

## [ServletRequest.setCharacterCoding\(\) Ignored](#)

By [Roger Keays](#), 2 April 2012

Take a look at this block of servlet code and see if you can tell me what the output should be when I send x=

```
Log.warning(Charset.defaultCharset().toString());
Log.warning(request.getCharacterEncoding());
Log.warning(request.getParameter("x"));
request.setCharacterEncoding("UTF-8");
Log.warning(request.getCharacterEncoding());
Log.warning(request.getParameter("x"));
```

Well, here is the output

```
UTF-8
null
```

```
UTF-8
```

Compare this code

```
Log.warning(Charset.defaultCharset());
Log.warning(request.getCharacterEncoding());
request.setCharacterEncoding("UTF-8");
Log.warning(request.getCharacterEncoding());
Log.warning(request.getParameter("x"));
```

Which gives us something more sensible:

```
UTF-8
null
UTF-8
```

Lesson of the day?

**You can't change the request encoding after request.getParameter() has been called.**

In my case a third-party filter was causing the damage, so I made a new filter at the top of the chain just to call `request.setCharacterEncoding()`. Note also that the default character encoding which I set with `-Dfile.encoding=UTF-8` had no effect on the decoding of request parameters. Also, the form is POSTed with `multipart/form-data` and I added `accept-charset` to force the browser to send UTF-8:

```
<form method="post" enctype="multipart/form-data" accept-charset="UTF-8" >
  <input type="text" name="x"/>
  <input type="submit"/>
</form>
```

Here is the filter code for the lazy folks.

```
/**
 * This filter ensures that all requests and responses are encoded and
 * decoded with UTF-8. It must appear first in the filter chain, because
 * as soon as you call request.getParameter(), the request encoding cannot
 * be changed.
 */
public class UTF8Filter implements Filter {

    @Override
    public void init(FilterConfig fc) throws ServletException {}

    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        chain.doFilter(request, response);
    }

    @Override
    public void destroy() {}
}
```

Okay, and the `web.xml` config for the really lazy folks.

```
<!-- Set UTF-8 before request.getParameter() is called -->
<filter>
  <filter-name>UTF8Filter</filter-name>
```

```
<filter-class>au.com.ninthavenue.web.webcore.util.UTF8Filter</filter-class>
</filter>
<filter-mapping>
  <filter-name>UTF8Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

## About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world.

Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.