

## Export/ Import/ Löschen ins Labsystem integrieren

Es soll eine Export/ Importfunktion für das „Labssystem“ (<http://ilab.net.in.tum.de>) programmiert werden. Dazu sind die Elemente eines so genannten „Lab“ zu exportieren und die so gesicherten Daten später wieder in die Instanz aus der exportiert wurde oder einer auf einem anderen Rechner importiert werden.

0. Als erstes ist ein Textfile anzulegen, in dem die Zuordnung Elementkennung (ein Buchstabe) zu Tabellename konfiguriert wird. Eine zu sichernde Einheit setzt sich immer aus der Elementkennung sowie einem fortlaufenden einmaligen Index in der Tabelle zusammen. Daher ist es zweckmäßig, gleich auch diese Indexspalte in der Export/ Import-Konfiguration anzugeben:

Elementkennung	Tabelle.Index
p	pages.idx
c	collections.idx
m	
i	

Der Export soll in eine versionierbare Textdatei/ Ordner-Struktur erfolgen, die dann in ein CVS eingchecked werden soll. Im System werden Inhalte immer „hinten“ angefügt. Das heißt, ein neues Element eines Typs bekommt einen neuen Index in der entsprechenden Tabelle.

Beim Export sollen die Indizes nach einem deterministischen Schema umgeschrieben werden. Es soll jeweils mit 1 begonnen werden. Dies ist relevant, damit von verschiedenen Instanzen konsistent versioniert werden kann.

Beim Umschreiben muss Buch geführt werden, da die Indizes sowohl in der „Payload“ der einzelnen Elemente als Links als insbesondere auch im Aggregationselementen c (und auch l) auftauchen und dort umgeschrieben werden müssen. (Beim Export und später wieder beim Import!)

1. Es muss eine entsprechende Übersetzungstabelle für die Elemente in einem Lab erstellt werden, die obigen Anforderungen genügt.

Ein Lab besteht beispielsweise aus Folgendem:

(Zur Semantik: ein großes C bedeutet: „stelle alle Elemente jeweils als eine Seite dar“; ein kleines c bedeutet: „stelle alle enthaltenen Elemente als eine Seite dar.)

- a. PreLab:

p5

C6 ( p6 c2 ( p7 m2 ) c3 ( p8 m3 m4 m5 ) c4 ( p9 m6 ) c5 ( p10 m7 ) )

C11 ( c7 ( p11 m8 m9 ) c8 ( p12 m10 ) c9 ( p13 m11 ) c10 ( p14 m12 ) )

C13 ( p15 p16 p17 c12 ( p18 m13 m14 ) ) c14 ( p19 m15 )

C19 ( c16 ( p20 m16 ) c17 ( p21 m17 ) )

C21 ( p23 p24 c20 ( p25 m19 ) p26 p27 p28 )

C22 ( p29 ) c26 ( p30 i19 )

- b. Lab:  
c24 ( p149 p31 i3 i4 ) c27 ( p32 i5 ) c28 ( p33 i6 i7 ) c29 ( p34 i8 i9 ) c30 ( p35 i10 ) c31  
( p36 i11 i178 i177 ) c32 ( p37 i12 i13 i14 ) c33 ( p38 i15 p39 i16 i17 ) c34 ( p40 i18 )  
c26 ( p30 i19 ) p123
- 2. Anhand der Tabelle muss beim Export eine geeignete Umbenennung der Elemente erfolgen (remapping). Am besten ist wohl eine Ordnerstruktur wie folgt:

`[_UniqueID\{p, c, m, i}\{00000..99999}.txt`

So sollen nun die Daten exportiert werden (dabei „Payload“ auch umschreiben!!!).

- 3. Beim Export sollen entstehen:
  - a. PDF des Lab mit Lösungen
  - b. Export des Lab mit Lösungen als versionierbaren Text.
  - c. PDF des Lab ohne Lösungen
  - d. Export des Lab ohne Lösungen als versionierbaren Text.
- 4. Die Daten sollen direkt in ein CVS (bei uns) eingchecked werden können.
- 5. Es soll in der GUI die Option bestehen, das Lab nach dem Export zu entfernen. (Dazu müssen eventuell noch bestehende Abhängigkeiten [garbage collection; Element sonst noch wo referenziert?] geprüft werden.
- 6. Für Export und Import soll eine GUI-Seite im System erstellt werden.
  - a. Diese GUI-Seite soll bei möglicher Verbindung zum Server die installierten und installierbaren Labs anzeigen. (Dazu brauchen diese ggf. noch eine zu integrierende ID)

Beim Import muss ein umgekehrtes Rückmapping anhand der tatsächlichen Einfügeposition in der Instanz in die integriert wird, erfolgen.

Beim Löschen muss eine Abhängigkeitsanalyse erfolgen. Löschen soll unabhängig von Import/ Export möglich sein.