Annotations - Default Bean Names - The Special Case

Annotations - Default Bean Names ... and the Special Case

In general, when using Annotations, for the default bean name, Spring uses the following rule.

If the annotation's value doesn't indicate a bean name, an appropriate name will be built based on the short name of the class (with the first letter lower-cased).

For example:

HappyFortuneService --> happyFortuneService

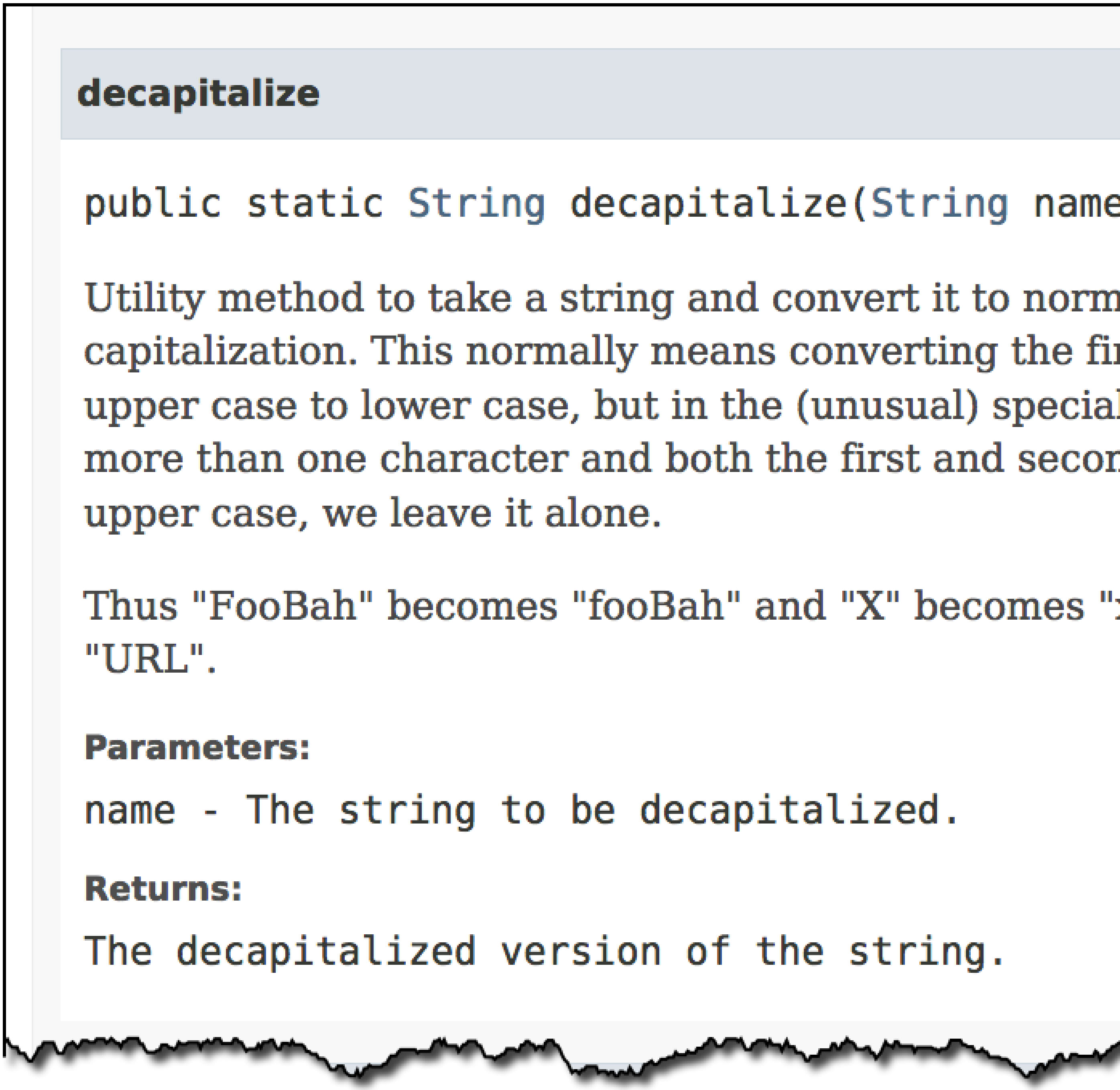
However, for the **special case of when BOTH the first and second characters of the class name are upper case**, then the **name is NOT converted**.

For the case of RESTFortuneService

RESTFortuneService --> RESTFortuneService

No conversion since the first two characters are upper case.

Behind the scenes, Spring uses the **Java Beans Introspector** to generate the default bean name. Here's a screenshot of the documentation for the key method.



decapitalize

```
public static String decapitalize(String name
```

Utility method to take a string and convert it to norm
capitalization. This normally means converting the fi
upper case to lower case, but in the (unusual) specia
more than one character and both the first and secon
upper case, we leave it alone.

Thus "FooBah" becomes "fooBah" and "X" becomes "
"URL".

Parameters:

name - The string to be decapitalized.

Returns:

The decapitalized version of the string.

Also, here's a link to the documentation.

- https://docs.oracle.com/javase/8/docs/api/java/beans/Introspector.html#decapitalize(java.lang.String)

As always, you can give explicity names to your beans.

```
@Component("foo")
public class RESTFortuneService .... {

}
```

Then you can access it using the name of "foo". Nothing tricky to worry about :-)

Hope this helps. Happy Coding! :-)