` on the input stream and reads from the `BufferedReader` thereby reading from the URL. Everything read is copied to the standard output stream:

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

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

        URL oracle = new URL("http://www.oracle.com/");
        BufferedReader in = new BufferedReader(
        new InputStreamReader(oracle.openStream()));

        String inputLine;
        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);
        in.close();
    }
}
```

When you run the program, you should see, scrolling by in your command window, the HTML commands and textual content from the HTML file located at `http://www.oracle.com/`. Alternatively, the program might hang or you might see an exception stack trace. If either of the latter two events occurs, you may have to set the proxy host so that the program can find the Oracle server.

Next page: Connecting to a URL