AiP Beroun s.r.o., http://www.aipberoun.cz, tel.: +420 311 611 237, fax: +420 311 611 238
Talichova 807, Beroun, 266 01; IČO: 25778943, DIČ: 25778943

REGISTR DIGITALIZACE HISTORICKÝCH FONDŮ

Specifikace REST rozhraní pro klientské aplikace a další systémy

Technická specifikace

verze 1.0

Ing. Karel Kučera, AiP Beroun s.r.o.

Obsah

1 Úvod o dokumentu

1.1 Účel

Tento dokument je součástí technické dokumentace softwarového řešení projektu „Registr digitalizace historických fondů" jež je součástí VaV 2016. Dokument by měl dále sloužit jako jeden z výchozích informačních zdrojů pro další rozvoj tohoto systému, především pro údržbu stávajících a tvorbu nových klientských aplikací využívajících služeb serveru RDHF. Dále také například pro vývoj prostředků umožňujících využití služeb RDHF dalším kooperujícím systémům v této oblasti.

1.2 Předpokládaný čtenář

Dokument je určen pro členy týmu, realizujícího výše zmíněné projekty, především pak programátorům. Dále je určen všem, kteří se budou podílet na dalším budoucím rozvoji systému „Registr digitalizace historických fondů".

1.3 Termíny a konvence

Termíny použité v tomto dokumentu jsou popsány a vysvětleny v dokumentu [1].

1.4 Reference

V dokumentu se odkazujeme na následující literaturu:

[1] AiP Beroun, „Registr digitalizace historických fondů - softwarové řešení, Technická specifikace," Beroun, 2016.

2 Úvod

Registr digitalizace historických fondů (dále RDHF) a URI resolver pro historické dokumenty (dále resolver) jsou realizovány jako vícevrstvá aplikace, skládající se ze serverové části a klientských aplikací. Serverová část je společná pro RDHF i Resolver. Klientské aplikace jsou realizovány formou webových a nativních aplikací. Se serverovou částí komunikují prostřednictvím bezestavového HTTP rozhraní REST.

Ve své počáteční podobě RDHF obsahuje tři základní typy klientských aplikací: Aplikace pro veřejné uživatele umožňující přístup ke službám RDHF prostřednictvím webového rozhraní, aplikace pro pořizování a údržbu dat správci systému a aplikaci pro administrátora. Ta umožňuje údržbu celého systému RDHF (respektive jeho serverové části).

V tomto dokumentu je podrobně popsáno aplikační rozhraní, které umožňuje klientským aplikacím komunikovat se serverem RDHF. Aplikační rozhraní (API) je rozděleno na tři části, a to podle typů klientských aplikací. Veřejné rozhraní – umožňuje přístup k RDHF běžným uživatelům nevyžaduje autentifikaci. Další dvě rozhraní pro správce a administrátory vyžadují pro práci s RDHF autentifikaci uživatele.

Díky tomu, že při začátku vývoje systému RDHF i vzniku tohoto dokumentu ještě nebyla ustálení terminologie, názvy volání a především názvy parametrů volání nemusí vždy odpovídat současné platné terminologii. Vzhledem k tomu, že zmiňované názvy nejsou veřejné a jsou použity pouze při programování jednotlivých částí systému, není ovšem tato skutečnost nijak na závadu.

3 Technická specifikace rozhraní

3.1 Volání funkcí serveru

REST rozhraní k serverové části projektu bude mít vždy následující strukturu:

```
http://<adresa_serveru>:<port>/<typ_rozhraní>/<operace>?<parametry>
```

kde <adresa_serveru> je URL adresa, na které je k dispozici server RDHF a <port> je port pro server RDHF
<typ_rozhraní> - (resource) definuje činnost, kterou bude server provádět. Činnosti lze rozdělit podle typu uživatelských rozhraní na:

- veřejné či uživatelské - volný přístup prostřednictvím webové aplikace
- činnosti pro správu dat RDHF a Resolveru – jsou přístupné prostřednictvím buď webové nebo nativní klientské aplikace přihlášenému uživateli
- činnosti pro administraci projektu – přístupné nativní klientské aplikaci určené administrátoru projektu – vyžaduje přihlášení

search - veřejné uživatelské rozhraní
manage - rozhraní pro správu dat
admin - rozhraní pro administraci systému

Volání rozhraní **search** jsou přístupné prostřednictvím metod **GET** a **POST**, rozhraní **manage** a **admin** jsou přístupné POUZE prostřednictvím metody **POST**!

příklad REST volání:

```
http://rdhf.cz:8080/search/digCopyRec?shelfmark=xxxx&repository=yyyy
&settlement=zzz
```

Příklad uvádí volání veřejného rozhraní search se žádostí o předání záznamu(ů) o digitální(ch) kopii(ích) k fyzickému dokumentu určenému jeho (současnými nebo dřívějšími) lokačními údaji signatura (shelfmark), repository a settlement.

3.2 Odezva na volání serveru

Odezva serveru (response) je realizována v souladu se specifikací HTTP ve strukturované textové podobě ve formátu JSON. Struktura odezvy je definována takto:

```
{"request" : "<název_requestu>", "params" :
     {"<parameter 1>" : "<value1>",
```

```
      "<parameter 2>" : "<value2>",
            …
      },
      "result" : {

            <odezva specifická pro každé volání (request) při
            úspěšném volání, v případě neúspěchu slovní popis
            chyby >
      },
      "errorCode" : <číslo chyby, při úspěšném volání errorCode = 0,
      tj. bez chyby>
}
```

3.3 Formát předávaných parametrů

HTTP parametry jednotlivých volání jsou předávány jako řetězce v kódování UTF-8. Parametry pro předání datumu a času mají formát YYYY-MM-dd HH:mm:ss, kde:

YYYY - rok (čtyři znaky), např. 2015
MM - měsíc (dva znaky), tj. 01, 12 apod.
dd - den v měsíci (dva znaky), tj. 03, 28, 31 apod.
HH - hodiny 0-23 (dva znaky), tj. 00, 11, 23 apod.
mm - minuty (dva znaky), tj. 05, 32, 59 apod.
ss - sekundy (dva znaky), tj. 00, 32, 59 apod.

Správný čas je v rozsahu 00:00:00 – 23:59:59

3.4 Identifikace uživatele

Rozhraní pro správu záznamů (API Manage) a administraci systému (API Admin) pro provedení requestu požadují identifikaci uživatele. Ta je provedena pomocí parametru userId, jehož platnost je ověřována na straně serveru. Identifikátor userId se skládá, ze třech částí:

- Role - role uživatele např. správce digitálních kopií, správce konkordancí, supervisor, administrátor
- Oprávnění – oprávnění daného uživatele jako zápis záznamu, editace, mazání
- Kontrolní kód – kód předaný serverem na vyžádání klienta, do serveru se vrací ke kontrole platnosti jako součást userId

Struktura identifikátoru:

```
RRRCTRLCTPRMuuuuuuuu – 8 znaků = kód uživatele z Drupalu
 |      |   |_______ 3 znaky = oprávnění uživatele
```

```
|      |__________ 6 znaků = kontrolní kód
|_________________ 3 znaky = role uživatele
```

Role uživatele – udává, se kterými typy dat smí nakládat (RDHF, konkordance, datová úložiště), každá role ke každému typu dat je vyjádřena jedním znakem, které lze podle role i kombinovat.
1 – správce, 2 - supervisor, 999 – administrátor

```
RRR
|||__ digitální kopie
||___ konkordance
|____ datová úložiště
```

Dále jsou uvedeny příklady rolí:

000 - obecný uživatel – nemá právo na nic ☺
001 - správce digitálních kopií
002 - hlavní správce digitálních kopií (supervisor digitálních kopií)
010 - správce konkordancí
020 - hlavní správce konkordancí (supervisor konkordancí)
100 - správce datových úložišť
011 - správce digitálních kopií a konkordancí
202 - hlavní správce digitálních kopií a datových úložišť (supervisor)
999 - administrátor

Oprávnění uživatele – určuje činnosti, které je uživatel oprávněn s daty provádět.

```
PRM
|||__ digitální kopie
||___ konkordance
|____ datová úložiště
```

000 - pouze číst
001 - zápis do tabulky digitálních kopií
002 - zápis a editace v tabulce digitálních kopií
003 - mazání v tabulce digitálních kopií
004 - úplný přístup do tabulky digitálních kopií (zápis, editace mazání)
010 - zápis do tabulky konkordancí
020 - zápis a editace konkordancí
030 - mazání konkordancí
040 - úplný přístup ke konkordancím
100 - zápis do tabulky datových úložišť
200 - zápis a editace záznamů v tabulce datových úložišť
300 - mazání v tabulce datových úložišť
400 - úplný přístup k tabulce datových úložišť

Tato oprávnění lze samozřejmě kombinovat, takže např.

044 - zápis, editace i mazání (úplný přístup) v tabulkách konkordancí a digitálních kopií
111 - pouze zápis do všech tabulek

Tímto mechanismem lze upravit oprávnění různých osob v definovaných rolích. Lze např. zajistit, že administrátor, který má přístup ke všem tabulkám, nemůže vkládat či upravovat záznamy v některých tabulkách.

```
999xy4azx4330000023
```

Pro příklad výše uvedený identifikátor je předán pro administrátora (999), mezi jehož administrátorské povinnosti patří kompletní údržba tabulky datových úložišť (4) a možnost (na požádání) mazat záznamy z ostatních tabulek (nesmí je ani vytvářet, ani editovat). Uživatel je v systému práv Drupalu veden pod číslem (identifikátorem) 00000023.

Kontrolní kód je generován serverem a má omezenou časovou platnost. Kontrolní kód je předán serverem na vyžádání klienta pomocí requestu „ctrlCode". Klient s pomocí tohoto kontrolního kódu sestaví platný identifikátor uživatele (userId) a předá ho jako parametr pro requesty API Manage a Admin. Request ctrlCode je dostupný přes rozhraní Manage.

Parametr userId pro requesty rozhraní Manage a Admin je zakódovaný pomocí mechanismu base64.

3.5 Kódy chybových hlášení

INTERNAL = 1 - Obecná, blíže nespecifikovaná chyba. Tato chyba například vzniká v prostředcích, které používá server RDHF
GENERAL = 1 - dtto
SERVER_DISABLED = 128 – server není pro uživatele (kromě administrátora) přístupný

Obecné chyby:
NOT_FOUND = 1024
INVALID_NAME = 1025
OUT_OF_RANGE = 1026
INVALID_INPUT_VALUE = 1027
INVALID_OAI_IDENTIFIER = 1028
INVALID_REQUEST = 1029
CONFIG_MANAGER = 1030
UNKNOWN_DATASTORAGE = 1031

Přístup do databáze:

DB_INPUT_RECORD = 1280
DB_REC_NOT_FOUND = 1281
DB_REC_ALREADY_EXISTS = 1282

Přístupová práva:
INVALID_RIGHTS = 1536
INVALID_USERID = 1537
INVALID_CTRL_CODE = 1538

4 Funkce veřejného rozhraní search

V této části dokumentu jsou popsána volání veřejného rozhraní search pro registr digitalizace, digitální konkordance a resolver.

4.1 Registr digitalizace

4.1.1 digCopyIds

Request vrátí v odezvě seznam identifikátorů všech digitálních kopií, vyhledaných podle zadaných parametrů volání. Volání umožňuje použití dvou sad parametrů. V obou případech server ověří zadané nebo sestavené FyzId v tabulce konkordancí.

Volání:

```
/search/digCopyIds?
```

Parametry:

1. hledání podle identifikátoru fyzického dokumentu

```
fyzId - identifikátor fyzického dokumentu
```

2. hledání podle lokačních údajů

```
shelfmark  - signature fyzického dokumentu
repository - repository fyzického dokumentu
settlement - settlement fyzického dokumentu
```

Odezva:

```
"result": {
     "regFyzId":"<zjištěné RegFyzId>",
     "digCopyIds": […,…,…]
}
```

4.1.2 digCopyRec

Request vrátí v odezvě kompletní záznam(y) o digitální kopii (nebo kopiích) daného fyzického dokumentu jako výsledek hledání podle zadaných parametrů volání.

Volání:

```
/search/digCopyRec?
```

Parametry:

1. Hledání podle identifikátoru digitální kopie. (Toto volání bude použito např. po získání identifikátorů pomocí requestu **digCopyIds**).

```
digCopyId – identifikátor digitální kopie dokumentu
```

2. Hledání podle identifikátoru fyzického dokumentu.

```
fyzId - identifikátor fyzického dokumentu
```

3. Hledání podle lokačních údajů

```
shelfmark  - signatura fyzického dokumentu
repository - repository fyzického dokumentu
settlement - settlement fyzického dokumentu
```

Odezva:

```
"result": {
     "regFyzId":"<zjištěné RegFyzId>",
     "digCopyRec": [{"DigCopyId":"<digCopyId>",…},{…},{…}]
}
```

4.1.3 allDigCopyIds

Request vrátí v odezvě všechny identifikátory digitálních kopií, které odpovídají zadaným lokačním údajům, přičemž povinné parametry jsou repository a settlement. Pokud není uveden parametr signatura, vyhledají se všechny dokumenty odpovídající dané repository.

Volání:

```
/search/allDigCopyIds?
```

Parametry:

```
shelfmark  - celá nebo část signatury fyzického dokumentu
repository - repository fyzického dokumentu
settlement - settlement fyzického dokumentu
firstRow – číslo záznamu, od kterého se mají záznamy předávat v
rozsahu 1..počet záznamů, není-li uvedeno, předává se od začátku (1)
rowCount - počet předaných záznamů, není-li uvedeno, předá se
vše
```

```
orderBy – pole, podle kterého se bude řadit výsledek. Lze řadit
podle shelfmark, repository nebo settlement
orderDir – vzestupně/sestupně  - ASC/DESC, není-li uvedeno řadí
se vzestupně (ASC).
```

Odezva:

```
"result": {
     "digCopyIds": [
          {"regFyzId":"<RegFyzId>","digCopyId":"<digCopyId>"},
          {…,…},
          {…,…}
     ]
}
```

4.1.4 allDigCopyRec

Request vrátí v odezvě všechny záznamy o digitálních kopiích, které odpovídají zadaným lokačním údajům, přičemž povinné parametry jsou repository a settlement. Pokud není uveden parametr signatura, vyhledají se všechny dokumenty odpovídající dané repository.

Volání:

```
/search/allDigCopyRec?
```

Parametry:

```
shelfmark  - celá nebo část signatury fyzického dokumentu
repository - repository fyzického dokumentu
settlement - settlement fyzického dokumentu
firstRow – číslo záznamu, od kterého se mají záznamy předávat v
rozsahu 1..počet záznamů, není-li uvedeno, předává se od začátku (1)
rowCount - počet předaných záznamů, není-li uvedeno, předá se
vše
orderBy – pole, podle kterého se bude řadit výsledek. Lze řadit
podle shelfmark, repository nebo settlement
orderDir – vzestupně/sestupně  - ASC/DESC, není-li uvedeno řadí
se vzestupně (ASC).
```

Odezva:

```
"result": {
     "digCopyRec": [
          {"digCopyId":"<DigCopyId>",
           "regFyzId":"<RegFyzId>",
```

```
            "shelfmark":"<signatura>",
            "repository":"<repository>",
            "settlement":"<settlement>",
           "repAbbrev":"<zkratka repository>",
           "country":"<country>",
           "author":"<autor>",
           "title":"<titul>",
           "origDate":"<datum původu>",
           "dataStorageId":"<DataStorageId>",
           "digCopyRootUrl":"<začátek digitální kopie>",
           "source":"<zdroj záznamu>",
           "permalink":"<persistentní (permanentní) identifikátor
           zdrojového záznamu>"
     },{……}, {……}]
}
```

4.1.5 getFyzId

Z lokačních údajů sestaví identifikátor fyzického dokumentu FyzId a předá ho v odezvě. Neprovádí žádná ověření, pouze sestaví identifikátor.

Volání:

```
/search/getFyzId?
```

Parametry:

```
shelfmark  - signatura fyzického dokumentu
repository - repository fyzického dokumentu
settlement - settlement fyzického dokumentu
```

Odezva:

```
"result": {
     "fyzId":"<identifikátor fyzického dokumentu>",
     "repositoryAbbrev":"<odpovídající zkratka repository>",
     "repositoryName":"<korektní název repository v knihovny.xml>",
     "settlement":"<korektní název pro settlement v knihovny.xml>",
     "country":"<země repository>"
}
```

4.1.6 verifyRepository

Ověří, zda vložené údaje o názvu místa uložení (repository a settlement) mají odpovídající zkratku repository v souboru knihovny.xml.

Volání:

```
/search/verifyRepository?
```

Parametry:

repository - repository fyzického dokumentu
settlement - settlement fyzického dokumentu

Odezva:

```
"result": {
     "repositoryAbbrev":"<odpovídající zkratka repository>",
     "repositoryName":"<korektní název repository v knihovny.xml>",
     "settlement":"<korektní název pro settlement v knihovny.xml>",
     "country":"<země repository>"
}
```

4.1.7 getRepositoryList

Request v odezvě vrátí seznam všech zkratek a názvů knihoven (repositories), jejich umístění a zemi.

Volání:

/search/getRepositoryList?

Parametry:

firstRow - číslo (1..n), od kterého se mají záznamy předat, není-li uvedeno, předává se od začátku (1)
rowCount - počet předaných záznamů, není-li uvedeno, předá se vše

Odezva:

```
"result": {
     "repositoryList":[{
          "repositoryAbbrev":"<zkratka>"
          "repositoryName":"<název repository>",
          "settlement ":"<settlement>",
          "country":"<země>",
     }]
}
```

4.2 Konkordance

4.2.1 concordRec

Volání ve své odezvě předá záznam z tabulky konkordancí, identifikovaný parametrem fyzId.

Volání:

```
/search/concordRec?
```

Parametry:

```
fyzId – identifikátor záznamu (a zároveň nový identifikátor fyzického dokumentu v tabulce konkordancí)
```

Odezva:

```
"result": {
    "regFyzId":"<RegFyzId>",
    "newFyzId":"<FyzID podle nových lokačních údajů>",
    "shelfmark":<signatura>",
    "repository:"<nová repository fyzického dokumentu>",
    "settlement:"<nový settlement fyzického dokumentu>",
    "country:"<země>",
    "olderChange:"<změna před zápisem do RDHF>",
    "changeDate:"<přibližná doba změny lokačních údajů>",
    "changeNote:"<poznámka ke změně lokačních údajů>",
    "validityFrom:"<platnost lokačních údajů OD>",
    "validityTo:"<platnost lokačních údajů DO>",
    "note:"<poznámka>"
}
```

4.2.2 getRegFyzId

Z lokačních údajů sestaví FyzId a ověří ho v tabulce konkordancí. V odezvě předá RegFyzId a jemu odpovídající identifikační (lokační) údaje. Pokud vytvořené fyzId není obsaženo v tabulce RDHF, tzn., neexistuje jako permanentní identifikátor (a také k němu neexistuje záznam o digitální kopii), vrátí chybovou informaci o neexistenci takového identifikátoru.

Volání:

```
/search/getRegFyzId?
```

Parametry:

1. `fyzId - provede se kontrola v tabulce konkordancí`

2. lokační údaje, nejprve se zjistí FyzId, poté se provede kontrola v tabulce konkordancí
   shelfmark - signatura fyzického dokumentu
   repository - repository fyzického dokumentu
   settlement - settlement fyzického dokumentu

Odezva:

```
"result": {
      "regFyzId":"<identifikátor fyzického dokumentu>",
      "repositoryAbbrev":"<odpovídající zkratka repository>",
      "repositoryName":"<korektní název repository v knihovny.xml>",
      "settlement":"<korektní název pro settlement v knihovny.xml>",
      "country":"<země repository>"
}
```

4.2.3 docLocationHistory

Request vyhledá v tabulce konkordancí veškeré záznamy o změnách umístění požadovaného fyzického dokumentu a předá tyto záznamy v odezvě. Předpoklad je, že u naprosté většiny fyzických dokumentů došlo ke změně jejich lokačních údajů v celé jejich známé historii řádově v počtu jednotek (možná někde výjimečně desítek), a proto výsledná odezva nebude nijak mimořádně dlouhá.

Volání:

```
/search/docLocationHistory?
```

Parametry:

1. fyzId - provede se vyhledání v tabulce konkordancí

2. lokační údaje (nějaké známé), nejprve se zjistí FyzId, poté se provede vyhledání v tabulce konkordancí
   shelfmark - signatura fyzického dokumentu
   repository - repository fyzického dokumentu
   settlement - settlement fyzického dokumentu

Odezva:

```
"result": {
      "regFyzId":"<zjištěné RegFyzId>",
      "regShelfmark":"<RegShelfmark>",
      "regRepository":"<RegRepository>",
      "regSettlement":"<RegSettlement>",
      "regCountry":"<RegCountry>",
      "author":"<Autor dokumentu>",
      "title":"<Název dokumentu>",
      "origDate":"<Vznik dokumentu>",
      "concordance": [{
```

```
            "fyzId":"<FyzId>",
            "shelfmark":"<Shelfmark>",
            "repository":"<Repository>",
            "settlement":"<Settlement>",
            "country":"<Country>",
            "olderChange":"<ano nebo ne (1/0)>",
            "changeDate":"<datum změny>",
            "changeNote":"<change note>",
            "validityFrom":"<ValidityFrom>",
            "validityTo":"<ValidityTo>",
            "note":"<regNote>",
            "recordAuthor":"<Autor záznamu>",
            "recordDate":"<Datum vložení záznamu>"
      },
      {…},
      {…}]
}
```

4.3 Resolver

4.3.1 gotoDigCopy

Request v odezvě vrátí všechny dostupné informace potřebné k zobrazení digitální kopie dokumentu.

Volání:

```
/search/gotoDigCopy?
```

Parametry:

```
digCopyId – identifikátor digitální kopie dokumentu
```

Odezva:

```
"result": {
      "regFyzId":"<zjištěné RegFyzId>",
      "dataStorageId":"<identifikátor datového úložiště>",
      "digCopyRoot":"<URL na začátek digitální kopie>",
      "source":"<název digitální repository (např.Manuscriptorium)>",
      "permalink":"<perzistentní odkaz na zdrojová metadata
(deskriptivní) k fyzickému dokumentu>",
      "digitizationId":"<>",
      "digLibraryId":""<>"
}
```

Poslední dvě pole záznamu odezvy se prozatím nebudou generovat (nebo zůstanou prázdná), protože v současné době nejsou tyto informace k dispozici.

4.3.2 gotoDocAllDigCopies

Request v odezvě předá všechny přístupové parametry ke všem digitálním kopiím požadovaného fyzického dokumentu.

Volání:

```
/search/gotoDocAllDigCopies?
```

Parametry:

Pro identifikaci fyzického dokumentu a tím i jeho digitální kopie jsou povoleny tři skupiny parametrů. Server při identifikaci postupuje následovně:

a. pokud existuje parametr fyzId – použije ho ke zjištění RegFyzId a ostatní identifikační parametry ignoruje
b. neexistuje-li parametr fyzId, server hledá parametr repositoryAbbrev. Pokud tento parametr existuje, server převezme parametr shelfmark a ostatní parametry ignoruje.
c. Pokud v parametrech není ani fyzId ani repositoryAbbrev, server použije třetí sadu identifikačních parametrů, a to shelfmark, repository a settlement.

1. fyzId - provede se kontrola v tabulce konkordancí

2. repositoryAbbrev - zkratka organizace (knihovny.xml) – musí odpovídat zkratce v souboru knihovny.xml
   shelfmark - signatura fyzického dokumentu

3. shelfmark - signatura fyzického dokumentu
   repository - repository fyzického dokumentu
   settlement - settlement fyzického dokumentu

Odezva:

```
"result": {
     "regFyzId":"<zjištěné RegFyzId>",
     "digCopies":[{
          "dataStorageId":"<identifikátor datového úložiště>",
          "digCopyRoot":"<URL na začátek digitální kopie>",
          "source":"<název digitální repository (např.
Manuscriptorium)> ",
          "permalink":"<perzistentní odkaz na zdrojová metadata
(deskriptivní) k fyzickému dokumentu>",
          "digitazationId":"<>", "digLibraryId":""<>"},
     {…,…},{…,…} ]
}
```

5 Funkce rozhraní manage

V této části dokumentu jsou popsány volání API pro správce dat systému RDHF. Některá tato volání jsou též určena pro volání klientskou aplikací administrátora.

5.1 Requesty pro správu systému

5.1.1 ctrlCode

Předá klientu kontrolní kód, který následně slouží k sestavení identifikátoru uživatele userId. Tento request nemá parametry.

Volání:

```
/manage/ctrlCode?
```

Odezva:

```
"result": {"ctrlCode":"<kontrolní kód>"}
```

5.2 Registr digitalizace

5.2.1 addRdhfRecord

Vloží nový záznam do hlavní tabulky registru digitalizace RDHF (tabulka digitálních kopií dokumentu).

Volání:

```
/manage/addRdhfRecord?
```

Parametry:

```
userId – identifikátor přihlášeného uživatele (base64)
```

Pro identifikaci fyzického dokumentu a tím i jeho digitální kopie jsou povoleny tři skupiny parametrů. Server při identifikaci postupuje následovně:

a. pokud existuje parametr fyzId – použije ho a ostatní identifikační parametry ignoruje
b. neexistuje-li parametr fyzId, server hledá parametr repositoryAbbrev. Pokud tento parametr existuje, server převezme parametr shelfmark a ostatní parametry ignoruje.

c. Pokud v parametrech není ani fyzId ani repositoryAbbrev, server použije třetí sadu identifikačních parametrů, a to shelfmark, repository a settlement.

Identifikační údaje:

1. fyzId - provede se kontrola v tabulce konkordancí
   shelfmark - signatura fyzického dokumentu

2. repositoryAbbrev - zkratka organizace (knihovny.xml) – musí odpovídat zkratce v souboru knihovny.xml
   shelfmark - signatura fyzického dokumentu

3. shelfmark - signatura fyzického dokumentu
   repository - repository fyzického dokumentu
   settlement - settlement fyzického dokumentu

Další parametry:

country – země umístění fyzického dokumentu
author - informace o autoru(ech) dokumentu
title – titul (tituly) dokumentu
origDate – datace vzniku dokumentu
digCopyRoot – URL na začátek digitální kopie
source – název digitální repository (např. Manuscriptorium)
permalink - perzistentní odkaz na zdrojová metadata (deskriptivní) k fyzickému dokumentu

Odezva:

```
"result": {
     "regFyzId":"<RegFyzId>",
     "digCopyId":"<vytvořený identifikátor digitální kopie>"
}
```

5.2.2 updateRdhfRecord

Request provede aktualizaci požadovaného záznamu v hlavní tabulce RDHF (tabulka digitálních kopií dokumentů). Jako odezvu vrátí kompletní novou verzi záznamu.

Volání:

/manage/updateRdhfRecord?

Parametry:

userId - identifikátor přihlášeného uživatele (base64)
digCopyId – identifikátor digitální kopie záznamu – identifikuje záznam, který se bude aktualizovat

Parametry, které lze v záznamu aktualizovat:

shelfmark – signatura – můžeme upravit pouze formálně, tj. změnit pouze nevýznamné znaky, např. 49/70 změnit na 49-70 apod. Při úpravě signatury NESMÍ dojít ke změně FyzId (tím by muselo dojít i ke změně identifikátoru záznamu digCopyId)

author - informace o autoru(ech) dokumentu

title – titul (tituly) dokumentu

origDate – datace vzniku dokumentu

digCopyRoot – URL na začátek digitální kopie

source – název digitální repository (např. Manuscriptorium)

permalink - perzistentní odkaz na zdrojová metadata (deskriptivní) k fyzickému dokumentu

Parametry repository, settlement a country se aktualizují automaticky podle informací v souboru knihovny.xml. Tak dojde automaticky k nahrazení „alternate" názvu za „correct" (podle knihovny.xml) apod. Vstupní parametry, které nebudou uvedeny, neovlivní původní obsah příslušných polí záznamu.

Odezva:

```
"result": {
      "digCopyRec": {                        // Upravený RDHF record
            "digCopyId":"<DigCopyId>",
            "regFyzId":"<RegFyzId>",
            …
            …
      }
}
```

5.2.3 deleteRdhfRecord

Request smaže požadovaný záznam v Registru digitalizace historických fondů (tabulka digitálních kopií dokumentů). V odezvě vrátí identifikátor (RegFyzId) smazaného záznamu. Smazání záznamu je nevratné.

Volání:

/manage/deleteRdhfRecord?

Parametry:

userId - identifikátor přihlášeného uživatele (base64)

digCopyId – identifikátor digitální kopie záznamu – identifikuje záznam, který se má smazat

Odezva:

```
"result": {
      "digCopyId":"<digCopyId>"   // identifikátor smazaného záznamu
}
```

5.3 Konkordance

5.3.1 addConcordRecord

Request přidá záznam do tabulky konkordancí. Vrátí nově vytvořené informace – NewFyzId a autor a datum vložení záznamu.

Volání:

```
/manage/addConcordRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
```

Pro identifikaci záznamu lze použít tři skupiny parametrů:

a) RegFyzId – tento identifikátor je známý, v tabulce RDHF musí být záznam s daným persistentním identifikátorem, jinak SE NESMÍ vložit záznam do tabulky konkordancí. Před vložením záznamu do tabulky konkordancí se VŽDY ověří existence záznamu v RDHF.
b) Zkratka repository odpovídající souboru knihovny.xml a signatura. Sestaví se FyzId a následně se stejně jako a) ověří existence záznamu v RDHF.
c) Signatura, repository a settlement. Sestaví se FyzId a stejně jako a) ověří existence záznamu v RDHF.

Pozn.: Protože při vytváření záznamu do tabulky konkordancí je vždy známý identifikátor RegFyzID, skupiny parametrů b) a c) jsou pouze doplňkové a neměly by být primárně používány.

```
1. regFyzId

2. regRepositoryAbbrev
   regShelfmark

3. regShelfmark
   regRepository
   regSettlement
```

Další parametry:

```
shelfmark – nová signatura fyzického dokumentu
repository – nová repository fyzického dokumentu
settlement – nový settlement fyzického dokumentu
country – země
```

olderChange – 0/1 – 1= změna vznikla před zápisem do RDHF
changeDate – přibližná doba změny lokačních údajů
changeNote – poznámka ke změně lokačních údajů
validityFrom – platnost lokačních údajů OD
validityTo – platnost lokačních údajů DO
note - poznámka

Autor a čas vložení záznamu se vloží automaticky podle systémového datumu a přihlášeného autora.

Odezva:

```
"result": {
      "regFyzId":"<RegFyzId>",
      "newFyzId":"<FyzID podle nových lokačních údajů>
}
```

5.3.2 updateConcordRecord

Request provede aktualizaci požadovaného záznamu v tabulce konkordancí. Jako odezvu vrátí kompletní novou verzi záznamu.

Volání:

/manage/updateConcordRecord?

Parametry:

userId - identifikátor přihlášeného uživatele (base64)
newFyzId – identifikátor záznamu (a zároveň nový identifikátor fyzického dokumentu)

Lokační údaje jako je repository, settlement a country jsou podobně jako v případě aktualizace záznamu Rdhf aktualizovány automaticky vložením korektních údajů ze souboru knihovny.xml.
Parametry, které lze v záznamu aktualizovat:

shelfmark – podobně jako u Rdhf lze dělat pouze formální změny v signatuře, tj. nahrazovat nevýznamné znaky apod.
olderChange – ano/ne – změna vznikla před zápisem do RDHF
changeDate – přibližná doba změny lokačních údajů
changeNote – poznámka ke změně lokačních údajů
validityFrom – platnost lokačních údajů OD
validityTo – platnost lokačních údajů DO
note - poznámka

Parametry repository a settlement budou použity pro vytvoření zkratky organizace (knihovny, vlastníka…), která MUSÍ BÝT shodná s původní zkratkou repository v záznamu! Editace těchto polí tedy umožní pouze formální změnu lokačních údajů, jako je např. nahrazení „alternate" názvu za „correct" (podle knihovny.xml) apod.

Autor a čas aktualizace záznamu se vloží automaticky podle systémového datumu a přihlášeného autora. Parametry, které nebudou uvedeny, neovlivní původní obsah příslušných polí záznamu.

Odezva:

```
"result": {
      "regFyzId":"<RegFyzId>",
      "newFyzId":"<FyzID podle nových lokačních údajů>",
      "shelfmark":<signatura>",
      "repository:"<nová repository fyzického dokumentu>",
      "settlement:"<nový settlement fyzického dokumentu>",
      "country:"<země>",
      "olderChange:"<změna před zápisem do RDHF>",
      "changeDate:"<přibližná doba změny lokačních údajů>",
      "changeNote:"<poznámka ke změně lokačních údajů>",
      "validityFrom:"<platnost lokačních údajů OD>",
      "validityTo:"<platnost lokačních údajů DO>",
      "note:"<poznámka>"
}
```

5.3.3 deleteConcordRecord

Request smaže požadovaný záznam v tabulce konkordancí a v odezvě vrátí identifikátor FyzId smazaného záznamu. Smazání záznamu je nevratné.

Volání:

```
      /manage/deleteConcordRecord?
```

Parametry:

```
      userId - identifikátor přihlášeného uživatele (base64)
      newFyzId – identifikátor fyzického dokumentu, který
identifikuje (změnový) záznam v tabulce konkordancí
```

Odezva:

```
"result": {
      "fyzId":"<NewFyzId>"  // identifikátor smazaného záznamu
}
```

5.3.4 getDataStorageList

Request předá v odezvě kompletní seznam datových úložišť (nebo jeho požadovanou část). Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (viz rozhraní „admin" a request „serverAccess").

Volání:

```
/manage/getDataStorageList?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
firstRow – číslo záznamu, od kterého se mají záznamy předávat v rozsahu 1..počet záznamů, není-li uvedeno, předává se od začátku (1)
rowCount - počet předaných záznamů, není-li uvedeno, předá se vše
```

Odezva:

```
"result": {"dataStorageList" :[
      {"dataStorageId":"<identifikátor datového úložiště>",
       "dataStorageURL":"<URL datového úložiště>"
      }, {…,…}, … … ]
}
```

5.3.5 addDataStorageRecord

Request přidá záznam do tabulky datových úložišť. Jako odezvu vrátí kompletní novou verzi záznamu. Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (Viz rozhraní „admin" a request „serverAccess").

Volání:

```
/manage/addDataStorageRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
dataStorageId – identifikátor datového úložiště
dataStorageUrl – URL datového úložiště
```

Odezva:

```
"result": {"dataStorageId":"<identifikátor datového úložiště>",
       "dataStorageURL":"<URL datového úložiště>"
}
```

5.3.6 updateDataStorageRecord

Request provede aktualizaci požadovaného záznamu v tabulce datových úložišť. Jako odezvu vrátí kompletní novou verzi záznamu. Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat

s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (Viz rozhraní „admin" a request „serverAccess").

Volání:

```
/manage/updateDataStorageRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
dataStorageId – identifikátor datového úložiště = záznamu, kde bude aktualizováno URL edatového úložiště
dataStorageUrl – (nové) URL datového úložiště
```

Odezva:

```
"result": {"dataStorageId":"<identifikátor datového úložiště>",
    "dataStorageURL":"<URL datového úložiště>"
}
```

5.3.7 deleteDataStorageRecord

Request smaže požadovaný záznam v tabulce datových úložišť. Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (Viz rozhraní „admin" a request „serverAccess").

Volání:

```
/manage/deleteDataStorageRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
dataStorageId – identifikátor datového úložiště, které bude smazáno
```

Odezva:

```
"result": {"dataStorageId":"<identifikátor datového úložiště, které
    bylo smazáno>"}
```

6 Funkce rozhraní admin

Tato část dokumentu popisuje volání rozhraní admin, které je určeno klientské aplikaci pro administrátora k spravování serveru RDHF.

6.1 ctrlCode

Předá klientu kontrolní kód, který následně slouží k sestavení identifikátoru uživatele userId. Tento request nemá parametry. Volání tohoto requestu z rozhraní admin umožní (administrátoru) pracovat se serverem i v případě, že je server RDHF pro ostatní uživatele nedostupný (viz request „serverAccess").

Volání:

```
/admin/ctrlCode?
```

Odezva:

```
"result": {"ctrlCode":"<kontrolní kód>"}
```

6.2 dataImport

Request provede import dat do tabulky digitálních kopií z požadovaného zdroje. V současné době je k dispozici poze jeden zdroj importních dat a tím je systém Manuscriptorium. Obsluha requestu, tedy import dat do RDHF, probíhá na serveru ve zvláštním vláknu. Právě probíhající import lze zastavit prostřednictvím requestu „stopDataImport". Informace o právě probíhajícím importu může klientská aplikace získávat pravidelným voláním requestu „importStatus". Podrobně je tento mechanismus popsán v [1].

Volání:

```
/admin/dataImport?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora (base64)
```

Odezva:

```
"result":"Data import ze zdroje Manuscriptorium byl zahájen … … "
```

6.3 stopDataImport

Request zastaví provádění právě spuštěného importu dat.

Volání:

```
/admin/stopDataImport?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora (base64)
```

Odezva:

```
"result":"Data import ze zdroje Manuscriptorium byl přerušen uživatelem v … …"
```

6.4 importStatus

Předá informace o stavu importu od začátku nebo od okamžiku posledního předání informací requestem „importStatus" do okamžiku nového volání tohoto requestu. Server RDHF při importu dat do tabulky digitálních kopií vytváří množinu informací o stavu importu jednotlivých záznamů, kterou předá jako odezvu na request „importStatus". Po odeslání těchto informací server začne vytvářet novou množinu informací o importu, kterou předá jako odezvu při následujícím requestu „importStatus". Klientská aplikace by tak měla periodicky volat request „import status" pro získání kompletního logu importu. Podrobně je tento mechanismus popsán v [1].

Volání:

```
/admin/importStatus?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora (base64)
```

Odezva:

```
"result": {"items":[
     {"status":"<status (stav procesu – začátek, konec, přerušení, číslo chyby při zpracování záznamu apod.)>",
      "type":"<typ informace, id vstupního záznamu apod.>",
      "text":"<vlastní text informace>"
     }, {… …}, {… …}, … ]
}
```

6.5 reloadConfig

Zajistí znovunačtení konfiguračního souboru serveru za běhu. To umožní změnit některé parametry systému bez nutnosti restartovat server RDHF. Protože

systém je bezestavový (implementuje rozhraní REST), změny v konfiguraci serveru se projeví bezprostředně po změně při následujícím volání serveru.

Volání:

```
/admin/reloadConfig?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
```

Odezva:

```
"result":{"sqlDriverClassName":"<SQL driver name>",
          "sqlDatabaseName":"<jméno databáze>",
          "repositoryFtp":"<FTP adresa (cesta) k souboru
          knihovny.xml >",
          "localWorkFile":"<cesta k lokálnímu (pracovnímu) souboru
          knihovny.xml na serveru RDHF>",
          "workFileDate":"<datum a čas aktuálního pracovního
          souboru knihovny.xml>"
}
```

6.6 reloadRepositoryList

Zajistí znovunačtení souboru knihovny.xml. Server RDHF kontroluje automaticky aktuálnost pracovního souboru knihovny.xml v pravidelných intervalech dle nastavení (nyní jedna hodina) a pokud došlo od poslední kontroly ke změně v referenčním souboru knihovny.xml provede aktualizaci pracovního souboru. Pokud se ale referenční (zdrojový) soubor změní a vznikne naléhavá potřeba aktualizovat pracovní soubor serveru okamžitě, lze tuto jinak automaticky zajišťovanou činnost provést ručně spuštěním requestu „reloadRepositoryList.“ Změna v pracovním souboru knihovny.xml se projeví bezprostředně v následujícím requestu.

Volání:

```
/admin/reloadRepositoryList?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
```

Odezva:

```
"result": {"repositoryFtp":"<FTP adresa (cesta) k souboru
knihovny.xml >",
          "localWorkFile":"<cesta k lokálnímu (pracovnímu) souboru
          knihovny.xml na serveru RDHF>",
```

```
            "workFileDate":"<datum a čas aktuálního pracovního
            souboru knihovny.xml>"
}
```

6.7 version

Request předá informaci o verzi serveru a datumu jejího vydání.

Volání:

```
/admin/version?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
```

Odezva:

```
"result": {"version":"<verze systému>",
           "date":"<datum verze>"
}
```

6.8 serverAccess

Request umožní povolit či zakázat přístup všem uživatelům (kromě administrátora) k běžícímu serveru. Zakázání přístupu uživatelům umožní administrátoru provádět servisní práce na serveru RDHF (např. provádět změny v nastavení, zálohování dat apod.). Tyto činnosti (především zálohování či optimalizace dat) potom nebudou ovlivněny prací uživatelů s daty RDHF. Jako odezvu server vrátí aktuální stav po provedení requestu.

Volání:

```
/admin/serverAccess?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
enable – povolení či zakázání přístupu ["true"/"false"]
```

Odezva:

```
"result\":{"serverEnabled":"<true/false>"}
```

6.9 getServerAccess

Request zjistí aktuální stav serveru RDHF, tj. zdali je či není server RDHF přístupný uživateli.

Volání:

```
/admin/getServerAccess?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
```

Odezva:

```
"result\":{"serverEnabled":"<true/false>"}
```

6.10serverInfo

Předá dostupné informace o nastaveních (konfiguraci) serveru RDHF.

Volání:

```
/admin/serverInfo?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
```

Odezva:

```
"result": {"serverEnabled":"<true/false>",
          "version":"<verze systému>",
          "date":"<datum verze>",
          "sql": {
               "driverClassName":"<SQL driver name>",
               "databaseName":"<jméno databáze>"
               },
          "repositoryList":{
               "ftp":"<FTP adresa souboru knihovny.xml >",
               "localWorkFile":"<cesta k lokálnímu (pracovnímu)
          souboru knihovny.xml na serveru RDHF>",
               "workFileDate":"<datum a čas aktuálního pracovního
          souboru knihovny.xml>"
          }
}
```

6.11 getDataStorageList

Request předá v odezvě kompletní seznam datových úložišť (nebo jeho požadovanou část). Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě,

že server RDHF není přístupný běžnému uživateli (viz rozhraní „admin" a request „serverAccess").

Volání:

```
/admin/getDataStorageList?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele = administrátora
(base64)
firstRow – číslo záznamu, od kterého se mají záznamy předávat v
rozsahu 1..počet záznamů, není-li uvedeno, předává se od začátku (1)
rowCount - počet předaných záznamů, není-li uvedeno, předá se
vše
```

Odezva:

```
"result": {"dataStorageList" :[
     {"dataStorageId":"<identifikátor datového úložiště>",
      "dataStorageURL":"<URL datového úložiště>"
     }, {…,…}, … … ]
}
```

6.12 addDataStorageRecord

Request přidá záznam do tabulky datových úložišť. Jako odezvu vrátí kompletní novou verzi záznamu. Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (Viz rozhraní „admin" a request „serverAccess").

Volání:

```
/admin/addDataStorageRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
dataStorageId – identifikátor datového úložiště
dataStorageUrl – URL datového úložiště
```

Odezva:

```
"result": {"dataStorageId":"<identifikátor datového úložiště>",
      "dataStorageURL":"<URL datového úložiště>"
}
```

6.13 updateDataStorageRecord

Request provede aktualizaci požadovaného záznamu v tabulce datových úložišť. Jako odezvu vrátí kompletní novou verzi záznamu. Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (Viz rozhraní „admin" a request „serverAccess").

Volání:

```
/admin/updateDataStorageRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
dataStorageId – identifikátor datového úložiště = záznamu, kde
bude aktualizováno URL edatového úložiště
dataStorageUrl – (nové) URL datového úložiště
```

Odezva:

```
"result": {"dataStorageId":"<identifikátor datového úložiště>",
      "dataStorageURL":"<URL datového úložiště>"
}
```

6.14 deleteDataStorageRecord

Request smaže požadovaný záznam v tabulce konkordancí. Tento request je možno volat z rozhraní manage a admin. S použitím rozhraní admin je možno pracovat s databází datových úložišť i v případě, že server RDHF není přístupný běžnému uživateli (Viz rozhraní „admin" a request „serverAccess"). Smazání záznamu je nevratné.

Volání:

```
/admin/deleteDataStorageRecord?
```

Parametry:

```
userId - identifikátor přihlášeného uživatele (base64)
dataStorageId – identifikátor datového úložiště, které bude
smazáno
```

Odezva:

```
"result": {"dataStorageId":"<identifikátor datového úložiště>"}
```