

# Tapestry5HowToNotMakeCustomComponent

I wanted to create a custom component and ended up doing few things the wrong way. Some will find these thing obvious, but I belive to some it will be usefull as well.

## beforeRenderTemplate / beforeRenderBody

Documentation clearly states that if a component doesn't have body, RenderBody phase is skipped, never the less I used that phase to create a custom component that outputs HTML by it self. What happened is that it worked when I used the component this way:

```
<t:MyComp source="listData"> </t:MyComp>
```

but it did not work when the space between tags was removed, or written as single tag

```
<t:MyComp source="listData"/>
```

If you don't intend to use the template supplied to your component's body (between tags), the proper place for your code is inside beforeRenderTemplate.

*HINT: you can use @BeforeRenderTemplate, @BeforeRenderBody annotations instead method names*

## non component classes/interfaces in myApp.pages and myApp.components and myApp.base packages

pages and components are instrumented by tapestry, they are also in a separate class loader, and you should avoid using their packages for any non component classes/interfaces. You are most likely to get **ClassCastException** or **CoercionException**.

For me these problems were result of trying to make a component as fast as I can. I was eager to use the class reloading feature, and was also lazy to separate an interface to a top level class, so I created an interface for a ListModel in my component and implemented it as inner class of the test page. I was perplexed by the exceptions that were thrown, and at first thought that the cause was in me using generics. At this time I can not in detail explain why these errors occur, but to recommend avoiding this situation.