

Autor: Martin Fechner
Eingesetzte Software: exist-db

Ziel: exist-db Setup

Dokumentation der Benutzung von exist-db für Entwickler von ediarum-Projekten.

Zusammenfassung

Einige Punkte im ersten Entwurf.

Inhalt

exist-db.....	2
1. Projektmigration.....	2
2. Verzeichnisstruktur.....	2
/opt/server/exist_cagb.....	2
3. ediarum Synchronisationsworkflow.....	2
4. Nutzer.....	2

exist-db

1. Projektmigration

Schritte zur Migration eines Projektes aus einer exist-db-Instanz (2.1, 2.2, 3.0_RC) in eine andere (3.1):

Vorbedingung: Die App ediarum.EXIST muss auf der Version 3.1 laufen.

1. App installieren und einrichten (Passwörter ändern, Controller aktivieren, Port ändern)
2. Projekt in der App einrichten
3. Nutzer einrichten
4. Datenverzeichnis kopieren
5. Projektdateien inkl. Webseite einspielen und Funktionsfähigkeit der Skripte testen
6. Registerverbindungen prüfen
7. Synchronisationen prüfen

2. Verzeichnisstruktur

Bsp.:

/opt/server/exist_cagb

Ist das Verzeichnis der eXist-Instanz mit entsprechendem Namen (hier: exist_cagb; evtl. ist der Versionsname aussagekräftiger, wenn mehrere Projekte in der DB liegen). Darin liegen drei Verzeichnisse:

- **backup/:** Verzeichnis mit den Backup-Dateien. Konfiguration in /existb_db/conf.xml unter <scheduler>
- **data/:** Das Verzeichnis mit der Datenbank. Wird dieses gelöscht und die Datenbank neu gestartet, wird der Inhalt neu (und leer) aufgebaut. Beachte: Es wird dann ein ‚admin‘ Benutzeraccount angelegt mit leerem Passwort.
- **existb_db/:** Die Installation der Datenbank selbst. Zur Konfiguration wichtig ist die ‚conf.xml‘.
- **existb-db-setup-2.2.jar:** Die benutzte Installationsdatei der Datenbank.

3. ediarum Synchronisationsworkflow

Es gibt drei (bzw. vier) Instanzen plus das **GIT**, in welchem die Entwicklung stattfindet und gespeichert wird:

- **LOCAL:** Lokale Entwicklungsinstanz
- **DEV:** Die Entwicklungsinstanz zum Entwickeln und Testen von Features auf dem Server.
- **EDIT:** Die Produktivinstanz, auf der das fertig aktuelle publizierte System läuft und auf der die Vorhaben arbeiten, inkl. internen Webseite.
- **PUBLIC:** Die öffentliche Instanz, die die externe Webseite ausliefert.

Der Synchronisationsworkflow von ediarum sieht etwa folgendermaßen aus:

- **GIT:** Definiert ANT-Skript, die das Ausspielen der relevanten Projektdateien inkl. Webseite exkl. der Daten (/data, /archive, /internal_data, /external_data) nach LOCAL, DEV, EDIT und PUBLIC erlaubt.
- **LOCAL:** Synch definiert, der sich die aktuellen Daten (/data) von EDIT holt. Manuell zu starten.
- **DEV:** Synch definiert, der sich die aktuellen Daten (/data) von EDIT holt. Manuell zu starten.
- **EDIT:** Definiert Synch, die die aktuellen Daten nach PUBLIC schreibt. Tägliche synchronisation.
- **PUBLIC:** Keine Synch definiert

4. Nutzer

In eXist und Oxygen sind problemlos E-Mail-Adressen als Nutzernamen möglich.