

Properties

Table of contents

1 Properties.....	2
2 ModuleGenerator.....	2

1 Properties

There are various ways to configure forrest with your custom properties. Since version 0.9 the preferred method is the `forrest.properties.xml` file.

We need to prevent having references from the core to any plugin. This can be done with the new properties system. You can override any of the following values (example based on pdf plugin) or add as much new once via a `forrest.properties.xml` in your project.

```
<property name="output.pdf.fontFamily.serif" value="serif" />
<property name="output.pdf.fontFamily.sansSerif" value="sans-serif" />
<property name="output.pdf.fontFamily.monospace" value="monospace" />
```

Implementing this properties in your project allows you to change the font style of the pdf plugin.

FIXME (forrest-dev):

There have been a proposal about the naming convention about the properties system. Add the outcome to this document.

2 ModuleGenerator

Produces properties from an input-module. This is been managed by the the ModuleGenerator. The generator contacts the input module and gets an iterator of all attribute names. This iterator will be then used to receive the values of the key. Till now we only use a Object to String cast but an input module does not have to return a simple String but can return any object. This may cause `ClassCastException`s for the values.

- [Properties of the project input module](#) - used in the sitemap like {properties:content.xdocs}
- [Properties of the date input module](#) - we cast an object to a string, but the module returns `java.util.Date`. This causes a `ClassCastException`.