

# CoolUrls

NOTE: This is outdated information that applies only to Tapestry 3 and 4.

## CoolUrls

It doesn't contain the character string that shows that it is Tapestry in URL. The extension is not needed. (Do not be.)

### Why is it so?

<http://www.w3.org/Provider/Style/URI.html>

### URL Example

- [PageService](#)
  - /path/to/page
- [DirectService](#)
  - /path/to/page/jwc
- [RestartService](#)
  - /path/to/restart

<http://nageyarinadake.minidns.net/CoolUrls> It is being written in Japanese and English(Autoselect). It serves with Tapestry.

```
*** ./tapestry/engine/AbstractService.java          Sat Apr 23 20:24:43 2005
--- AbstractService.java          Fri Apr 29 00:18:28 2005
*****
*** 76,82 ***
        throw new ApplicationRuntimeException(ex);
    }

!        return new EngineServiceLink(cycle, serviceName, serviceContext, squeezed, stateful);
!    }

!        /**
--- 76,84 ----
        throw new ApplicationRuntimeException(ex);
    }

!        ILink link = new EngineServiceLink(cycle, serviceName, serviceContext, squeezed, stateful);
!        System.out.println( "link-url:" + link.getURL() );
!        return link;
!    }

!        /**
*** ./tapestry/engine/EngineServiceLink.java        Tue Mar 29 11:28:27 2005
--- EngineServiceLink.java        Fri Apr 29 01:02:47 2005
*****
*** 47,52 ***
--- 47,53 ----

        private IRequestCycle _cycle;
        private String _service;
+       private String[] _context;
        private String[] _parameters;
        private boolean _stateful;

*****
*** 84,111 ***
        boolean stateful)
        {
            _cycle = cycle;
!            _service = constructServiceValue(serviceName, serviceContext);
            _parameters = serviceParameters;
            _stateful = stateful;
-        }
-    }
```

```

-     private String constructServiceValue(String serviceName, String[] serviceContext)
-     {
-         int count = Tapestry.size(serviceContext);
-
-         if (count == 0)
-             return serviceName;
-         !
-         !
-         !     StringBuffer buffer = new StringBuffer(serviceName);
-         !
-         !     for (int i = 0; i < count; i++)
-         !     {
-         !         buffer.append('/');
-
-         !         buffer.append(serviceContext[i]);
-         !     }
-
-         return buffer.toString();
-     }
-
-     public String getURL()
--- 85,99 ----
-     {
-         boolean stateful)
-     {
-         _cycle = cycle;
-         _service = serviceName;
-         !     _context = serviceContext;
-         !     _parameters = serviceParameters;
-         !     _stateful = stateful;
-
-         int count = Tapestry.size(_context);
-         if (count==0)
-         !             System.err.println("context size 0:" + this._service);
-
-     }
-
-     public String getURL()
*****
*** 161,180 ****
-
-     private String constructURL(StringBuffer buffer, String anchor, boolean includeParameters)
-     {
-         !     buffer.append(_cycle.getEngine().getServletPath());
-
-         if (includeParameters)
-         {
-             !     buffer.append('?');
-             !     buffer.append(Tapestry.SERVICE_QUERY_PARAMETER_NAME);
-             !     buffer.append('=');
-             !     buffer.append(_service);
-
-             !     int count = Tapestry.size(_parameters);
-
-             for (int i = 0; i < count; i++)
-             {
-                 !     buffer.append('&');
-
-                 !     buffer.append(Tapestry.PARAMETERS_QUERY_PARAMETER_NAME);
-                 !     buffer.append('=');
--- 149,173 ----
-
-     private String constructURL(StringBuffer buffer, String anchor, boolean includeParameters)
-     {
-         !     int count = Tapestry.size(this._context);
-         !     for (int i = 0; i < count; i++)
-         !     {
-         !         if (this._context[i].charAt(0)!='/')
-         !             buffer.append('/');
-         !         buffer.append(this._context[i]);
-         !     }
-
-         if (includeParameters)

```

```

        {
!           count = Tapestry.size(_parameters);

            for (int i = 0; i < count; i++)
            {
!               char sep = '&';
!               if (i==0)
!                   sep = '?';
!               buffer.append(sep);

                buffer.append(Tapestry.PARAMETERS_QUERY_PARAMETER_NAME);
                buffer.append('=');
*** ./tapestry/engine/DirectService.java          Sat Apr 23 22:03:19 2005
--- DirectService.java          Fri Apr 29 02:13:50 2005
*****
*** 74,87 ****

                boolean complex = renderPage != componentPage;

!           String[] context = complex ? new String[4] : new String[3];

                int i = 0;

-           String stateful = cycle.getEngine().isStateful() ? STATEFUL_ON : STATEFUL_OFF;
-
-           context[i++] = stateful;
-
                if (complex)
                    context[i++] = renderPage.getPageName();
--- 74,83 ----

                boolean complex = renderPage != componentPage;

!           String[] context = complex ? new String[3] : new String[2];

                int i = 0;

                if (complex)
                    context[i++] = renderPage.getPageName();

*** ./tapestry/asset/AssetService.java          Tue Mar 29 11:28:27 2005
--- AssetService.java          Fri Apr 29 01:22:52 2005
*****
*** 90,96 ****

                // Service is stateless

!           return constructLink(cycle, Tapestry.ASSET_SERVICE, null, parameters, false);
            }

            public String getName()
--- 90,102 ----

                // Service is stateless

!           int count = Tapestry.size(parameters);
!           String parameter = (String)parameters[0];
!           int dot = parameter.lastIndexOf('.');
!           String[] context = new String[1];
!           context[0] = parameter.substring(0,dot);
!
!           return constructLink(cycle, Tapestry.ASSET_SERVICE, context, null, false);
            }

            public String getName()
*****
*** 137,143 ****

                Tapestry.format("service-incorrect-parameter-count", Tapestry.ASSET_SERVICE, new Integer(2));

```

```

        String resourcePath = (String) parameters[0];
!       String checksum = (String) parameters[1];

        URL resourceURL = engine.getResourceResolver().getResource(resourcePath);

--- 143,149 ----
        Tapestry.format("service-incorrect-parameter-count", Tapestry.ASSET_SERVICE, new Integer(2));

        String resourcePath = (String) parameters[0];
!       String contentType = (String) parameters[1];

        URL resourceURL = engine.getResourceResolver().getResource(resourcePath);

*****
*** 145,163 ****
        throw new ApplicationRuntimeException(
            Tapestry.format("missing-resource", resourcePath));

-       String actualChecksum = engine.getResourceChecksumSource().getChecksum(resourceURL);
-
-       if (!actualChecksum.equals(checksum))
-       {
-           throw new ApplicationRuntimeException(
-               Tapestry.format("AssetService.checksum-failure", checksum, resourcePath));
-       }

        URLConnection resourceConnection = resourceURL.openConnection();

        ServletContext servletContext = cycle.getRequestContext().getServlet().getServletContext();

!       writeAssetContent(engine, cycle, output, resourcePath, resourceConnection, servletContext);
    }

    /** @since 2.2 */
--- 151,162 ----
        throw new ApplicationRuntimeException(
            Tapestry.format("missing-resource", resourcePath));

        URLConnection resourceConnection = resourceURL.openConnection();

        ServletContext servletContext = cycle.getRequestContext().getServlet().getServletContext();

!       writeAssetContent(engine, cycle, output, resourcePath, resourceConnection, servletContext,
contentType);
    }

    /** @since 2.2 */
*****
*** 168,180 ****
        ResponseOutputStream output,
        String resourcePath,
        URLConnection resourceConnection,
!       ServletContext servletContext)
    {
        // Getting the content type and length is very dependant
        // on support from the application server (represented
        // here by the servletContext).

-       String contentType = servletContext.getMimeType(resourcePath);
-       int contentLength = resourceConnection.getContentLength();

        try
--- 167,179 ----
        ResponseOutputStream output,
        String resourcePath,
        URLConnection resourceConnection,
!       ServletContext servletContext,
!       String contentType)
    {
        // Getting the content type and length is very dependant

```

```

        // on support from the application server (represented
        // here by the servletContext).

        int contentLength = resourceConnection.getContentLength();

        try

```

#### ServiceFilter.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public abstract class ServiceFilter implements Filter
{
    protected String _applicationPath = "/app";
    protected String _serviceName;

    public void init(FilterConfig c)
        throws ServletException
    {
        final String applicationPath = c.getInitParameter("application-path");
        if (applicationPath!=null)
            this._applicationPath = applicationPath;

        this._serviceName = this.initServiceName();
        if (this._serviceName==null)
            throw new ServletException("could not initialized service name");

    }

    public void destroy()
    {
    }

    public String getApplicationPath()
    {
        return this._applicationPath;
    }

    public String getServiceName()
    {
        return this._serviceName;
    }

    protected void replaceSlashToPeriod(char[] array,int start,int end)
    {
        for (int i=start;i<end;i++)
            if (array[i]=='/')
                array[i] = '.';
    }

    public void doFilter(final ServletRequest q,final ServletResponse s,final FilterChain c)
        throws IOException,ServletException
    {
        this.doHttpFilter((HttpServletRequest)q,(HttpServletResponse)s,c);
    }

    public void doHttpFilter(final HttpServletRequest request,final HttpServletResponse response,final
FilterChain chain)
        throws IOException,ServletException
    {
        final StringBuffer buffer = new StringBuffer();

        //application servlet
        buffer.append(request.getContextPath());
        buffer.append(this.getApplicationPath());

        //service parameter
        buffer.append("?service=");

```

```

        buffer.append(this.getServiceName());

        //construct path
        this.constructPath(buffer,request);

        //dispatching
        final String redirectPath = buffer.toString();
        System.err.println( "redirect-path:" + redirectPath );
        final RequestDispatcher dispatcher = request.getRequestDispatcher(redirectPath);
        dispatcher.forward(request,response);
    }

    protected abstract String initServiceName();
    protected abstract void constructPath(final StringBuffer buffer,final HttpServletRequest request);
}

```

#### PageServiceFilter.java

```

import javax.servlet.http.*;

public class PageServiceFilter extends ServiceFilter
{
    protected String initServiceName()
    {
        return "page";
    }

    protected void constructPath(final StringBuffer buffer,final HttpServletRequest request)
    {
        final String servletPath = request.getServletPath();
        final char[] file = servletPath.toCharArray();
        this.replaceSlashToPeriod(file,1,file.length);
        buffer.append(file);
    }
}

```

#### DirectServiceFilter.java

```

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

public class DirectServiceFilter extends ServiceFilter
{
    protected String _stateful = "0";

    public void init(FilterConfig c)
        throws ServletException
    {
        super.init(c);

        final String stateful = c.getInitParameter("stateful");
        if (stateful!=null)
            this._stateful = "1";
    }

    protected String initServiceName()
    {
        return "direct";
    }

    protected void constructPath(final StringBuffer buffer,final HttpServletRequest request)
    {
        final String servletPath = request.getServletPath();
        //stateful
        buffer.append('/');
        buffer.append(this._stateful);

        //pages and form
        final int lastIndex = servletPath.lastIndexOf('/');
        final char[] file = servletPath.toCharArray();
        this.replaceSlashToPeriod(file,1,lastIndex);
        buffer.append(file);
    }
}

```

#### [RestartServiceFilter.java](#)

```

import javax.servlet.http.*;

public class RestartServiceFilter extends ServiceFilter
{
    protected String initServiceName()
    {
        return "restart";
    }

    protected void constructPath(final StringBuffer buffer,final HttpServletRequest request)
    {
    }
}

```

#### web.xml

```
<filter>
  <filter-name>page</filter-name>
  <filter-class>PageServiceFilter</filter-class>
</filter>

<filter>
  <filter-name>direct</filter-name>
  <filter-class>DirectServiceFilter</filter-class>
</filter>

<filter>
  <filter-name>css</filter-name>
  <filter-class>AssetServiceFilter</filter-class>
  <init-param>
    <param-name>extention</param-name>
    <param-value>css</param-value>
  </init-param>
  <init-param>
    <param-name>content-type</param-name>
    <param-value>text/css</param-value>
  </init-param>
</filter>

<filter>
  <filter-name>restart</filter-name>
  <filter-class>RestartServiceFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>page</filter-name>
  <url-pattern>/Home</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>direct</filter-name>
  <url-pattern>/Home/FormName</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>restart</filter-name>
  <url-pattern>/Restart</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>css</filter-name>
  <url-pattern>/org/apache/tapestry/page/Exception</url-pattern>
</filter-mapping>
```