

BookLab

by RENware Software Systems

BookLab System

(c) 2023 RENware Software Systems

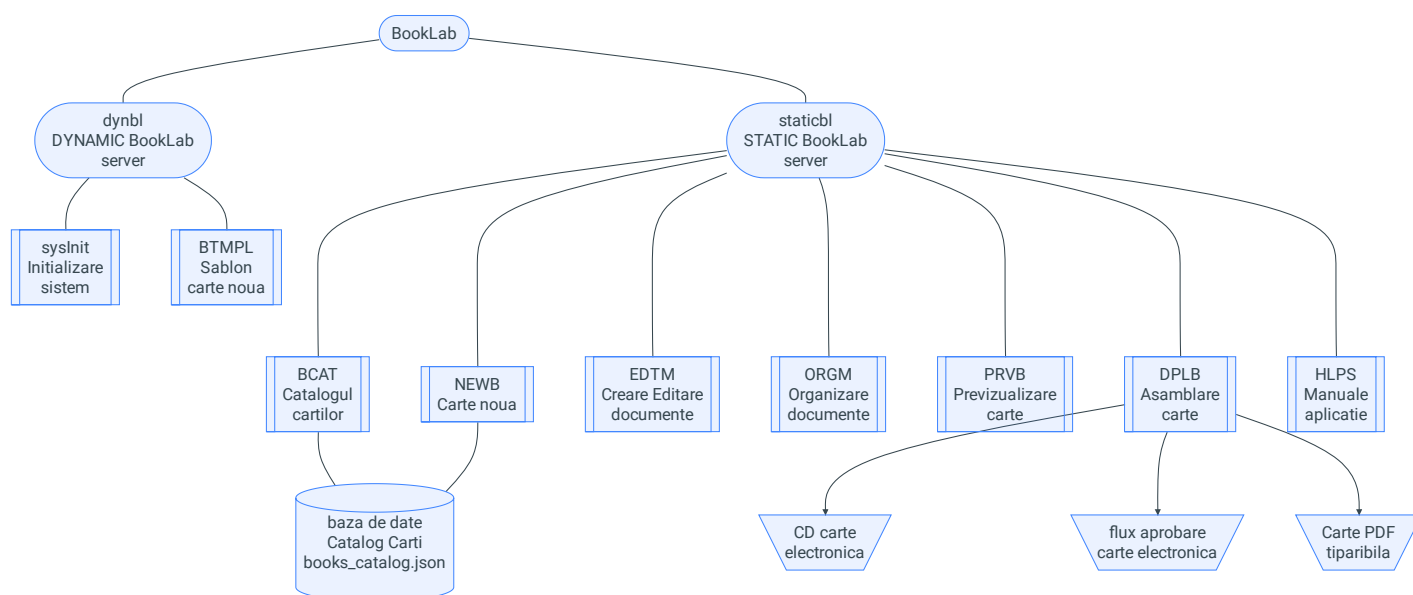
Arhitectura sistemului

Cuprins:

- Arhitectura sistemului
 - Introducere
 - sysInit Initializare sistem
 - BCAT catalogul cartilor
 - BTMPL sablon carte noua
 - NEWB creare carte noua
 - EDM preluare si editare continut
 - ORGM organizare structura carte
 - PRVB previzualizare si testare carte
 - DPLB asamblare publicare carte
 - HLPS help asistenta si manuale
 - Organizarea aplicatiei software
 - Note

Introducere

Arhitectura sistemului pleaca de la structura de baza a acestuia, structura redată "pe scurt" mai jos. Diagrama puțin mai elaborată dar cu descrierea componentelor se găsește în [documentul 130.02-Overview.md](#), secțiunea "Structura și componenta BookLab" ce se recomandă a fi citită înainte de a începe studiul acestuia, pentru o mai bună înțelegere a lui.



Referitor la *locatia fizica a componentelor* se face remarcă ca toate componentele sistemului ce prezintă interfața vizuală cu utilizatorul au reprezentare atît în directorul `docs/` pentru partea de *front-end* (client side) cît și în rădăcina întregului sistem pentru partea de *back-end* (server-side). (Directorul `docs/` este generat prin compilarea aplicației în faza de development din directorul `doc_src/.`)

Conceptele de server și client

- Se atrage atenția asupra tratării corecte din punct de vedere conceptual a noțiunilor de `server` și `client`. Arhitectura conceptuală de client-server este pe deplin implementată cu respectarea tuturor principiilor aferente și o consimizare a acestui lucru va permite o înțelegere corectă a detaliilor tehnice ale sistemului de către echipa tehnică de lucru.
- În general în toată documentația produsului prin termenul *CLIENT* se va înțelege și partea de *CLIENT FRONTEND* iar prin cea de *SERVER* și partea de *SERVER BACKEND*.
- Locațiile pentru *partea de server static* (componenta `staticbl`) sunt situate după o organizare clasică de site static începînd din directorul `docs/` iar codul "de start" este prezent în fișierul `index.html` (presupunînd că la utilizarea unui HTTP server anume la dorința clientului, acesta este configurat să funcționeze așa în lipsa specificării unui fișier în rutele URL).
- În general (dacă nu specifică în clar altfel) `code-name` al componentelor este același atît pentru SERVER cît și pentru CLIENT. Diferențierea între acestea se întîmplă deoarece răspund servere diferite pentru porturile aferente (80 client și 4111 server). Astfel pe ruta "normală" a unei componente, de ex. ptr *NEWB*: `http://booklab.../newb` va răspunde întîi componenta CLIENT (prezintă sub forma unui `index.html` în directorul `docs/newb/`).

sysInit Initializare sistem

Pornirea sistemului comportă două mari componente "on top", și anume:

- `staticbl` - **STATIC BookLab** serverul *HTTP static* acesta va deservi partea de web compusă din "fișiere" statice. Acestea reprezintă fișierele HTML aferente interfeței UI dar și celelalte fișiere "auxiliare" precum: CSS-uri, JS-uri, imagini, etc
- `dynblsrv` - **DYNAMIC BookLab** serverul *CGI dinamic* ce permite rularea de scripturi Python server-side și deservește componentele care execută operațiile necesare (de ex. create carte nouă). Acest server este cunoscut și sub numele de `BookLab API`

De asemenea referitor la initializarea sistemului trebuie spus și că **prima instalare a acestuia** se face cu o serie de programe / scripturi. Nu se intră în detalii deoarece procedura de instalare a sistemului este una cît se poate de clasică pentru instalările executate manual pe sisteme Linux iar pentru sisteme Windows acesta este tot una clasică, cu un simplu wizard "standardizat" și care execută scrierea în toate locațiile necesare (ex. Registry).

i parametrii

- *code-name:* `sysInit`
- *locatie DYNAMIC BookLab:* `sysInit/`
- *locatie script PRIMA INSTALARE:* `setup/`
- *Document descriere detaliata:* [810.05a-sysInit_System_Process.md](#)

BCAT catalogul cartilor

Acest modul este responsabil de catalogul (lista) cartilor si de sablonul de creare a unei carti. Astfel el contine urmatoarele elemente:

- **sablonul** implicit de creare a unei carti noi (si singurul in versiunea 1 a aplicatiei). Datele acestui sablon se afla in subdirectorul `bk_tmpl/`.
- **catalogul cartilor** ce este baza de date cu informatiile de baza despre carti. Acest catalog contine informatiile necesare pentru identificarea si referentierea unei carti in aplicatie. *Datele detaliata* aferente unei carti se gasesc in directorul aferent acesteia, director al carui nume se regaseste in acest catalog. Catalogul este stocat in format `JSON` si se gaseste in fisierul `books_catalog.json`.
- **interfata UI "Catalog carti"** ce este interfata CLIENT-UI¹ pentru managementul catalogului cartilor oferind operatii tip CRUD asupra bazei de date cu catalogul cartilor.

i parametrii

- *code-name:* `bcat`
- *locatie DYNAMIC BookLab:* `docs/data/` cu baza de date catalogul cartilor `books_catalog.json`
- *locatie STATIC BookLab:* `docs/bcat/index.html` (`doc_src/index.md` before 1-st compilation)
- *Document descriere detaliata:* [810.05a-bcat_System_Process.md](#)

BTMPL sablon carte noua

-#TODO...



parametrii

- *code-name*: `book_tmpl`
- *locatie DYNAMIC BookLab*: `docs/book_tmpl/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-btmpl_System_Process.md](#)

NEWB creare carte noua

-#TODO...

**i** parametrii

- *code-name*: newb
- *locatie DYNAMIC BookLab*: newb/
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-newb_System_Process.md](#)

EDTM preluare si editare continut

-#TODO...



parametrii

- *code-name*: `edtm`
- *locatie DYNAMIC BookLab*: `edtm/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-edtm_System_Process.md](#)

ORGM organizare structura carte

-#TODO...



parametrii

- *code-name*: `orgm`
- *locatie DYNAMIC BookLab*: `orgm/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-orgm_System_Process.md](#)

PRVB previzualizare si testare carte

-#TODO...



parametrii

- *code-name*: prvb
- *port for running HTTP preview server*: 8111 (DIFFERENT than 80 which is reserved for main application)
- *locatie DYNAMIC BookLab*: prvb/
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-prvb_System_Process.md](#)

DPLB asamblare publicare carte

-#TODO...



parametrii

- *code-name*: `dplb`
- *locatie DYNAMIC BookLab*: `dplb/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-dplb_System_Process.md](#)

HLPS help asistenta si manuale

Manuale aplicatiei sunt de doua feluri (categorii) majore:

- **Manuale de utilizare**, acestea prezinta informatii pentru utilizarea curenta a aplicatiei. Catalogul acestora se poate accesa de la <SERVER NAME>/help/880.30-EUMA_catalog.html.
- **Manuale de configurare**, acestea prezinta informatii pentru configurarea aplicatiei. Catalogul acestora se poate accesa de la <SERVER NAME>/help/880.30-ADMA_catalog.html.

Pentru ambele tipuri de manuale se va prezenta un *catalog* ce contine lista tuturor manualelor disponibile in acea categorie. Lista contine numele manualului, o scurta descriere (daca numele nu este suficient de elocvent) si linkul catre manualul respectiv.

Manualele sunt localizate in sectiunea statica a aplicatiei (vezi si sectiunea "[Serverul HTTP static](#)") in directorul `docs/help/`.

Organizarea aplicatiei software

Cod document metodologic: 810.05b Application Software Organization

In principiu aplicatia (sistemul) este organizata pe principiul *directoarelor de fisiere*. Aceasta permite o "rapida" regasire si recunoastere a informatiilor necesare chiar de la nivelul sistemului de operare (fara a necesita instrumente speciale ci doar comenzile existente in sistemul de operare). De asemenea o astfel de organizare este foarte apropiata de standardele clasice / originale ale aplicatiilor web.

Directoarele principale ale aplicatiei sunt enumerate la fiecare componenta din arhitectura si nu necesita explicatii suplimentare fata de cele scrise deja (vezi sectiunea [Arhitectura sistemului](#)). Detalii privind structura interna a fiecarui director aferent unei componente se regasesc in documentul de descriere detaliata aferent acelei componente. In situatii in care s-au considerat necesare diferite detalii pur specifice si "locale", in directorul componentei respective se gaseste un fisier `README_<cod-nume componenta>.md` ce prezinta aceste lucruri. Aceste documente se adreseaza in special dezvoltarilor aplicatiei (dar nu in mod obligatoriu) iar ca exemplu de ce poate contine un astfel de fisier este prezentarea modalitatii de pregatire a mediului local pentru dezvoltare.

Note

-
1. UI - desemneaza in general "Interfata Utilizator" (engl: User Interface) grafica, interactiva ce este afisata de catre browser (sau alta aplicatie GUI de acces) [↩](#)
-

Last update: September 9, 2023