

# Apache Lenya

## Features

Mit einem Asterisk (\*) markierte Features wurden in Version 2.0 hinzugefügt.

### Inhaltsmodell und Repository

- Dokumente werden im Dateisystem abgelegt
- flache Dokumentenstruktur \*
- Sprechende, nicht-kryptische URLs
- beliebige Anzahl von Sprachversionen
- konfigurierbare Sets von Meta-Daten mit einem oder mehreren Werten pro Element \*
- beliebige Inhaltsformate (XML, Text, binär)
- Zugriff auf das Repository via WebDAV

### XML-Unterstützung

- Validierung (Relax NG, XSD)
- flexible Pipelines zur XML-Verarbeitung (Apache Cocoon)
- XSLT, XInclude, XPath, SAX, DOM
- XML kann aus vielfältigen Quellen generiert werden, z.B.:
  - XML: Daten aus externen RSS-Feeds einbinden
  - HTML: Page-Scraping, Legacy-Websites einbinden
  - SQL: Daten aus Datenbanken einbinden
  - Text: Text-Dateien parsen und daraus XML-Daten generieren

### Website-Management

- mehrere Websites („Publikationen“) per Lenya-Instanz
- Neue Websites können über das GUI hinzugefügt werden \*
- Publication-Templating: \*
  - Publikationen werden auf der Basis einer Master-Publikation erstellt
  - Eigenschaften können überschrieben werden (z.B. XSLT oder CSS)
  - Publikationsbaum mit beliebiger Tiefe
  - Benutzerkonten können gemeinsam oder nach Publikation getrennt verwaltet werden
- eigene Proxy-Konfiguration pro Publikation
- Cut/Copy/Paste von Teilbäumen im Site-Tree
- Übersicht über Dokumente nach Sprache, Ressourcen-Typ und Workflow-Zustand \*

### Link-Management

- Dokumente werden über UUIDs (Universally Unique Identifier) referenziert \*
- Funktion zum Finden aller internen Links, deren Ziel nicht existiert \*
- Warnung, falls ein zu publizierendes Dokument Links zu unveröffentlichten Dokumenten enthält

### Inhaltstypen

- beliebige XML-Strukturen mit Validierung
- Binärdateien und Bilder werden als Dokumente gehandhabt \*
- Bild-Upload in WYSIWYG- und Quelltext-Editoren
- OpenDocument-Dokumenttyp wird mitgeliefert

## Suchmaschine

- basierend auf Apache Lucene
- inkrementelle Indexierung nach jeder Änderung
- flexible Indexierung und Queries von XML- und Metadaten
- PDF-Indexierung \*

## Bearbeitung der Inhalte

- WYSIWYG-Editoren
  - BXE (XML, Firefox, Relax NG)
  - Kupu (XHTML, Firefox, IE)
  - FCKEditor (XHTML, Firefox, IE, Safari, Opera, ...)
  - TinyMCE (XHTML, Firefox, IE, Safari, Opera, ...)
  - Optional: Xopus (XML, IE, XSD)
- Quelltext-Editoren
  - XML-Quelltext-Editor, Syntax-Highlighting mit CodePress (<http://www.codepress.org>) kann installiert werden
  - formularbasierter Editor („Form-Wizard“)
- Editoren für spezielle Dokumenttypen können mit dem Usecase-Framework implementiert werden \*

## Layout

- flexible Pipelines zur XML-Verarbeitung: Transformation, Filtern, Aggregation, Inklusion, ...
- XSLT für XML-, (X)HTML- und Text-Rendering
- CSS zur optischen Gestaltung
- XSL-FO für dynamische PDF-Generierung
- Unterstützung für anwendungsspezifische Templating-Mechanismen
- Framework zur Implementierung von Navigationselementen
- Komponenten und XSLTs zur SVG-Verarbeitung \*
- serverseitige Skalierung von Bildern \*
- automatisch generierte Layout-Bilder, z.B. abgerundete Ecken \*

## Workflow

- beliebige Workflow-Deklarationen
- Modell: Zustandsautomat (deterministischer endlicher Automat)
- vordefinierter Workflow als Beispiel: bearbeiten, zur Prüfung einreichen, zurückweisen, publizieren, deaktivieren
- Dokumente können zwischen verschiedenen Bereichen verschoben werden, z.B. Bearbeitung, Staging, Live, Archiv, Papierkorb
- zeitgesteuerte Ausführung von Workflow-Übergängen
- Notifikation (E-Mail, Posteingang im CMS-GUI)

## Versionierung

- Sicherheitskopie der letzten  $n$  Revisionen
- Revisionen können angesehen und wiederhergestellt werden

## Benutzungsoberfläche

- Bearbeitungsumgebung: WYSIWYG
- Standard: Pull-Down-Menü am oberen Seitenrand
- inaktive Menüpunkte sind deaktiviert, mit Tooltips zur Erklärung
- beliebig anpassbar (XSLT, CSS)
- Sitetree: Javascript, inkrementell geladen für bessere Skalierbarkeit

## Zugriffsschutz

- flexible rollenbasierte Policies pro Seite, Usecase und Workflow-Übergang
- Rollen können Nutzern, Gruppen und IP-Bereichen zugewiesen werden
- SSL-Verschlüsselung kann pro Seite bzw. URL-Bereich verlangt werden
- Optionen zur Authentifizierung:
  - Verwaltung der Benutzerkonten im CMS
  - LDAP
  - CIFS
  - Single Sign-On über Shibboleth (in Arbeit)
  - Integration anwendungsspezifischer Authentifizierungsmechanismen ist möglich

## Entwicklung

- erweiterbar: eigene Module, Ressourcen-Typen etc. können hinzugefügt werden
- mächtige Java-API für Dokumentenmanagement und XML-Verarbeitung mit SAX und DOM
- Verarbeitungskomponenten können in Java oder XSLT implementiert werden
- Usecase-Framework zur einfachen Implementation interaktiver Funktionen
- Cocoon Flow mit Continuations zur Verwaltung des Sitzungszustands
- eingebaute Profiling-Möglichkeiten \*
- einfache Integration von JUnit-Tests und Canoo-Webtests \*