

Module der ExpoDB

Die ExpoDB gliedert sich in einzelne Module, die aufeinander aufbauen. Entsprechend der Bedarfe der Museen werden die Module z.T. zeitlich parallel und z.T. nacheinander realisiert. Dabei wird eine Vorgehensweise der agilen Softwareentwicklung gewählt, nach der früh Prototypen bereitgestellt werden, die entsprechend des Feedbacks der Nutzer anschließend verfeinert werden. Entsprechend sind einzelne Module noch vollständig zurückgestellt, weil sie erst nachgelagert benötigt werden.



expo.trafo

Datenverarbeitungen in der ExpoDB werden in deren zentralen Modul expo.trafo durchgeführt, das bereits rege im Einsatz ist. Um die vielfältigen Bedarfe von Import-, Export-, Datenablage- und Pflegekomponenten zu bedienen, ist dieses Modul besonders leistungsfähig und flexibel konfigurierbar. [weiterlesen](#)

expo.sync

Das Modul expo.sync exportiert Daten aus *imdas pro* in die ExpoDB. Da dies die Voraussetzung der Bereitstellung solcher Daten ist, wurde expo.sync als erstes realisiert und wird bereits allnächtlich eingesetzt. Dazu werden über ein *PL/SQL-Skript* nicht sensible Daten aus den *imdas pro* unterlegten Datenbanken als XML zur Abspeicherung in expo.storage transferiert. [weiterlesen](#)

expo.digest

Das Modul expo.digest bezeichnet die Online-REST-Schnittstelle für Daten aus der ExpoDB. Sie wurde bereits prototypisch realisiert und definiert zum einen eine Anfragesyntax, welche Selektionskriterien, Sortierung, Paginierung als "Demonstrator" und Format einer Datenbereitstellung festlegt, zum anderen wertet sie Anfragen auf der Datenablage aus und transformiert die Ergebnisse mit expo.trafo in das gewünschte Format. [weiterlesen](#)

expo.delivery

Das Modul expo.delivery dient zur Auslieferung von Medien zu den Datensätzen, in der Regel digitale Bilder, über eine Online-Schnittstelle. Die Parametrisierung ermöglicht die Identifizierung des gewünschten Bildes sowie die Festlegung der Auflösung. Mit sehr ähnlicher Parametrisierung war das Modul bislang für die Digitalen Kataloge der Museen in Verwendung, so dass es für Bilder und PDF-Dateien bereits zur Verfügung steht. [weiterlesen](#)

expo.storage

Die Datenvorhaltung in der ExpoDB wird derzeit über die Suchmaschinentechologie Apache-Solr realisiert. Für den Bestand jedes Museums wird darin ein sog. Core realisiert, der als Index zur Identifizierung die *imdas-pro*-ID, ggf. wiederholbare Indexe für alle suchbaren Felder, für Sortierungen und für die Facetten sowie den Datensatz im Gesamten als XML-String enthält, wie dieser aus *imdas-pro* übertragen wurde. [weiterlesen](#)

expo.registry

Die Flexibilität der ExpoDB wird zu einer Fülle von Beschreibungen von Formaten, Datenverarbeitungen, Workflows etc. führen. expo.registry ermöglicht den Online-Zugriff, um solche Konfigurationen zu verwalten und die Pflege von vorhandenen und das Hinzufügen von neuen Konfigurationen zu ermöglichen. Dieses Modul ist bislang noch nicht implementiert, sondern behilft sich zur Organisation von Pipeline-Definitionen etc. mit Dateiodnern; lediglich ein Upload von zusätzlichen XSL-Skripten für expo.digest ist bereits möglich. [weiterlesen](#)

expo.notify

Die ExpoDB beinhaltet viele Prozesse, die regelmäßig ausgelöst oder über Maschinen-Maschine-Schnittstellen angefordert im Hintergrund ohne bewußtes Zutun stattfinden. Um solche Vorgänge, z.B. die regelmäßige Aktualisierung aus *imdas pro* überwachen zu können, ist eine Komponente erforderlich die Log-Informationen sammelt und Mechanismen zur Benachrichtigung von Zuständigen besitzt. Dieses Modul ist noch nicht realisiert; bislang werden automatisch lediglich die täglichen Datenlieferungen in die ExpoDB per E-Mail mitgeteilt. [weiterlesen](#)

expo.dashboard

Das Modul expo.dashboard bezeichnet die gemeinsame Oberfläche der anderen Module, über die Museumsmitarbeiter Prozesse konfigurieren, auslösen und überwachen können sowie Statistiken abrufen etc.. SWBtrafo besitzt bereits eine entsprechende Oberfläche, über die Pipeline-Parameter eingestellt und z.B. XML-Daten zur Verarbeitung hochgeladen werden können. Aber auch hier ist noch Konzeptions- und Implementierungsarbeit zu leisten. [weiterlesen](#)

expo.curation

Bislang beinhaltet die ExpoDB nur Daten, die in *imdas pro* gepflegt werden. Perspektivisch soll die ExpoDB allerdings auch externe Daten z.B: aus Anwendungen zur Nutzerpartizipation aufbewahren. Für solche Daten benötigt man eine Pflegekomponente. Bislang ist allerdings dieser gesamte Bereich noch nicht einmal konzipiert, so dass zunächst genauer Bedarfe erhoben werden müssten. [weiterlesen](#)

expo.ingest

Über die ExpoDB sollen auch Daten, die extern, sei es durch Nutzerpartizipation oder in weiteren Anwendungen entstanden sind, nach *imdas pro* übertragen werden. Dazu werden die Mechanismen von *imdas pro* verwendet, um die erforderlichen Konsistenzbedingungen der Datenbank bei Importen zu gewährleisten. Eine erste Realisation solcher Vorgänge wird derzeit für Daten des Asset-Managementsystem Cumulus des Landesmuseums Württemberg implementiert. Insbesondere leistet expo.ingest dazu die Aufbereitung der Daten entsprechend Format-Anforderungen von *imdas pro*. [weiterlesen](#)

expo.archive

Das Backup der Oracle-Datenbanken, mit denen *imdas pro* arbeitet, wird mit den üblichen Aufbewahrungsfristen entsprechend dem Datensicherungskonzept vorgehalten. Für längere Fristen werden die Museumsdaten derzeit nicht aufbewahrt, da sie sich in *imdas pro* nicht mehr nutzen lassen würden, sobald sich dessen Datenbankschema geändert hat. Stattdessen wird überlegt, in der ExpoDB die XML-Daten zu archivieren, die auch ohne Datenbankmanagementsystem langfristig interpretierbar bleiben. [weiterlesen](#)