

第8章 Java Servlet

8.1 Servlet とは

Java Applet が Web のクライアント（ブラウザ）側で実行されるプログラムであるのに対し、Java Servlet は、Web の _____ で実行されるプログラムである。例えば、アクセスカウンタなどはサーバ側で実行しなければ実現できないプログラムである。

Servlet と関連する言葉として、_____ (Common Gateway Interface) がある。CGI は Web サーバ上でプログラムを実行し、動的に HTML 形式などのデータなどを作成して Web ブラウザに渡すための仕組み（プログラムと Web サーバの間のデータのやりとりの約束事）である。また CGI に従って実行されるプログラム自体のことも CGI と呼ばれる。CGI は記述するプログラミング言語を問わないが、Perl や C を使うことが多いようである。

Java Servlet も CGI と同じように、Web サーバ上でプログラムを実行し、動的に HTML 形式などのデータなどを作成して Web ブラウザに渡すための仕組み、またはその仕組みにしたがって実行されるプログラム、のことであるが、Java 専用設計されたインタフェースであるところが CGI と異なる。

Java Servlet を実行するためには Web サーバ（Apache など）と Servlet の仲立ちをするプログラムが必要である。このようなプログラムを Servlet コンテナという。（Servlet コンテナ自体が Web サーバの役割まで果たすこともある。）無料で手に入る Servlet コンテナとして、有名なものに Jakarta Tomcat¹ がある。

クライアントからの接続の受け付けや HTTP プロトコルの取扱などの部分は Web サーバ（あるいは Servlet コンテナ）が行なうので、Java Servlet は専らページの生成に相当する部分を担当する。

Servlet コンテナのインストールの方法や、Servlet の配置の方法は、Servlet コンテナによって大きく異なるので、ここでは Servlet のプログラムの書き方に絞って、説明することにする²

8.2 簡単な Servlet

Servlet は、CGI と同様、HTML のデータ³を動的に生成するサーバ側のプログラムである。

Servlet は `HttpServlet` というクラスを継承して作成し、ページを出力するための処理は基本的に _____（または _____）というメソッドの中に記述する。`doGet/doPost` は `HttpServletRequest` 型、`HttpServletResponse` 型の 2 つの引数を取る。

また、Servlet に関するクラスを参照するために、通常、次の 2 行:

```
import javax.servlet.*;
import javax.servlet.http.*;
```

¹<http://jakarta.apache.org/tomcat/>

²このプログラムのコンパイル方法や、サーバに配置する方法はここでは説明しないので、これらの点に興味のある人は、Web 上の文献を探してみることに。

³JPEG や PNG や SVG など HTML 以外のデータを出力する CGI/Servlet もある。

の `import` 文が最初のほうに必要である。

例題 8.2.1 現在の時刻を表示する *Servlet*

ファイル *MyDate.java*

```
import java.io.*;
import java.util.*; // Date クラス用に必要
import javax.servlet.*;
import javax.servlet.http.*;

public class MyDate extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) {
        res.setContentType("text/html");
        try {
            PrintWriter out = res.getWriter();
            out.println("<html><head></head><body>");

            Date d = new Date();
            out.println(d.toString());

            out.println("</body></html>");
            out.close(); // close を忘れない
        } catch (Exception e) {}
    }
}
```

MyDate.java の場合 `doGet` メソッドの最初の

```
res.setContentType("text/html");
```

は、以下に続くデータが *HTML* のデータであるということをブラウザに伝える役割を持つ。(ファイル形式ごとに *ContentType* として指定する文字列 “*MIME type*” は決まっている。)

また、

```
PrintWriter out = res.getWriter();
```

は、ブラウザにデータを送るための出力ストリームを取得する。これ以降 `out` オブジェクトの `println` (あるいは `print`) メソッドを呼び出すことにより、データを出力することができる。

8.3 少し高度な *Servlet*

Servlet は、通常クライアントから受けとったデータ (フォームのデータ) により振舞を変えたり、サーバ側のデータベースにアクセスしたりするなど、さまざまな処理を行なうことができる。そのすべてを少ないページで紹介することはできないが、次に紹介する例は、URL 中の *QueryString* と呼ばれるパラメータ (URL 中の “?” から後の文字列) を受けとり、それによって出力を変えるプログラムである。

例題 8.3.1 SVG 形式のグラフ生成

ファイル *GraphSVG.java*

```
import java.io.*;
import java.util.StringTokenizer;
import javax.servlet.*;
import javax.servlet.http.*;

public class GraphSVG extends HttpServlet {
    final private static int ht = 20, unit = 5;
    public void doGet(HttpServletRequest req, HttpServletResponse res) {
        res.setContentType("image/svg+xml");
        String str = req.getQueryString();
        StringTokenizer st = new StringTokenizer(str, "_");
        int c = st.countTokens();

        try {
            PrintWriter out = res.getWriter();
            out.println("<svg width='"+(unit*100)+"' height='"+(c*ht)+"'>");

            int i;
            for (i=0; i<c; i++) {
                out.print("<rect stroke='red' fill='orange'");
                int w = Integer.parseInt(st.nextToken());
                out.print(" x='0' y='"+(i*ht)+"' width='"+(unit*w)+"'");
                out.println(" height='"+ht+"' />");
            }
            out.println("</svg>");
            out.close();
        } catch (Exception e) {}
    }
}
```

このプログラムでは、

```
String str = req.getQueryString();
```

の部分で、この *Servlet* を呼び出す *URL* の “?” 以降の部分の文字列を変数 *str* に代入している。例えば次のような *URL* でこの *Servlet* を呼び出すと、

```
http://...略.../GraphSVG?10_40_70_20_80
```

10_40_70_20_80 という文字列が変数 *str* に代入される。

また、このプログラムの出力は *HTML* 形式ではなく *SVG*⁴ という形式になっている。ここでは詳しく説明しないが、*SVG* は *HTML* と同じようなタグ付きのテキストでベクター形式の画像を表現する規格である。

Servlet の書き方自体は、出力形式が *HTML* でも *SVG* でも大きく異なることはない。ただし、ブラウザにデータの内容を示す *ContentType* は、*HTML* の場合、

```
res.setContentType("text/html");
```

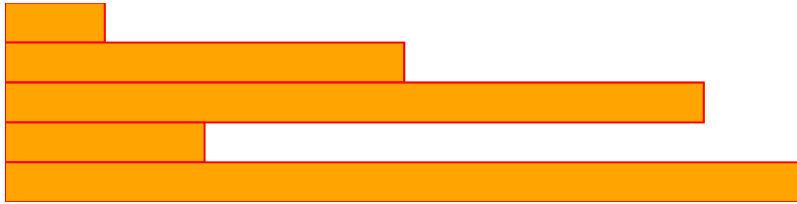
であったのが、*SVG* の場合、

⁴<http://www.w3.org/TR/SVG/>

```
res.setContentType("image/svg+xml");
```

とする必要がある。

例えば、`10_40_70_20_80` という文字列をパラメータとして渡すと、この *Servlet* は次のようなグラフを生成する。



Applet のグラフィックスを描画する時と異なり、この画像はサーバ側で生成されていることに注意する。

また、JPEG や PNG 形式などの画像を *Servlet* で生成することも可能である。ただし、これらの画像形式はすべてバイナリ形式であるため、HTML や SVG のようなテキスト形式に比べると、ややプログラムに手間がかかる。

キーワード Servlet, Jakarta Tomcat, doGet/doPost, SVG, getQueryString,