

Using project sitemaps

Table of contents

| | |
|---------------------------------------|---|
| 1 Introduction..... | 2 |
| 2 How does it work?..... | 2 |
| 3 Example uses of this technique..... | 2 |
| 3.1 Adding a new content type..... | 2 |

1 Introduction

After Forrest 0.6 it is now possible for projects to intercept the core sitemaps, without needing to copy the main sitemaps and keep them synchronised. This will enable hassle-free update to future Forrest versions.

Note:

We advise you to spend time to understand the Apache Cocoon sitemap. See [Cocoon sitemap](#) and [Cocoon concepts](#) and related component documentation. The Forrest sitemap is broken into multiple files. The main one is **sitemap.xmap** which delegates to others. See the [Sitemap Reference](#) for a tour of the default sitemap.

2 How does it work?

If a project has a `sitemap.xmap` file in its documentation dir, that gets mounted automatically by Forrest and becomes part of the processing: it is a preprocessing step, and is the first one to handle the request. Because of this it can serve any file directly. If it does not want to serve a file, it can simply not match the URL and Forrest will take care of it as usual.

The cool thing is that if that pipeline serves an xml representation, Forrest will provide a skinned version of it.

So if the project sitemap matches `test.xml` and transforms that to a correctly structured Forrest intermediate "document-v*", then the user will see `test.html` fully rendered by Forrest.

Of course, to resolve the directories in your sitemap it is important to use the 'project:' and 'forrest:' variables to prevent any possible issue in the future.

3 Example uses of this technique

3.1 Adding a new content type

See the section "Advanced customizations: sitemap.xmap" in the [Using Forrest](#) document and then follow the [Example: Adding a new content type](#).