

# Obfuscating urls

[Bookmarkable Link](#)

## Table of contents

- [How to obfuscate/encrypt a wicket url](#)
  - [Wicket 1.3](#)
  - [Wicket 1.2](#)
  - [Wicket 1.1](#)

## How to obfuscate/encrypt a wicket url

From time to time users ask how to obfuscate wicket urls. Instead of `myApp?component=1&version=0&interface=IRedirectListener` they asked for `myApp?sdf897sD879ddfD8...` and `myApp/sdf897sD879ddfD8` and many more. Due to varying requirements such as being Google and/or cluster compliant we decided to provide "hooks" build into the core to allow for virtually any obfuscating algorithm to be implemented by wicket users. Hopefully users will contribute their implementations back to the project.

Classes involved in encrypting and decrypting URLs are `WebResponse` and `WebRequest`. The default implementations provided by Wicket don't encrypt the URL at all, but subclasses (currently provided by core as well) like `WebResponseWithCryptedUrl` and `WebRequestWithCryptedUrl` do. In order for your application to use them you must subclass `WebApplication.newWebRequest()` and `WebApplication.newWebResponse()` like in the snippet shown below.

Note, this changed slightly in Wicket 1.2 and 1.3, as can be seen by comparing the fragments below:

### Wicket 1.3

#### Wicket 1.3+

```
protected IRequestCycleProcessor newRequestCycleProcessor()
{
    return new WebRequestCycleProcessor()
    {
        protected IRequestCodingStrategy newRequestCodingStrategy()
        {
            return new CryptedUrlWebRequestCodingStrategy(new WebRequestCodingStrategy());
        }
    };
}
```

The [Jasypt](#) (Java Simplified Encryption) framework has some Wicket-specific support for this functionality. See <http://www.jasypt.org/wicket.html> for more info.

### Wicket 1.2

#### Wicket 1.2+

```
public final class SignIn2Application extends WicketExampleApplication
{
    ....
    protected IRequestCycleProcessor newRequestCycleProcessor()
    {
        return new CompoundRequestCycleProcessor(new CryptedUrlWebRequestCodingStrategy(
            new WebRequestCodingStrategy()), null, null, null, null);
    }
}
```

### Wicket 1.1

## Wicket 1.1

```
public final class SignIn2Application extends WicketExampleApplication
{
    ....
    /**
     * @see wicket.protocol.http.WebApplication#newWebRequest(javax.servlet.http.HttpServletRequest)
     */
    protected WebRequest newWebRequest(HttpServletRequest servletRequest)
    {
        return new WebRequestWithCryptedUrl(servletRequest);
    }

    /**
     * @see wicket.protocol.http.WebApplication#newWebResponse(javax.servlet.http.HttpServletResponse)
     */
    protected WebResponse newWebResponse(HttpServletResponse servletResponse) throws IOException
    {
        return new WebResponseWithCryptedUrl(servletResponse);
    }
}
```