SLUB
Wir führen Wissen.

Webservice SLUBArchiv

SLUB Dresden

Version 1.0.4, 2022-02-09

Inhaltsverzeichnis

Release Note 1.0.4 vom 2022-02-09

- Initiale Version
- Ergänzung HTTP Response Codes
- Ergänzung heartbeat
- Klarstellung zu Datum
- Erläuterung zu 409 ergänzt
- Hinweis auf Retry-After Header
- Kleine grammatikalische Korrekturen

Überblick

Für jeden Dienstnehmer wird unter der zugeordneten IP (mit Hostname) ein Webservice angeboten, mit dem er den Status seiner eingelieferten SIPs abfragen kann.

Der Webservice bietet zurzeit folgende Schnittstellen an:

- Auskunft über den Archivierungsstatus eines IEs

Der Webservice arbeitet synchron, dies bedeutet, Anfragen wären sofort verarbeitet und zurückgemeldet. Sollte eine Abfrage des Langzeitarchivsystems nicht möglich sein oder der Webservice nicht arbeiten, so erfolgt eine Fehlermeldung über einen TimeOut. Die Anfragen können dann zu einem späteren Zeitpunkt wiederholt werden.

Dieses Dokument ist ein Teil der Übernahmespezifikation für das SLUBArchiv. Zur Übernahmespezifikation gehören die folgenden Dokumente [1]:

- In der **Übernahmevereinbarung** zwischen SLUB und Dienstnehmer sind die Daten, Ansprechpartner und organisatorischen Randbedingungen beschrieben. Dies schliesst die zu verwendenden Handreichungen für Dateiformate bzw. Objektgruppen mit ein.
- In der **Langzeitarchivfähige Dateiformate** sind die Formate aufgeführt, die die SLUB als potenziell archivfähig bewertet und für die die Funktionalitäten der Formaterkennung, Formatvalidierung und Metadatenextraktion in einem ausreichenden Maß durch das SLUBArchiv gewährleistet werden könnten. Die genaue Festlegung erfolgt spezifisch für jeden Workflow und jede Objektgruppe auf Basis der ermittelten signifikanten Eigenschaften.
- Die **SIP Spezifikation für automatischen Ingest SLUBArchiv** beschreibt den Aufbau der Ablieferungspakete (englisch: Submission Information Package, SIP) mit denen der Dienstnehmer die zu archivierenden Dokumente für das SLUBArchiv bereitstellt.
- Die **DIP Spezifikation für automatischen Access SLUBArchiv** beschreibt den Aufbau eines Auslieferungspaketes (englisch: Dissemination Information Package, DIP), welches für die automatische Weiterverarbeitung zielgruppengerechter Ausspielungen (Access) von im SLUBArchiv archivierter digitaler Datenobjekte (IE) geeignet ist
- Die **Workflow Spezifikation für automatisierte Interaktionen mit dem SLUBArchiv** beschreibt den Prozess der Übergabe zu archivierender Dokumente in das SLUBArchiv (Ingest / AIP Update), das Fehlerprotokoll und den Zugriff auf die archivierten digitaler Objekte (Access).

- Das Dokument **Spezifikation Rechteauszeichnung SLUBArchiv** beschreibt, wie rechtliche Informationen zu einem Datenobjekt kodiert und abgelegt werden müssen.
- Das Dokument **Webservice SLUBArchiv** beschreibt Funktionen, die Dienstnehmer nutzen können, um Informationen über ihre Daten im SLUBArchiv über einen Webservice abzufragen.
- Vom SLUBArchiv verwendete Begriffe sind im **Glossar SLUBArchiv** definiert.

Auskunft über den Archivierungsstatus

Mit dieser Schnittstelle kann abgefragt werden, ob ein abgeliefertes Dokument mit einer bestimmten externen ID und aus einem bestimmten Workflownamen im Langzeitarchiv erfolgreich archiviert wurde. ID und Workflow entsprechen den Werten der Tags 'externalWorkflow' und 'externalId', die in der SIP-Spezifikation beschrieben sind.

Im Fall einer positiven Antwort wird ein Datum mitgeliefert, das dem 'exportToArchiveDate' - Datum entspricht (siehe SIP-Spezifikation). Damit kann geprüft werden, ob eine SIP mit einem AIP Update erfolgreich war.

Beispiel 1. Zeitlicher Ablauf

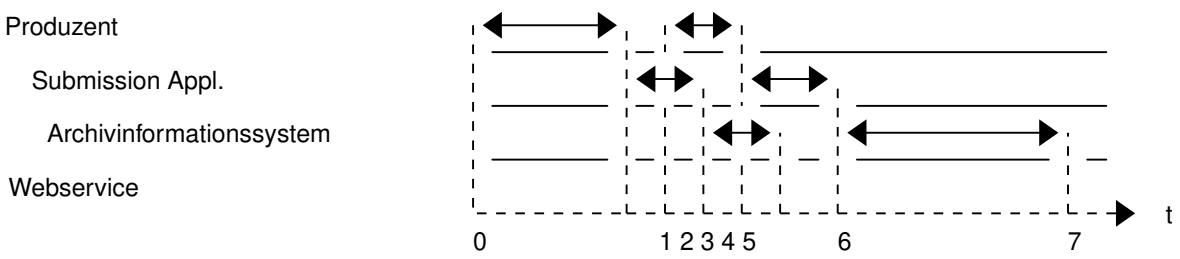

Abbildung 1. Zeitdiagramm

Hier liefert der Produzent zum Zeitpunkt t0 eine SIP mit dem exportToArchiveDate t0 aus. Die Submissionapplication beginnt diese zum Zeitpunkt t1 zu prozessieren. Zum Zeitpunkt t2 liefert der Produzent eine weitere SIP mit dem exportToArchiveDate t2 aus, welche zu t4 von der Submissionapplication verarbeitet wird.

Zu den Zeitpunkten t3 und t6 übergibt die Submissionapplication diese SIPs an das Archivinformationssystem. Dieses prüft die SIPs, reichert diese an und erzeugt ein entsprechendes AIP zu den Zeitpunkten t5 und t7.

Dabei hat das AIP zum Zeitpunkt t5 das exportToArchiveDate t0, und das AIP, welches zum Zeitpunkt t7 gespeichert wurde, das exportToArchiveDate t2.

Daraus folgt, dass die erste SIP vom Produzenten erst zum Zeitpunkt t5, das zweite SIP erst zum Zeitpunkt t7 gelöscht werden darf.

Sollten vom Produzenten mehrere SIPs mit gleicher externalID, aber unterschiedlichen exportToArchiveDate abgeliefert worden sein, so meldet der

> Webservice dasjenige exportToArchiveDate-Datum zurück, welches dem letzten, vom SLUBArchiv **vollständig erfolgreich** archiviertem Dokument entspricht.

Beispiel 2. SIP Versionen

Wird ein bestimmtes Dokument in Version 1 mit exportToArchiveDate = t0 eingeliefert, und der Produzent liefert, während diese Version im SLUBArchiv noch prozessiert wird, eine zweite Version mit exportToArchiveDate = t1 ein, dann würde der Webservice keinen Treffer zurückmelden.

Ist Version 1 bereits erfolgreich archiviert, aber Version 2 noch in der Verarbeitung, dann wird der Webservice den Wert t0 zurückliefern, da dieser dem exportToArchiveDate der Version 1 des Dokumentes entspricht.

Ist dann auch Version 2 erfolgreich verarbeitet und archiviert, dann wird der Webservice den Wert t1 liefern.

Anfrageparameter

Die Abfrage benötigt HTTP GET-Request mit folgenden Parametern:

- external Workflow: workflow=<externalWorkflow>
- external ID: id=<externalId>

XML Anfrage, mit positiver Antwort

Beispiel Anfrage für Workflow "ingestTestSC00240199" und ID "201608190001"

```
$> lynx "http://${hostname}:3000/is_archived.xml?workflow=ingestTestSC00240199&id=201608190001"
```

Beispiel Antwort in XML für Workflow "ingestTestSC00240199" und ID "201608190001"

```
<slubarchiv version="1.0">
  <in_permanent>true</in_permanent>
  <last_archival_date>20160819T130000.00</last_archival_date>
</slubarchiv>
```

XML Anfrage, mit negativer Antwort

Beispiel Anfrage für Workflow "ingestTestSC00240199" und nichtexistierende ID "999999999999999999999999"

```
$> lynx "http://${hostname
}:3000/is_archived.xml?workflow=ingestTestSC00240199&id=999999999999999999999999"
```

Beispiel Antwort in XML für Workflow "ingestTestSC00240199" und ID "201608190001"

```
<slubarchiv version="1.0">
 <in_permanent>false</in_permanent>
</slubarchiv>
```

JSON Anfrage, mit positiver Antwort

Beispiel Anfrage für Workflow "ingestTestSC00240199" und ID "201608190001"

```
$> lynx "http://${hostname}:3000/is_archived.json?workflow=ingestTestSC00240199&id=201608190001"
```

Beispiel Antwort in JSON für Workflow "ingestTestSC00240199" und ID "201608190001"

```
{
 "last_archival_date":"20160819T130000.00",
 "in_permanent":true,
 "version":"1.0"
}
```

JSON Anfrage, mit negativer Antwort

Beispiel Anfrage für Workflow "ingestTestSC00240199" und nichtexistierende ID "999999999999999999999999"

```
$> lynx "http://${hostname
}:3000/is_archived.json?workflow=ingestTestSC00240199&id=999999999999999999999999"
```

Beispiel Antwort in JSON für Workflow "ingestTestSC00240199" und ID "201608190001"

```
{
 "in_permanent":false,
 "version":"1.0",
 "last_archival_date":null
}
```

Heartbeat

Über die GET-Anfrage auf '/heartbeat' kann geprüft werden, ob der WebService zur Verfügung steht.

Beispiel Anfrage für Heartbeat mittels curl

```
$> curl -v "http://${hostname}:3000/heartbeat"
```

Der Service antwortet im Erfolgsfall mit dem HTTP Response Code 204.

Beispiel Antwort für Heartbeat, Ausgabe curl

```
...
* Connected to localhost (127.0.0.1) port 3000 (#0)
> GET /heartbeat HTTP/1.1
> Host: localhost:3000
> User-Agent: curl/7.64.0
> Accept: */*
>
< HTTP/1.1 204 No Content
< Server: Mojolicious (Perl)
< Date: Fri, 12 Feb 2021 14:52:04 GMT
< Content-Type: text/html;charset=UTF-8
<
* Connection #0 to host localhost left intact
```

HTTP Response Codes

Der Webservice implementiert nach RFC7231 [https://datatracker.ietf.org/doc/html/rfc7231] folgende HTTP Response Codes:

- 200 - Ok
- 204 - No content (für '/heartbeat', zeigt nur an, ob der Webservice "lebt")
- 400 - Bad request (für falsche Aufruf-Parameter)
- 404 - Not found (für falsche Address-Parameter)
- 409 - Conflict (wenn es Konflikte im SLUBArchiv gibt)
- 429 - Too many requests (wenn das Archivinformationssystem zu hohe Last hat)
- 500 - Internal server error (sollte nicht vorkommen)
- 503 - Service not available (wenn das Archivinformationssystem zurzeit nicht verfügbar ist)

Alle Anfragen, die nicht mit einem Code 200 bzw. 204 beantwortet werden, werden vom Webservice verworfen, d.h. der Client muss diese später erneut stellen.

Der vorhandene 'Retry-After' Header sollte ausgewertet werden, siehe auch RFC7231 [https://datatracker.ietf.org/doc/html/rfc7231#section-7.1.3].

Im Falle von 503 sollte der Webservice die nächsten Stunden nicht angefragt werden.

Im Falle von 429 sollte die Anzahl der Anfragen an den Webservice pro Zeiteinheit gedrosselt werden. Es gibt im Wesentlichen zwei Ursachen:

1. die Archivsoftware Rosetta ist unter zu hoher Last
2. der Webservice wird zu schnell hintereinander nach der gleichen ID gefragt

Der Grund für den 429 wird als Textantwort zurückgegeben.

Im Falle von 409 arbeitet das Team des SLUBArchiv an einer Lösung. Eine erneute Anfrage innerhalb mehrerer Tage ist nicht sinnvoll.

[1] Die genannten Dokumente sind auf der Webseite des SLUBArchivs unter https://slubarchiv.slub-dresden.de/technische-standards-fuer-die-ablieferung-von-digitalen-dokumenten/ veröffentlicht und sind dort, technisch bedingt, spezifischer benannt.