

# Apache Forrest Dokumentation - Konfiguration von Forrest

## Table of contents

1 Konfiguration von Forrest.....	2
2 status.xml.....	2
3 forrest.properties.....	2
4 skinconf.xml.....	2
5 index.xml.....	3
6 site.xml.....	4
7 tabs.xml.....	4
8 Content.....	5
9 Modifizierung der Skins.....	5

## 1. Konfiguration von Forrest

Nachdem ein Projekt erfolgreich erzeugt wurde, kann mit der Konfiguration des Projektes begonnen werden. Dazu können die nachfolgend vorgestellten Dateien editiert werden, welche sich alle innerhalb des erzeugten Projektes befinden.

## 2. status.xml

Die Datei status.xml befindet sich im Root-Ordner des Projektes. Zweck dieser Datei ist es, Informationen über den Status des Projektes bzw. die Entwickler bereitzustellen. Die am Projekt beteiligten Personen werden im Element developers mit Namen, E-Mail-Adresse und einer ID aufgeführt. Weitere mögliche Angaben sind vorgenommene Änderungen im Element changes sowie eine TODO-Liste im Element todo.

## 3. forrest.properties

Die zentrale Konfigurationsdatei für jedes Projekt ist die Datei forrest.properties, welche sich im Root-Ordner befindet. In dieser Datei können die Einstellungen, die Forrest beim Erzeugen der Website verwendet, Validierungs-Einstellungen sowie allgemeine Pfad-Einstellungen, vorgenommen werden. Alle Einstellungen in der Datei sind im Ur-Zustand als Kommentare vorhanden, d.h. Forrest verwendet die Standardeinstellungen.

Soll nun ein Parameter geändert werden, muss dieser zunächst auskommentiert werden, bevor die Änderung aktiv wird. Soll beispielsweise der Projektname geändert werden, muss zunächst das Kommentarzeichen in der Zeile `#project.name=my-project` entfernt werden und dann der neue Projektname eingetragen werden: `project.name=xml-seminar`

Um ein anderes Skin als das Standard-Skin "pelt" für eine Projekt zu nutzen, kann der Parameter `#project.skin=pelt` modifiziert werden. `project.skin=tigris`

In diesem Beispiel wird nun das Skin "tigris" verwendet, welches sich im Hinblick auf Layout und Design vom Standard-Skin unterscheidet. Die Konfiguration der weiteren Parameter erfolgt analog zu obigem Beispiel. Wichtige Parameter sind:

- `xdocs-dir`: Der Ordner für die XML-Quelldateien
- `content-dir`: Der Root-Ordner für alle Inhalte
- `images-dir`: Der Ordner für alle Grafiken
- `skins-dir`: Ort eines eventuell modifizierten Skins
- `skinconf`: Skin-Konfigurationsdatei

## 4. skinconf.xml

In der Datei skinconf.xml, welche sich im Ordner `"/src/documentation"` befindet, wird die

Anzeige von grafischen Elementen und die Farbwerte gesteuert, sowie allgemeine Einstellungen wie beispielsweise das Copyright der Website vorgenommen. Über die disable-Elemente werden die angezeigten Icons wie beispielsweise das Icon zum Öffnen des aktuellen Inhalts als PDF-Dokument: false false false Obige Einstellungen ergeben beispielsweise folgende Anzeige auf jeder Website: Eine weitere Möglichkeit innerhalb der skinconf.xml ist die Angabe eines Projekt- bzw. Gruppenlogos. Neben dem Namen kann eine Beschreibung des Logos sowie ein Verweis angegeben werden. Eine korrekte Angabe des Pfades ist für die Anzeige des Logos Voraussetzung. Die Anzeige eines Favicon - sofern vom Browser unterstützt - ist über das Element favicon-url möglich. Die Copyright-Angaben werden über die Elemente year und vendor eingestellt: Über das feedback-Element kann eine E-Mail-Adresse definiert werden, die auf allen Seiten im Fuß-Bereich sichtbar ist. Das Attribut "to" ist für den angezeigten Namen zuständig, während das Attribut "href" die eigentliche E-Mail-Adresse definiert: Die Farbwerte können ebenfalls in der skinconfig.xml editiert werden. Dazu musst zunächst ein gewünschter Skinname (z.B. Forrest) auskommentiert werden. Danach können die Farbwerte mithilfe des RGB-Farbschemas angepasst werden: Das Attribut "name" definiert den Namen des Bereiches für den der Farbwert gültig ist. Das Attribut "value" gibt den RGB-Farbwert an.

### 5. index.xml

Die Datei index.xml befindet sich im Ordner "src\documentation\content\xdocs". Diese Datei ist als Vorlage für den reinen Inhalt gedacht. Besteht das Projekt beispielsweise aus mehreren Ordnern, kann diese Datei als Vorlage in den entsprechenden Ordner kopiert werden und danach angepasst werden. Die Datei beginnt mit dem Element document. Der Titel des Dokuments wird über das Element title innerhalb des Elements header definiert.

Der sichtbare Bereich des Dokuments beginnt wie in HTML mit dem body-Element. Innerhalb dieses Elements können nun die gängigen HTML-Elemente wie beispielsweise strong, table, tr oder td vorkommen. Wenn Elemente in der DTD fehlen, können diese aufgenommen werden. Neben diesen gängigen HTML-Elementen gibt es die Möglichkeit spezifische Forrest-Elemente zu verwenden. Das wichtigste Element ist hierbei das section-Element, mit dem ein separates Kapitel definiert wird. Der Name eines solchen Kapitels wird über das title-Element definiert: Das besondere an dem section-Element ist die automatische Generierung einer Sitemap. Diese Sitemap ist im Dokumentenanfang sichtbar, enthält den Titel des Kapitels und verlinkt per Sprungmarke auf das Kapitel. Nachfolgende Abbildung verdeutlicht den Einsatz des section-Elements. Der Einsatz dieses Elementes ist vor allem bei sehr umfangreichen Dokumenten mit vielen, einzelnen Kapiteln sinnvoll. Weitere hilfreiche Elemente sind beispielsweise note, source und warning, mit deren Hilfe sich Informationen, Quellcode und Warnungen besonders hervorheben lassen:

Note:

Informationen

Quellcode

**Warning:**

Warnung

## 6. site.xml

In der Datei site.xml (befindet sich in "src\documentation\content\xdocs"), der wichtigsten Konfigurationsdatei innerhalb von Forrest, werden alle Seiten innerhalb des Projekts aufgenommen. Dies ist zwingend erforderlich, da sonst Verlinkungen nicht funktionieren. Aus der site.xml wird auch das linke Navigationsmenü generiert. Eine weitere Funktion der site.xml ist das Bereitstellen von externen Links. Die site.xml beginnt mit dem Element site. Innerhalb dieses Elements können beliebig viele Child-Elemente vorkommen, welche wiederum Child-Elemente enthalten können. Forrest erstellt anhand dieser Hierarchie eine Menüstruktur, die das einfache Navigieren innerhalb des Projektes ermöglichen soll.

Über das Attribut "href" wird der Pfad bzw. der Dateiname angegeben. Forrest erkennt automatisch bei einer Angabe von "seite.html", dass diese HTML-Datei mithilfe der XML-Datei "seite.xml" generiert wird. Im obigen Beispiel befinden sich die Vorlesungen im Unterordner "lehre/", was über das Parent-Element vorlesungen, ebenfalls über das Attribut "href", angegeben wird. Alle Child-Elemente von vorlesungen erben dann den entsprechenden Pfad. Die Attribute "label" und "description" werden verwendet um das linke Menü zu generieren, wobei "label" dem sichtbaren Text und "description" dem beschreibenden Text, der erscheint wenn sich der Mauszeiger über dem sichtbaren Text befindet, entspricht (vergleichbar dem HTML-title-Attribut).

Ein besonderes Feature von Forrest ist die Möglichkeit, Kurzformen für interne Links und auch für externe Links zu verwenden. Interne Links lassen sich über die Kurzform "site:sitename" darstellen. Dies gilt für alle in der site.xml eingetragenen Seiten. Soll beispielsweise auf die Seite "forrest.html" verwiesen werden, genügt folgende Kurzform: [Forrest](#) ( ../forrest/ ) Forrest erkennt automatisch, dass damit die Seite "forrest.html" gemeint ist, unabhängig von wo aus der Link gesetzt wird. Dies erspart bei umfangreichen Dokumentationen mit vielen internen Links eine Menge Zeit und Arbeit. Über interne Pfade muss sich der Autor eines Projektes keine Gedanken mehr machen. Die Kurzformen für externe Verlinkungen funktionieren ähnlich.

## 7. tabs.xml

Die Datei tabs.xml befindet sich in "src\documentation\content\xdocs" und ist für die

Generierung der Tabs (Reiter) zuständig, welche sich horizontal oberhalb der Inhaltsseiten erstrecken (vgl. auch 2.1.4). Die Datei beginnt immer mit dem Element tabs. Innerhalb dieses Elements werden die einzelnen Tabs anhand der Attribute "id", "label" und "dir" definiert. Das Attribut "id" ist notwendig um in der site.xml (vgl. 1.4.5) Menüeinträge einem Tab zuzuordnen. In der site.xml wird dann die entsprechende ID verwendet. Das Attribut "label" wird, wie in Forrest üblich, für den sichtbaren Text verwendet, während das Attribut "dir" den Ordner für das Tab angibt. Wird "dir" verwendet, ist es erforderlich eine Startdatei, das sogenannte "indexfile" anzugeben

## **8. Content**

Zusätzlicher Content wie beispielsweise gepackte Dateien lässt sich im Ordner "src\documentation\content\xdocs" unterbringen. Es ist sinnvoll z.B. einen Unterordner "Downloads" anzulegen. Für Grafiken kann der bereits von Forrest angelegte Ordner "images" verwendet werden. Soll der Zugriff auf diese Dateien über das generierte Menü möglich sein, ist es erforderlich die entsprechenden Pfade in der Datei site.xml anzugeben.

## **9. Modifizierung der Skins**

Neben der Datei skinconf.xml, lässt sich das Design und die Anzeige bestimmter Elemente auch über sogenannte Skins modifizieren. Forrest liefert einen Satz von Skins bereits mit. Das Standard-Skin "pelt" kann in der Datei forrest.properties geändert werden. Wichtig ist hierbei, dass Forrest nicht auf das angelegte Projekt zugreift um das Skin zu verwenden, sondern die Dateien im Root-Ordner von Forrest, im Pfad "\src\core\context\skins\pelt", verwendet. Es ist möglich diese Dateien direkt zu modifizieren, was den Nachteil hätte, dass die Änderungen auch für zukünftige Projekte Bestand haben. Sinnvoller ist es zunächst den Ordner des entsprechenden Skins in das lokale Projekt zu kopieren. Der Ordner "common" muss dabei ebenfalls kopiert werden. Als Verzeichnis im lokalen Projekt kann der Ordner "src\documentation\skins" angelegt werden. Damit Forrest weiß, wo sich das zu verwendende Skin befindet, müssen die Änderungen in der Datei forrest.properties bekannt gemacht werden.