

# Bereitstellung von IIIF-Bildern im EDM

## Einleitung

Grundlegend können bei der Lieferung von **bildlichen** Ressourcen an den Kulturpool in Bezug auf IIIF exakt **drei** Szenarien auftreten, wenn Metadaten bereitgestellt werden. Die Szenarien sind aufeinander aufgebaut, steigen in ihrer Mehrwertigkeit, aber dementsprechend auch in ihrer Komplexität.

Die drei Szenarien lauten wie folgt:

- Es werden Ressourcen zur Verfügung gestellt, die nicht dem IIIF Protokoll entsprechen.
- Es werden IIIF Ressourcen *ohne* IIIF Manifest zur Verfügung gestellt und die Kompatibilität mit der IIIF Image API wird deklariert.
- Es werden IIIF Ressourcen *mit* IIIF Manifest zur Verfügung gestellt und die Kompatibilität mit der IIIF Image API wird deklariert.

Vorweg sei erwähnt, dass es möglich ist die genannten Szenarien theoretisch innerhalb einer Auslieferung miteinander zu kombinieren (auch wenn dies sehr selten passieren wird). So ist es z. B. vollkommen konform die folgenden Szenarien im Rahmen eines ausgespielten Objektes zu verbinden:

- Es werden Ressourcen zur Verfügung gestellt, die nicht dem IIIF Protokoll entsprechen.
- Es werden IIIF Ressourcen *mit* IIIF Manifest zur Verfügung gestellt und die IIIF Kompatibilität wird deklariert.

In Folge wird die korrekte technische Umsetzung des jeweiligen Szenarios und die zugehörigen Resultate im Sinne eines Mehrwertes geschildert.

# Ressourcen die *nicht* dem IIF Protokoll entsprechen

## Ausführung

- Eine oder mehrere Ressourcen der Klasse `edm:WebResource` zeigen die Existenz von bildlichen Inhalten an.
- Auf die deklarierten Ressourcen wird in anderen Feldern des EDM (z.B. `edm:isShownBy`) verwiesen.

## Beispiel

### Deklaration

```
<rdf:RDF [...]>
  [...]
  <edm:WebResource rdf:about="https://beispiel-institution.at/bild-1.jpg">
    [] ... ]
  </edm:WebResource>

  <edm:WebResource rdf:about="https://beispiel-institution.at/bild-2.jpg">
    [] ... ]
  </edm:WebResource>

  <edm:WebResource rdf:about="https://beispiel-institution.at/bild-3.jpg">
```

```
    [ ... ]  
    </edm:WebResource>  
  [ ... ]  
</rdf:RDF>
```

## Referenz

```
<rdf:RDF [...]>  
  [ ... ]  
  <ore:Aggregation [...]>  
    <edm:isShownBy rdf:resource="https://beispiel-institution.at/bild-1.jpg"/>  
    <edm:hasView rdf:resource="https://beispiel-institution.at/bild-2.jpg"/>  
    <edm:hasView rdf:resource="https://beispiel-institution.at/bild-3.jpg"/>  
  [ ... ]  
  [ ... ]  
  </ore:Aggregation>  
  [ ... ]  
</rdf:RDF>
```

## Resultat

- Der Kulturpool generiert auf Basis der drei bereitgestellten Ressourcen ein basales IIIF-Manifest und beliefert damit den Viewer des Kulturpools.
- Auf der Detailseite des beispielhaften Objektes ist es möglich durch die drei bereitgestellten Bilder zu blättern.
- Der Viewer weist grundsätzliche Funktionalitäten wie z. B. Rotation und Zoom auf.
- Die Auflösung des Bildes ist festgelegt und wird - z. B. im Zuge des Zoomens - *nicht* adaptiert.

# Ressourcen die dem IIIF Protokoll entsprechen

## IIIF Ressourcen *ohne* Manifest und mit Deklaration der IIIF Image API

### Ausführung

#### Wie oben ...

- Eine oder mehrere Ressourcen der Klasse `edm:WebResource` zeigen die Existenz von bildlichen Inhalten an.
- Auf die deklarierten Ressourcen wird in anderen Feldern des EDM (z.B. `edm:isShownBy`) verwiesen.

#### Darüberhinaus ...

- Die angeführten bildlichen Inhalte werden über eine IIIF Image API ausgeliefert.
- Die URL der Ressourcen führt jeweils zu einem über das IIIF-Protokoll passend ausgelieferten Bild.
- Die zugrundeliegende IIIF API wird als Ressource mit `<svcs:Service>` deklariert.
  - Innerhalb der `<svcs:Service>` Ressource wird das Level der bereitgestellten Image API via `<doap:implements>` deklariert.
  - Innerhalb der Ressource wird immer `<dcterms:conformsTo`  
`rdf:resource="http://iiif.io/api/image"/>`, d.h. der IIIF Standard ausgewiesen.

- Webressourcen, die bildliche Inhalte über das IIIF Protokoll ausliefern referenzieren mit `<svcs:has_service>` die Basis URI des jeweiligen Bildes.
- Der Identifier für `<svcs:Service>` (und dementsprechend auch in Verwendung bei `svcs:has_service`) ist die Basis URI des Bildes.

# Beispiel

## Deklaration

```
<rdf:RDF [...]>
  [...]
    <edm:WebResource rdf:about="https://beispiel-institution.at/iiif-service/bild-1/full/max/0/default.jpg">
      <svcs:has_service rdf:resource="https://beispiel-institution/iiif-service/bild-1/">
        [...] ... ]
    </edm:WebResource>

    <edm:WebResource rdf:about="https://beispiel-institution.at/iiif-service/bild-2/full/max/0/default.jpg">
      <svcs:has_service rdf:resource="https://beispiel-institution/iiif-service/bild-2/">
        [...] ... ]
    </edm:WebResource>

    <edm:WebResource rdf:about="https://beispiel-institution.at/iiif-service/bild-3/full/max/0/default.jpg">
      <svcs:has_service rdf:resource="https://beispiel-institution/iiif-service/bild-3/">
        [...] ... ]
    </edm:WebResource>

    <svcs:Service rdf:about="https://beispiel-institution/iiif-service/bild-1/">
      <dcterms:conformsTo rdf:resource="http://iiif.io/api/image/">
      <doap:implements rdf:resource="http://iiif.io/api/image/2/level1.json"/>
    </svcs:Service>

    <svcs:Service rdf:about="https://beispiel-institution/iiif-service/bild-2">
      <dcterms:conformsTo rdf:resource="http://iiif.io/api/image/">
      <doap:implements rdf:resource="http://iiif.io/api/image/2/level1.json"/>
    </svcs:Service>

    <svcs:Service rdf:about="https://beispiel-institution/iiif-service/bild-3">
      <dcterms:conformsTo rdf:resource="http://iiif.io/api/image/">
```

```
<doap:implements rdf:resource="http://iiif.io/api/image/2/level1.json"/>
</svcs:Service>

[ ... ]
</rdf:RDF>
```

## Referenz

```
<rdf:RDF [...]>
[ ... ]
<ore:Aggregation [...]>
  <edm:isShownBy rdf:resource="https://beispiel-institution.at/iiif-service/bild-1/full/max/0/default.jpg"/>
  <edm:hasView rdf:resource="https://beispiel-institution.at/iiif-service/bild-2/full/max/0/default.jpg"/>
  <edm:hasView rdf:resource="https://beispiel-institution.at/iiif-service/bild-3/full/max/0/default.jpg"/>
</ore:Aggregation>
[ ... ]
</rdf:RDF>
```

Bei dem angeführten Beispiel sei die Struktur der URLs wie folgt näher beschrieben:

- Eine URL in der Form von z. B. `https://beispiel-institution.at/iiif/bild-1/full/max/0/default.jpg` ist ein starker Indikator dafür, dass ein URI Syntax auf dem IIIF Protokoll ((`https://iiif.io/api/image/3.0/#2-uri-syntax`) bedient wird.
- Die URLs verweisen auf ein Bild. - Ein Webbrowser wäre fähig das Bild hinter der URL anzuzeigen.
- Die Syntax `full/max/0/default.jpg` garantiert im Sinne des IIIF Image Protokolls die Auslieferung eines passend formatierten Bildes:
  - `full`: Das Bild wird ohne jegliche Zuschnitte ausgeliefert.
  - `max`: Das Bild wird in voller größe ausgeliefert.
  - `0`: Das Bild wird ohne Rotation ausgeliefert.
  - `default`: Das Bild wird (im Gegensatz zu z. B. `gray` oder `bitonal`) in Standardqualität ausgeliefert.

Weiter sei der Zusammenhang zwischen den Identifikatoren der `edm:WebResource` Ressourcen und der `svcs:Service` Ressource hervorgehoben:

- `"rdf:about="https://beispiel-institution.at/iiif-service/bild-1/full/max/0/default.jpg"` ist die URL eines *konkreten Bildes inklusive aller notwendigen IIIF Parameter*.
- `"rdf:about="https://beispiel-institution/iiif-service/bild-1/"` ist die URL des Endpoints der Parameter entgegen nimmt um das konkrete Bild auszuliefern.

## Resultat

### Wie oben ...

- Der Kulturpool generiert auf Basis der drei bereitgestellten Ressourcen ein basales IIIF-Manifest und beliefert damit den Viewer des Kulturpools.
- Auf der Detailseite des beispielhaften Objektes ist es möglich durch die drei bereitgestellten Bilder zu blättern.
- Der Viewer weist grundsätzliche Funktionalitäten wie z. B. Rotation und Zoom auf.

### Darüberhinaus ...

- Die Auflösung des Bildes ist durch die Deklaration der IIIF Image API adaptiv wird im Zuges des Zoomens angepasst.

# IIIF Ressourcen *mit* Manifest und mit Deklaration der IIIF Image API

## Ausführung

### Wie oben ...

- Eine oder mehrere Ressourcen der Klasse `edm:WebResource` zeigen die Existenz von bildlichen Inhalten an.

- Auf die deklarierten Ressourcen wird in anderen Feldern des EDM (z.B. `edm:isShownBy`) verwiesen.
- Die angeführten bildlichen Inhalte werden über eine IIIF Image API ausgeliefert.
- Die URL der Ressourcen führt jeweils zu einem über das IIIF-Protokoll (für einen Webclient) passend ausgelieferten Bild.
- Die zugrundeliegende IIIF API wird als Ressource mit `<svcs:Service>` deklariert.
  - Innerhalb der `<svcs:Service>` Ressource wird das Level der bereitgestellten Image API via `<doap:implements>` deklariert.
  - Innerhalb der Ressource wird immer `<dcterms:conformsTo`  
`rdf:resource="http://iiif.io/api/image"/>`, d.h. der IIIF Standard ausgewiesen.
- Webressourcen, die bildliche Inhalte über das IIIF Protokoll ausliefern referenzieren mit `<svcs:has_service>` die IIIF Image API.
- Der Identifier für `<svcs:Service>` (und dementsprechend auch in Verwendung bei `svcs:has_service`) ist die Basis URI der IIIF API.

## Darüberhinaus ...

- Die via `edm:WebResource` deklarierten IIIF Ressourcen verweisen über `<dcterms:isReferencedBy>` auf ein IIIF Manifest aus der Presentation API.

## Beispiel

## Deklaration

```
<rdf:RDF [...]>
[ [...]
  <edm:WebResource rdf:about="https://beispiel-institution.at/iiif-service/bild-1/full/max/0/default.jpg">
    <svcs:has_service rdf:resource="https://beispiel-institution/iiif-service"/>
    <dcterms:isReferencedBy rdf:resource="https://beispiel-institution/iiif-service/manifest.json"/>
    [...] ... ]
  </edm:WebResource>

  <svcs:Service rdf:about="https://beispiel-institution/iiif-service/bild-1/">
    <dcterms:conformsTo rdf:resource="http://iiif.io/api/image"/>
    <doap:implements rdf:resource="http://iiif.io/api/image/2/level1.json"/>
  </svcs:Service>
```



```
[ ... ]  
</rdf:RDF>
```

## Referenz

```
<rdf:RDF [...]>  
[ ... ]  
[<ore:Aggregation [...]>  
  [<edm:isShownBy rdf:resource="https://beispiel-institution.at/iiif-service/bild-1/full/max/0/default.jpg"/>  
  ]  
</ore:Aggregation>  
[ ... ]  
</rdf:RDF>
```

## Resultat

### Wie oben ...

- Der Viewer weist grundsätzliche Funktionalitäten wie z. B. Rotation und Zoom auf.
- Die Auflösung des Bildes ist durch die Deklaration der IIIF Image API adaptiv wird im Zuges des Zoomens angepasst.

### Darüberhinaus ...

- Der Kulturpool verwendet das zur Verfügung gestellte IIIF-Manifest und beliefert damit den Viewer des Kulturpools.
- Auf der Detailseite des beispielhaften Objektes ist es möglich durch die über das Manifest bereitgestellten Bilder zu blättern.
- Der IIIF Viewer des Kulturpools zeigt die Metadaten aus dem Manifest direkt im Viewer an.

---

Version #11

Erstellt: 12 November 2024 13:26:12 von SU

Zuletzt aktualisiert: 12 November 2024 14:03:49 von SU