

# OBALKYKNIH.CZ API 3.0,

## IMPLEMENTACE PRO ALEPH OPAC

---

*Charakteristika:* soubor skriptů, obrázků a popisů k implementaci komunikace s API služby obalkyknih.cz, **verze 1.1**, používá API obalkyknih.cz ver. 3.0/3.1

Balíček skriptů obsahuje: obalky\_knih.js, obalky\_knih\_eng.js (skripty pro úplné zobrazení),  
obrázky gif,png (logo, tlačítka pro komentáře, hvězdičky pro hodnocení),  
obalkyknih\_3.0\_failover\_daemon.sh  
add\_toc.cgi, obalkyknih\_toc2load.csh, check\_all\_recs4tocs.pl, check\_new\_recs4tocs.pl  
(skripty pro import OCR obsahů)

*Použité technologie:* javascript, BASH, JSON, XMLHttpRequest (u posledních dvou se předpokládá podpora prohlížeče – je ve všech aktuálních i docela dávnějších verzích IE, FF, Chrome, Opera, Opera mini i Safari).  
Pro import OCR obsahů pak CGI, BASH, CSH a Perl (v rozsahu modulů Aleph ver. 20-22 distribuce Perlu)

*Související SW:* ALEPH - testováno na ver. 22.0-1, předpokládá se ale kompatibilita zpět od verze 18  
Internet Explorer – min. verze 8, některé grafické prvky CSS bez vlivu na funkčnost až od IE 10

*Autor:* matyas.bajger@osu.cz, duben 2015. Verze 1.1. listopad 2015

## Obsah

1.	Front-end doména obalkyknih.cz a Failover Daemon.....	2
1.1	Co dělá skript? .....	3
1.2	Ruční zastavení daemona .....	3
1.3	Ruční spuštění .....	3
2.	OPAC short-view .....	4
2.1	šablona short-include-2 .....	4
2.2	šablona short-a-body .....	4
2b.	OPAC má schránka .....	5
2b.1	šablona myshelf-short-head .....	5
2b.2	šablona myshelf-short-body .....	5
3.	OPAC full 999 view .....	6
3.1	SAMOSTATNÝ SKRIPT obalky_knih.js .....	6
3.1.1.	získání identifikátorů knihy .....	6
3.1.2	Neplatné identifikátory v podpolích z .....	7
3.1.3	Volání API .....	7
3.1.4	Zpracování výsledků / zobrazení .....	7

3.2 Přesměrování.....	10
3.3 Úprava www opac šablon.....	11
3.3.1 Volání funkce pro API .....	11
3.3.2 zobrazení výsledků .....	11
4. Obsah v OCR .....	12
4.1. Základní předpoklady spuštění:.....	12
4.2 Základní workflow .....	13
4.3 Nastavení.....	13
4.4. Import TOC na pozadí.....	14
Verze 1.1.....	14

## 1. Front-end doména obalkyknih.cz a Failover Daemon

API obalkyknih.cz může být kvůli vytížení přístupné na více koncových serverech/doménách. Na straně knihovní aplikace je nutno zajistit pravidelnou dostupnost API na aktivní doméně, případně přepnout na jiný front-end

Pravidelnou kontrolu zajistí failover daemon, který změní doménu primárního serveru za jiný. Aktuální doménu uloží v souboru „obalkyknih-url“ ve `www_f_{lng}` adresáři.

```
{cesta}/obalkyknih_3.0_failover_daemon.sh
```

Bash, možno uložit kamkoli na server pro uživatele aleph. Ve skriptu je potřeba nastavit úvodní proměnné obsahující

- domény frontendů
- četnost testování odpovědi
- cestu na soubor s aktuální doménou ve `www_f_{lng}`
- zaplým parametrem `switch2first_after_hour` (s hodnotou true) lze nastavit první uvedený frontend jako prioritní. Pokud se tento stane neaktivní a uběhne více než hodina, daemon se na bude opět snažit přepnout na první uvedený frontend. Jinak (false) setrvává stále na dalším (druhém, třetím...) frontendu, dokud je funkční.

```
frontend_urls='cache.obalkyknih.cz cache2.obalkyknih.cz '
check_frequency=60 #sekund
OPAC_full_path_file="$salephe_root/www_f_cze/obalkyknih-url
                  $salephe_root/www_f_eng/obalkyknih-url"
switch2first_after_hour=true
```

spuštění daemona při bootu serveru pomocí cronu (lze použít cron uživatele aleph)

```
:~>crontab -e
```

```
@reboot exec /{cesta}/obalkyknih_3.0_failover_daemon.sh  
>>/exlibris/aleph/u2X_X/alephe/scratch/obalkyknih_3.0_failover_daemon.log  
2>&1 &
```

'@reboot' spouští při startu serveru

'exec' a '&' na konci spouští skript na pozadí, jako daemona

log i error výstup jsou směřovány do log souboru v alephe/scratch

## 1.1 Co dělá skript?

- kill ostatních spuštěných týchž démonů (ochrana proti vícečetnému spuštění)
- načte první doménu frontendu a nastaví ji v souboru `www_f_{lng}/obalkyknih-url`
- spustí nekonečný cyklus, který s nastavenou přestávkou (sleep)
  - prověří odpověď api (`.../api/runtime/alive`)
  - pokud do 10 sekund (dle dokumentace) nezíská odpověď "ALIVE" spustí další (pod)cyklus, v němž se postupně kontrolují odpovědi dalších nastavených frontendů
    - pokud najde ALIVE frontend, nastaví ho v souboru `www_f_{lng}/obalkyknih-url` a vrátí se do nekonečného cyklu
    - nevrací-li žádný frontend odpověď ALIVE, nastaví v souboru `www_f_{lng}/obalkyknih-url` prázdný řetězec (""), počká nastavenou přestávku a znovu kontroluje všechny frontendy
    - frontend, který nevrací odpověď nebo není "ALIVE", není následující hodinu kontaktován (požadavek dokumentace)
- odpovědi i neodpovědi API loguje v souboru `$alephe_scratch/obalkyknih_3.0_failover_daemon.log`

## 1.2 Ruční zastavení daemona

pomocí parametru "kill"

```
:~>obalkyknih_3.0_failover_daemon.sh kill
```

## 1.3 Ruční spuštění

```
:~> obalkyknih_3.0_failover_daemon.sh  
    >>&"$alephe_scratch/obalkyknih_3.0_failover_daemon.log" &
```

## 2. OPAC short-view

Používá se zjednodušená forma API, která nevrací JSON objekt, ale přímo URL náhledu obálky. Je v ní nově parametr keywords a volán je aktivní front-end obalkyknih.cz.

Z knižních identifikátorů se pro volání používá pouze ISBN/ISSN/ISMN a jen jedno IS[BSM]N (v případě více IS[BSM]N je v proměnných pro OPAC-short view jen první z nich při použití expandovaného pole ISB)

### 2.1 šablona short-include-2

1. musí se přidat načtení URL aktivního frontendu
2. zjistí se dotaz pro URL parametr obalek "keywords"

```
<script type="text/javascript">
//obalky knih api 3.0 url cache
var obalkyKnihUrl=
<include>obalkyknih-url
;
var obalkyKnihKeywords = encodeURIComponent( ('$0400'.match(/=.*/) ||
['$0400'])[0].replace(/=\s*/, ''));
</script>
```

(pseudo-tag <include> nesmí být odsazen mezerami, taby apod. od začátku řádku)

### 2.2 šablona short-a-body

Do příslušného sloupce pro obálku vložit načtení obrázku obálky zanešovaného do linku na obalkyknih.cz přes jednoduché API

```
<span id="obalka-$0600"></span>
<script type="text/javascript">
    if ( '$0600'!='<BR>' && obalkyKnihUrl!='' ) {
        document.getElementById('obalka-$0600').innerHTML='<a
            href="http://www.obalkyknih.cz/view?isbn=$0600"
            title="Klikni pro přechod na portál obalkyknih.cz"></a>'; }
</script>
```

Zde se předpokládá, že ISBN se nachází v placeholderu (opac proměnné) \$0600, což odpovídá nastavení ISBN www\_tab\_short.{lng} s prvním sloupcem hodnotou 4:

4 L Obálka 00 00 0100 S ## ISB

Je-li ve www\_tab\_short.{lng} ISBN pod jiným sloupcem, pak hodnota placeholderu = (1\_sloupec + 2) \* 100

Zde se ISBN bere s expandovaného pole ISB získaného pomocí expanze `expand_doc_bib_isxn` v `expand` menu web-brief

Pokud ISBN není, tak Aleph vrací v placeholderu text '<br>' a API se zbytečně nevolá.

volání URL `http://'+obalkyKnihUrl+'/api/cover?isbn=...` vrací obrázek 170x240px

Lze stahovat i menší náhledy pomocí přidaného parametru `type`:

`&type=icon` : 54x68px

`&type=thumbnail` : 27x36px

## 2b. OPAC má schránka

### 2b.1 šablona myshelf-short-head

za tag `<body>` přidat

```
<script type="text/javascript">
//obalky knih api 3.0 url cache
var obalkyKnihUrl=
<include>obalkyknih-url
;
</script>
```

### 2b.2 šablona myshelf-short-body

Do příslušného sloupce pro obálku vložit načtení obrázku obálky zanestovaného do linku na `obalkyknih.cz` přes jednoduché API, již bez url parametru `keywords`

```
<span id="obalka-$0800"></span>
<script type="text/javascript">
    if ( '$0800'!='<BR>' && obalkyKnihUrl!='' ) {
        document.getElementById('obalka-$0800').innerHTML='<a
            href="http://www.obalkyknih.cz/view?isbn=$0800"
            title="Klikni pro přechod na portál obalkyknih.cz"></a>';
        }
    }
</script>
```

ISBN se předkládá v proměnné (placeholderu) `$0800`, odpovídá nastavení ISBN `www tab short.{lng}` s prvním sloupcem hodnotou 4. Jinak se jedná o ( číslo v prvním sloupci + 4 ) \* 100.

### 3. OPAC full 999 view

- volá "úplné" API vracející JSON objekt. Ten může obsahovat data (linky na):
  - link na obálku,
  - link na náhled obsahu,
  - link na PDF obsahu,
  - OCR obsahu
  - uživatelská hodnocení, uživatelské komentáře,
  - bibl. identifikátory a link na obalkyknih.cz
- API je voláno javascriptem asynchronně pomocí XMLHttpRequest, což ale nelze z jiné domény (cross-origin permissions). Potřebné je přesměrování je zajištěno instrukcí Apache v httpd.conf
- pro objemnost javascriptu (cca 20 kB) je tento v samostatném souboru mimo www opac šablony

#### 3.1 SAMOSTATNÝ SKRIPT obalky\_knih.js

uložit sem `$httpd_root/htdocs/obalky_dir/obalky_knih.js` nebo jinam na Apache. Do téhož adresáře uložit obrázky používané ve skriptu (logo obálek, obrázky pro hodnocení a komentáře). Při použití jiné cesty je nutno změnit cestu i při volání skriptu z OPAC `www_f_{lng}` šablon (viz níže) a cesty na obrázky v samotném skriptu `obalky_knih.js`.

Skript je jazykově závislý, české texty jsou u hodnocení a komentářů a rovněž u `alt` a `title` atributů náhledů obálek a obsahů. Pro anglickou verzi (`www_f_eng`) lze použít variantu `obalky_knih_eng.js`, která si rovněž volá anglická tlačítka s názvem `*_eng.gif`

Části skriptu `obalky_knih.js`:

##### 3.1.1. získání identifikátorů knihy

Soubor metod/funkcí pod objektem `identifiers` (mimo objekt `obalkyKnih`, neboť identifikátory lze použít i jinde).

Prochází celý html dokument (`document.body.textContent`) a hledá v něm pomocí regulárních výrazů identifikátory na základě syntaxe, příp. kontrolního součtu

```
identifiers.getISBNs() { ... }
```

hledá ISBN: 10 i 13 místná, s pomlčkami i bez nich; ISSN; ISMN: 10 i 13 místná. U nalezených čísel prověřuje kontrolní součet.

Dále hledá EAN kódy (Marc21 pole 024, 1.ind. 3), a to podle prefixu EAN s 8-13 následnými číslicemi. Jelikož v samotném záznamu se EAN kód zapisuje bez prefixu (na rozdíl od ČNB a OCLCno.), je nutno doplnit prefix při zobrazení pomocí řádku v `$data_tab/edit_doc_999.lng`:

```
## 0243# D LEAN Y E
```

```
identifiers.getOCLCno() { ... }
```

hledá čísla začínající prefixem "(OCoLC)".

```
identifiers.getCNB() { ... }
```

hledá čísla začínající prefixem "cnb".

Funkce vracejí pole s nalezenými identifikátory.

### 3.1.2 Neplatné identifikátory v podpolích z

by se bez úpravy rovněž použily pro volání API (pokud sedí kontr. součet isbn)

Nabízí se cca. 2 řešení

Var. 1 před neplatný identifikátor vložit vykřičník

funkce počítají s tím, že identifikátory začínající vykřičníkem (např. !978-80-542..., !cnb0015..) jsou neplatné a vyloučí je z vrácených výsledků.

```
xxx01/tab/edit_field.{lng}
```

1	#	015##	W	
2				a
2				z A
^!			^(chyb.)	
1	#	020##	W	
2				a
ISBN^				
2				q
^				
2				z A
^^ISBN^!			^(chyb.)	
1	#	035##	W	
2				a
2				z A
^!			^(chyb.)	

Var. 2 neplatné identifikátory vůbec nezobrazovat

vyloučit podpole "z" v 6. sloupci v edit\_field.{lng} (hodnota „-z“)

### 3.1.3 Volání API

```
obalkyKnih.ask() { ... }
```

- hlavní funkce, spouštět ji lze přes <body> atribut onload po načtení celé stránky (doporučeno, viz níže) nebo již dříve v \*-tail www šabloně
- zavolá výše uvedené metody objektu „identifiers“, tím získá identifikátory a poskládá je do URL
- pomocí XMLHttpRequest zavolá API na aktivním frontendu. Kvůli cross-domain musí být volání API směrováno na přesměrování na doméně OPACu (viz níže).
- Převede získaný JSON do objektu a pokud nalezne hodnoty obálek, obsahů, hodnocení zavolá funkce pro zobrazení nebo může vykonat cokoli jiného:

### 3.1.4 Zpracování výsledků / zobrazení

Probíhá pomocí řady metod/funkcí, které lze využít.

téměř všechny tyto funkce začínají definicí:

```
var targetEl=document.getElementById('XXX');
```

kde XXX je id atribut html elementu, kde se mají zobrazit výsledky vrácené v odpovědi API

Např.:

```
var targetEl=document.getElementById('ob_cover');
```

předpokládá někde na stránce element `<div id='ob_cover'></div>` (příp. `span`, `td`), kde se zobrazí obálka. Element může být ve výchozí podobě skryt pomocí `style="display:none; "` Funkce hodnotu `display` zruší – zviditelní jej.

`obalkyKnih.showCover(coverImg, backlink) { ... }`

Do určeného html elementu (výchozí je `id="ob_cover"`) vloží náhled obálky 170x240px doplněný o logo obalkyknih.cz. Náhled i logo jsou vloženy do odkazu na stránku knihy na [www.obalkyknih.cz](http://www.obalkyknih.cz)

`obalkyKnih.showTOC(pdfURL, thumbnail) { ... }`

Do určeného html elementu (výchozí je `id="ob_toc"`) vloží náhled obsahu 170x240px vložený do odkazu na PDF s obsahem.

`obalkyKnih.addOCRToc(toc2send, partRoot) { ... }`

Lze použít pro stažení plného textu obsahu do záznamu pro indexaci (apod.). Více viz kapitola "Obsah v OCR". Nechcete-li používat, raději volání této funkce zakomentujte.

`obalkyKnih.showRatingSimple = function(starsURL, backlink) { ... }`

Jednodušší podoba uživatelského hodnocení, bez možnosti přidávat hodnocení prostřednictvím OPACu (jen pasivní zobrazení). Do určeného html elementu (výchozí je `id="ob_rating"`) vloží obrázek s hvězdičkami generovaný službou obalkyknih.cz.

např.



<http://www.obalkyknih.cz/stars?value=50>

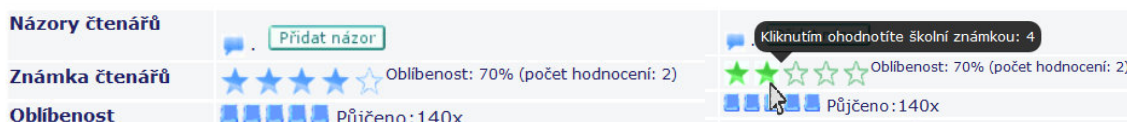
vložený do odkazu na stránku knihy na [www.obalkyknih.cz](http://www.obalkyknih.cz)

Ve výchozí podobě skriptu `obalky_knih.js` je tato varianta zakomentovaná. Pro použití odkomentujte řádek:

```
//obalkyKnih.showRatingSimple(obalkyKnih.json[i].rating_url, backlink);
```

`obalkyKnih.showRating = function(bookID, ratingAvg5, rating100, ratingCount) { ... }`

Funkce si sama v určeném elementu (výchozí je `id="ob_rating"`) generuje hvězdičky pro zobrazení a umožňuje přidávat vlastní hodnocení uložené v službě obalkyknih.cz přejetím myši (kurzorem) a kliknutím na hvězdičky. Po přejetí se v bublině vytvořené pomocí CSS zobrazuje vysvětlení (html atribut `"title2"`).




- pracuje jen s hodnocením 1,2,3,4,5 (podle hvězdiček) [ukázka](#)
- vedle hvězdiček má doprovodný text : hodnocení v procentech a počet hodnocení, resp. „Dosud nikdo nehodnotil. Buďte první!“ nebo „Děkujeme za vaše hodnocení“, příp. chybové hlášení
- asynchronně metodou POST posílá nové hodnocení na API obalkyknih.cz
- blokuje více přidání najednou (hvězdičky se nepodbarvují, v bublině je text „Již jste přidali hodnocení“), ale pouze do obnovení/znovunačtení stránky
- možnost kombinace s vlastními hodnoceními
- zahrnuje pouze české texty



- obsahuje 4 obrázky: rating\_plus.gif, rating\_minus.gif, rating\_add\_plus.gif, rating\_add\_minus.gif Ve výchozí podobě jsou očekávány ve stejném adresáři jako skript obalky\_knih.js.
- Dále je nutno do doplnit CSS style (např. v \$httpd\_root/htdocs/exlibris.css) – class „ratingStar“ :

```
.ratingStar { position: relative; text-decoration: none; width: 20px; height: 20px; float: left; margin: 2px; }
.ratingStar:hover:before { display: block; position: absolute; padding: .5em; content: attr(title2); min-width: 120px; text-align: center; width: auto; height: auto; white-space: nowrap; top: -32px; background: rgba(0,0,0,.8); border-radius:10px; color: #fff; font-size: .86em; }
.ratingStar:hover:after { display: block; position: absolute; content: ""; border-color: rgba(0,0,0,.8) transparent transparent; border-style: solid; border-width: 10px; height:0; width:0; top: -8px; left:1em; }
```

obalkyKnih.showReviews(review) {....}

Pro zobrazení a přidávání komentářů. Do nastaveného html elementu vloží vrácené komentáře uvedené ikonou . Po přejetí myši přes komentář se objeví CSS bublina s datem čtenáře a knihovnou, za níž byl komentář přidán.

### Úplné zobrazení záznamu



**Záznam 1 z 1**

**Názory čtenářů**

**Známka čtenářů**

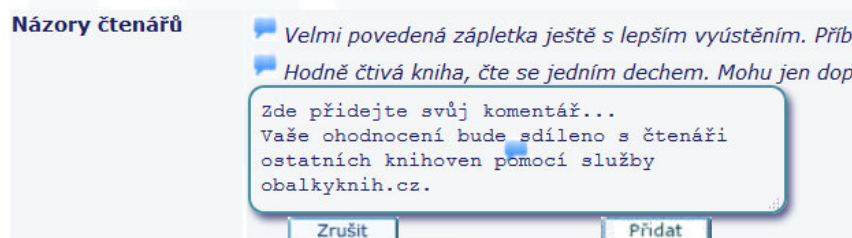
**Oblíbenost**

**Formát**

**Hlavní záhlaví**

**Název**

Za komentáře, příp. rovnou (pokud nejsou) se přidá tlačítko [Přidat názor](#). Kliknutím na něj se zobrazí element textarea s vysvětlením (mizí po kliknutí) pro přidání komentáře.



[ukázka](#)

Tlačítko "Zrušit" element zase skryje.

Tlačítkem "Přidat" se spustí metoda pro zaslání komentáře. Ověří se, jestli není poslán prázdný text a asynchronně POST metodou volá API obalkyknih.cz. Pokud API vrátí správnou odpověď ("ok"), nově přidáný komentář se zobrazí (+ zůstává stále možnost přidat další komentáře).

Jinak se za stávající komentáře přidá chybové hlášení (stejně jako u hodnocení se čeká na odpověď 15 sekund). Ke komentáři se přidává id komentáře – unix time v milisekundách.

- zahrnuje pouze české texty
- obsahuje 3 obrázky tlačítek: f-add-review.gif, f-add.gif, f-cancel.gif očekávané ve stejném adresáři jako skript obalky\_knih.js.

Dále je nutno do doplnit CSS styly (např. v `$httpd_root/htdocs/exlibris.css`) – class „userReviews“ a „newReview“:

```
.userReviews {font-style: italic; position: relative; float: left;}
.userReviews:hover:before { display: block; position: absolute; padding: .5em; content: attr(title2); min-width: 120px; text-align: center; width: auto; height: auto; white-space: nowrap; top: -32px; background: rgba(0,0,0,.8); border-radius:10px; color: #fff; font-size: .86em; }
.userReviews:hover:after { display: block; position: absolute; content: ""; border-color: rgba(0,0,0,.8) transparent transparent transparent; border-style: solid; border-width: 10px; height:0; width:0; top: -8px; left:1em; }
textarea.newReview {width: 25em; height: 5em;border: 2px solid #438E9C; padding: 0.5em; background: url('/obalky_dir/comment.png') no-repeat center center; color:#212063; border-radius: 10px; box-shadow: 3px 3px 4px 0px rgba(33,32,99,0.75); clear: left; display: block;}
```

Přidané komentáře lze administrovat (prohlížet, mazat) po přihlášení na webu [www.obalkyknih.cz](http://www.obalkyknih.cz). Tamtéž si lze nastavit týdenní zasílání reportu nově přidaných komentářů. Pro okamžité informování o nově přidaném komentáři si lze vytvořit vlastní server-side skript a zavolat jej ve funkci `targetEl.set()`. Nebo pravidelně kontrolovat `access_log` Apache (viz níže v „Přesměrování“).

## 3.2 Přesměrování

Pro volání API ve full zobrazení je kvůli cross-domain policy nutno do konfigurace Apache `$httpd_root/conf/httpd.conf` přidat direktivu:

```
#presmerovani pro api obalkyknih.cz
RewriteRule ^/obalky/(.*obalkyknih\.cz.+) http://$1 [p,L]
```

Předpokladem je ještě tamtéž (`httpd.conf`) výše zapnutá direktiva `RewriteEngine On`, což by ale mělo být výchozí nastavení. Po úpravě restart Apache.

Po tomto nastavení se API volá na doméně OPACU v cestě `/obalky/`, např.

`'http://aleph.osu.cz/obalky/cache.obalkyknih.cz/api/books?multi=...'` a Apache požadavek přesměruje na doménu `cache.obalkyknih.cz`.

Přesměruje se jen na doménu obsahující text `"obalkyknih.cz"`.

Flags v hranaté závorce znamenají: `p`=funguj jako proxy, tj. do prohlížeče jde odpověď z domény OPACu (nezbytné kvůli cross-origin), `L`=last, neprováděj další přesměrování

Při použití tohoto přesměrování lze `access_log` Apache použít k sledování nově přidaných komentářů a hodnocení, např. k posílání pravidelných přehledů/kontrol na mail. Např. pomocí:

```
:~>grep 'obalkyknih.cz' $httpd_root/logs/access_log | grep 'add_review=true'
```

Alternativně lze místo přesměrování Apachem použít cgi skript `redir.cgi` (součást balíčku ver 1.1), umístěný do adresáře s povolení cgi pro Apache. Pro přepnutí na přesměrování cgi zakomentujte (//) v skriptu `obalky_knih.js` řádek

```
obalkyKnih.url= '/obalky/'+obalkyKnih.domain+'/api/books?multi='  
+ encodeURIComponent(uri_multi);
```

a odkomentujte po něm následující řádek + upravte v něm relativní url cestu na cgi skript:

```
obalkyKnih.url= '/cgi-bin/uncgi/redir.cgi?redd=' +  
    encodeURI('http://' + obalkyKnih.domain + '/api/books?multi=' +  
    encodeURIComponent(uri_multi));
```

Upozornění: přesměrování přes `redir.cgi` nelze použít k posílání komentářů a hodnocení do `obalkyknih.cz` pomocí `uncgi` (není připraven na posílání POST požadavků).

## 3.3 Úprava `www.opac` šablon

### 3.3.1 Volání funkce pro API

Do šablon

`direct-head`

`full-set-head-bor`

`full-set-head-ill`

`full-set-head-ill-02`

`full-set-head-ill-nobor`

`myshef-full-head`

doplnit načtení externího javascriptu, spuštění hlavní funkce volání api a načtení URL aktivního frontendu

```
<head>  
....  
<script type="text/javascript"  
    src="/obalky_dir/obalky_knih.js"></script>  
(....)  
</head>  
<body (...) onload="obalkyKnih.ask(); (...) " (...)>  
<script type="text/javascript">  
//obalky knih api 3.0 url cache  
obalkyKnih.domain=  
<include>obalkyknih-url  
;  
</script>
```

### 3.3.2 zobrazení výsledků

Na stránce (v body) přidat elementy, v nichž se mají zobrazovat výsledky

zpravidla jde o šablony: `direct-tail`, `full-set-tail`, `myshef-full-tail`

Pro obálku je to: `<div id="ob_cover"></div>`

Pro obsah: `<div id="ob_cover"></div>`

Pro hodnocení: `<div id="ob_rating"></div>`

Pro komentáře: `<div id="ob_reviews"></div>`

Tyto divy umístit dle potřeby na stránku. Lze je zneviditelnit pomocí `style="display: none;"`. Skripty se postarají o smazání "none" hodnoty, pokud elementy naplní obsahem.

## 4. Obsah v OCR

Volitelně může javascript `obalky_knih.js` asynchronně volat shell skript `add_toc.cgi` (součást balíčku verze 1.1), který uloží na server obsah v čitelném textu pro pozdější zpracování, např. přidání do bibl. záznamu pro indexování, zobrazení.

### 4.1. Základní předpoklady spuštění:

1. Na stránce úplného zobrazení je globální proměnná `sysno` (`window.sysno`) s hodnotou systémového čísla, která se načte z "dolarových" proměnných ALEPH `www` serveru (placeholders). Konkrétní šablony tak mají obsahovat skript/část skriptu:

```
direct-head:  var sysno = '$0100';
head-1-bor:   var sysno='$0600';
head-1-nobor: var sysno='$0600';
myshef-full-head:  var sysno="$0100";
```

2. Apache je připraven spouštět cgi skripty
3. V biblio bázi je nadefinováno pole TOC pro vkládání textů obsahů. Konkrétně se jedná v základním nastavení o následující soubory a řádky v `$data_tab`, pokud chceme pole zobrazovat v OPAC-úplném a prohledávat v základním indexu všech slov WRD:

<code>check_doc_line:</code>	<code>AL XX</code>	<code>TOC</code>	<code>-</code>			
<code>check_doc_line:</code>	<code>AL XX</code>	<code>TOC</code>	<code>a 1 -</code>			
<code>check_doc_line:</code>	<code>AL XX</code>	<code>TOC</code>	<code>9 0 -</code>			
<code>edit_doc_999.cze:</code>	<code>## TOC</code>			<code># LObsah</code>		<code>Y</code>
	<code>E</code>					
<code>edit_doc_999.eng:</code>	<code>## TOC</code>			<code># LTOC</code>		<code>Y</code>
	<code>E</code>	<code>W</code>				
<code>tab01.cze:</code>	<code>D TOC</code>	<code>00 0000</code>	<code>TOC</code>	<code>LObsah</code>		
<code>tab01.eng:</code>	<code>D TOC</code>	<code>00 0000</code>	<code>TOC</code>	<code>LTable of Contents</code>		
<code>tab11_word:</code>	<code>TOC##</code>		<code>a</code>		<code>03</code>	<code>WRD</code>

Pole TOC může v OPAC – úplném zobrazení působit rušivě (příliš mnoho textu). Řešením může být výchozí zmenšení písma, pokud se v něm nenajde match dotazu a možnost zvětšení ikonou lupy.<sup>1</sup>

## 4.2 Základní workflow

Javascript `obalky_knih.js` volá asynchroně (XMLHttpRequest) cgi shell skript `add_toc.cgi` (součást balíčku od ver. 1.1). Cgi uloží získaný plný text obsahu do souboru v dočasném adresáři. Pravidelně, např. 1x týdně na pozadí spouštěný csh skript `obalkyknih_toc2load.csh` projde soubory s obsahy v dočasném adresáři a zajistí jejich import (pokud již záznam obsahuje pole TOC, nic se nepřidá).

## 4.3 Nastavení

1. **obalky\_knih.js** : odkomentujte (odstraňte úvodní lomítka) v řádku (ve výchozí podobě je funkčnost blokována):

```
// obalkyKnih.addOCRToc(obj.toc_full_text,partRoot);
```

a v násl. řádku nastavte relativní URL cestu k cgi skriptu:

```
sendTOC.open("POST", '/cgi-bin/add_toc.cgi', true);
```

2. **add\_toc.cgi** : uložte do adr. povoleného a pro cgi skripty. V úvodu skriptu se nastaví :

```
toc_tmp_dir="/exlibris/aleph/....."
admin_mail=' '
```

- a) adresář, kam se mají ukládat obsahy pro pozdější import (`toc_tmp_dir`). Soubory se jmenují "TOC"+`sysno[+"ROOT"].yyyymmdd` a obsahují text obsahu vrácený API (pozor na scratch adresáře, mohou být nějakým skriptem pravidelně promazávány)
- b) volitelně mail, kam skript pošle hlášení o chybě, pokud nastane  
Skript loguje do souboru 'add\_toc.log'

3. **obalkyknih\_toc2load.csh** : umístěte do libovolného uživ. Aleph přístupného adresáře.

V jeho úvodu je potřeba změnit nastavit parametry (proměnné)

```
set bibBase='XXX01'
set dirwithTOCs='/exlibris/aleph/u22_1/alephe/scratch' #cesta na
soubory, co se mají importovat, shodná s nastavením v add_toc.cgi
set adminMail=" " #pro posílání výsledků importu
```

Skript prochází adresář se soubory pro import (soubory 'TOC\*') a nalezené obsahy importuje do pole TOC (manage-18), pokud již záznam pole TOC nemá. Delší obsahy rozděljuje na jednotlivá pole TOC s max. délkou 1500 znaků, jelikož Aleph 18-22 má limit 2000 bytů na pole.<sup>2</sup> Rozděljuje přitom text jen na mezerách, nikoli uprostřed slov. Prověřuje se rovněž max. limit celého

---

<sup>1</sup> Vzorový skript je součástí balíčku: `lupa_vzor.js`

<sup>2</sup> Pro limity ALEPHu srov. [http://support.exlibrisgroup.com/articles/Q\\_A/Long-MARC-Fields-Limits-and-Splitting/?q=subfield+9&l=en\\_US&fs=Search&pn=3](http://support.exlibrisgroup.com/articles/Q_A/Long-MARC-Fields-Limits-and-Splitting/?q=subfield+9&l=en_US&fs=Search&pn=3)

záznamu – 42000 bytů<sup>3</sup> (lze změnit proměnnou 'recordLimit'), při jehož překonání se text obsahu ořízne do požadované velikosti. Dále se prověřuje, zda záznam nemá více ISBN (je zde předpoklad vícedílné publikace), ISBN v poli 902 a identifikace root záznamu v odpovědi API. Na tyto záznamy skript upozorní v zasílaném mailu.

Po importu je zaslán mail na nastavenou adresu pro případné prověření importovaných obsahů.

Skript 'obalkyknih\_toc2load.csh' se doporučuje pravidelně spouštět, např. pomocí \$alephe\_tab/job\_list 1x týdně.

## 4.4. Import TOC na pozadí

Součástí balíčku jsou také perl skripty `check_new_recs4tocs.pl` a `check_all_recs4tocs.pl`. Ty nemají spojitost s OPACem, ale slouží pro pravidelné/jednorázové kontroly nedávno změněných, resp. všech záznamů v katal. bázi oproti databázi obalkyknih.cz skrze jeho API. Dohledá v záznamech identifikátory, zavolá API obalkyknih.cz. Nalezené obsahy jsou uloženy do dočasného adresáře pro pozdější import, obdobně jako u skriptu `add_toc.cgi` pro pozdější import skriptem `obalkyknih_toc2load.csh`.

`check_new_recs4tocs.pl` prochází nedávno změněné záznamy (podle `z13_update_date`) a lze jej spouštět pravidelně např. přes \$alephe\_tab/job\_list `check_all_recs4tocs.pl` slouží k jednorázovému doplnění do všech záznamů v bázi

U skriptů je nutno nastavit cestu na Aleph Perl (úvodní řádek) a základní proměnné

```
my $sid = 'dbi:Oracle:host=localhost;sid=aleph22'; #srov. ENV
    proměnnou ORACLE_SID
my $bibBase = 'xxx01';
my $obDomainFile =
    '/exlibris/aleph/u22_1/alephe/www_f_cze/obalkyknih-url';
    #cesta, kde se najde aktivní frontend (jako pro OPAC)
my $dir4TOCs='/exlibris/aleph/u22_1/alephe/sctatch'; #cesta pro
    uložení obsahů pro pozdější import.
a u skriptu check_new_recs4tocs.pl počet dnů zpět, kdy byly změněny záznamy, jež se mají
    dohledat. Nastavení záleží na četnosti spouštění:
my $daysBack=7;
```

## Verze 1.1

Změny v skriptu `obalky_knih.js` jsou označeny komentářem '// ver 1.1.'

---

<sup>3</sup> Limit Alephu 18-22 je 45000 bytů, při importu se ale přidává ještě pole CAT, mohou zde být další přidání, proto je ponechána rezerva.

## **Opravy**

1. Failover Daemon - návrat na výchozí server po obnovení jeho funkčnosti (někdy zůstával na druhém bez ohledu na parametr `switch2first_after_hour=true`)
2. Chybné chování, pokud nebyl aktivní žádný frontend server
3. Chyba, pokud všechny identifikátory jednoho typu (ISBN,EAN,ČNB) byly nevalidní - zrušené (podpole z).
4. Přesměrování volání API pomocí `mod_proxy` Apache (Apache/2.2.29 (Unix) mod\_ssl/2.2.29 OpenSSL/0.9.8zf mod\_perl/2.0.5 Perl/v5.8.9) dle zkušeností OU někdy nefungovalo, příčinu se nepodařilo zjistit kromě hlášení `error_logu` Apache: *"child pid NNNN exit signal Segmentation fault (11)"* . Chyba se sama objeví i pomine. Problém s moduly Apache se objevil také v MSVK, kde běží celý OPAC pod secure https protokolem. Místo přesměrování Apache lze použít cgi skript `redir.cgi` (viz výše přesměrování).
5. OPAC – full view, načítání náhledu obálky a obsahu pro kompletně https zabezpečený opac ("načítání nezabezpečeného obsahu na zabezpečené stránce")

## **Novinky**

1. jako identifikátor pro volání API se bere také EAN kód - musí být v plném zobrazení s prefixem "EAN" (Marc21 pole 0243 )
2. Do balíčku doplněn cgi skript pro přidávání TOC obsahu do záznamů
3. vrátili-li API neuplnou odpověď (stava-se), chybi konec obsahu, JSON je nevalidní - ošetřeno, takže se z odpovědi načte aspoň ta část dat, která přijde