Bedienungsanleitung der AG Texttechnologie

Nummer: TTLab-MAN-2016-08

AG Texttechnologie
Goethe-Universität Frankfurt am Main
Fachbereich für Informatik und Mathematik
Robert-Mayer-Straße 10
D-60325 Frankfurt am Main
Prof. Dr. Alexander Mehler

Auslesen von Ressourcen

am Beispiel von Bildatlanten

Giuseppe Abrami, Sebastian Voinea

Zusammenfassung
In der folgenden Anleitung werden diverse Methoden für den Zugriff auf das Ressourcen-Management, entwickelt von der AG Texttechnologie, erläutert. Das Ressourcen-Management ist für alle Anwendungen identisch. Erklärt wird das Auslesen des Ressourcen-Managements der Projects „PHI Picturing Atlas“. Alle Anweisungen erfolgen per RESTful-Aufrufen. Die API-Dokumentation findet sich unter http://phi.resources.hucompute.org.

Inhaltsverzeichnis

1. Grundlegendes

Die Webservices der AG Texttechnologie sind bezüglich ihrer Aufgaben getrennt und operieren unabhängig von einander. Bei Bedarf kommunizieren die Webservices allerdings untereinander um die jeweiligen Aufgaben zu erfüllen. Alle Anfragen sind als RESTful[1]-Anfragen zu stellen. Die API-Dokumentation zeigt die Menge der Aufrufe sowie deren Nutzen und exemplarische Ergebnisse. Alle Anfragen und Ergebnisse werden als JSON[2] übergeben. Alle Ergebnisse enthalten Statusinformation darüber, ob der Aufruf erfolgreich durchgeführt werden konnte. Ist dies der Fall, erhält man das Ergebnis der Anfrage; wenn nicht, erhält man eine entsprechende Fehlermeldung. Erfolgreiche Aufrufe beinhalten das entsprechende Ergebnis.

```
{
  "success": false,
  "message": "Something bad happend..."
}
```

```
{
  "success": true,
  "result": [...]
}
```

Der Einfachheit lassen wir bei den künftigen Beispielergebnissen den Rahmen weg und geben nur den Inhalt von „result“ aus. Grundsätzlich erfolgt das Ressourcen-Management benutzerorientiert. Es gibt zwar die Möglichkeit, Ressourcen auch öffentlich zugänglich zu machen, jedoch ist für eine vollständige Nutzung ein Benutzerzugang notwendig. Um mit dem Ressourcen-Management zu kommunizieren, ist eine vorherige Authentifizierung nötig. Nach erfolgreicher Authentifizierung erhält der Benutzer eine `SESSION` welche bei künftigen Aufrufen mit übergeben werden muss.

1.1 RESTful Wrapper

Viele Programmiersprachen verfügen über eine Wrapper mit dem auf einfache Weise Anfragen an einen RESTful-Dienst gesendet werden. Obwohl in vielen Programmiersprachen diese Möglichkeit besteht, sind nicht alle gleich

[1] https://en.wikipedia.org/wiki/Representational_state_transfer
[2] http://www.json.org/

komfortabel. Für unsere künftigen Beispielaufrufe verwenden wir die Möglichkeiten von *jQuery*[3].

```
$.ajax({
  type:       "GET",
  dataType:   "json",
  url:        "https://phi.resources.hucompute.org/
  ↪  isAlive",
  success:    function(data){
    // do something with the result...
  },
  failure:    function(result){
    // do something with the failure...
  }
});
```

2. Authentifizierung

Zur Kommunikation mit dem Ressourcen-Management ist es erforderlich, sich zuvor als berechtigter Nutzer zu authentifizieren. Nach erfolgreiche Authentifizierung erhält man eine `SESSION`-ID, welche für alle künftigen Aufrufe benötigt wird. Man authentifiziert sich gegen den *Authority-Manager*. Der *Authority-Manager* verfügt ebenso wie das Ressourcen-Management über eine API-Dokumentation.[4]

2.1 Vorgehen

```
$.ajax({
  type: "POST",
  dataType: "json",
  url: "https://authority.hucompute.org/login",
  param:
  {
    username: USERNAME,
    password: PASSWORD
  },
  // [...]
  });
```

Der Benutzername (**USERNAME**) sowie das Passwort (**PASSWORD** als MD5-Hashwert[5]) sind zu übergeben.

Als Ergebnis, nach erfolgreicher Anmeldung, erhält man dann die Benutzerangaben sowie eine für diesen Benutzer gültige `SESSION`-ID.

```
{
 "user": "https://somesomain.de/user/0",
 "userName": "bsinclair",
 "fullName": "Bob Sinclair",
 "session": "34802du8928901i901290sklsop"
}
```

Um die Validität der gepeicherten `SESSION`-ID zu überprüfen, ist der Aufruf **/checklogin** auszuführen.

[3] https://jquery.com/
[4] https://authority.hucompute.org
[5] https://en.wikipedia.org/wiki/MD5

```
$.ajax({
  type: "GET",
  dataType: "json",
  url: https://authority.hucompute.org/checklogin",
  param:
  {
    session: SESSION
  }
  // [...]
  });
```

Eine erfolgreiche Abfrage produziert das gleiche Ergebnis wie eine erfolgreiche Anmeldung (s. o.).

2.2 Benutzer-Präferenzen

Der **Authority-Manger** verwaltet alle Informationen bezüglich der im System registrierten Benutzer. Darüberhinaus bietet er die Möglichkeit, *Benutzer-Präferenzen* zu verwalten.

2.3 Anzeigen von Benutzer Präferenzen

```
$.ajax({
  type: "GET",
  dataType: "json",
  url: "https://authority.hucompute.org/
  ↪  preferences",
  param:
   {
    session: SESSION,
    [property: string]
   },
   // [...]
  });
```

Der Parameter *property* ist optional. Ist dieser nicht vorhanden, werden alle *Properties* angezeigt.

```
[
  {
    "CURRVIEW": "start-view"
  }
]
```

Properties für die Präferenzen können beliebig gesetzt werden:

```
$.ajax({
  type: "POST",
  dataType: "json",
  url: "https://authority.hucompute.org/
  ↪  preferences",
  param:
   {
    session: SESSION,
    property: string,
    value: object
   },
   // [...]
  });
```

Properties können auch gelöscht werden:

```
$.ajax({
  type: "DELETE",
  dataType: "json",
  url: "https://authority.hucompute.org/
  ↪  preferences",
  param:
   {
     session: SESSION,
     property: string
   },
   // [...]
  });
```

3. Ressourcen-Manager

Nachdem man erfolgreich eine SESSION-ID vom *Authority-Manager* erhalten hat, empfiehlt es sich, diese SESSION-ID zu speichern. Die SESSION-ID ist Grundlage für alle weiteren Aufrufe. Durch die SESSION-ID erhalten wir einen benutzerbezogenen Zugriff auf die Ressourcen, welche durch den *Resource-Manager* verwaltet werden. Der *Resource-Manager* kann alle Arten von Ressourcen verwalten. Diese Anleitung wird insbesondere auf das Anzeigen von Bildern eingehen. Generell sind Dokumente im *Resource-Manager* in **Repositories** organisiert. Repositories bilden eine vergleichbare Struktur wie Ordner in einem herkömmlichen Dateisystem.

3.1 Repositories

Um sich alle Dokumente eines Repositories ausgeben zu lassen, muss man zunächst das gewünschte Repository ermitteln. Dies geht über den einfachen Aufruf **/repositories**. Hierbei benötigt man einen Knotenparameter (**node**) der als Quellrepository verwendet wird. Der Wert **root** steht hierbei für das Wurzelverzeichnis:

```
$.ajax({
  type:      "GET",
  dataType:  "json",
  url:       "https://phi.resources.hucompute.org/
  ↪  repositories",
  param:
   {
     session: SESSION_ID,
     node: "root"
   },
   // [...]
});
```

Durch diese Abfrage erhält man eine Auflistung aller Repositories, die im Wurzel-Repository enthalten sind. Die Repositories werden je nach Zugriffsberechtigung angezeigt.

```
{
 "total": 2,
 "data": [
   {
     "parent":
      ↪  "http://phi.resources.hucompute.org/
      ↪  repository/1",
     "access": 4,
     "id": "2_466",
     "text": "home",
     "leaf": false,
     "uri": "http://phi.resources.hucompute.org/
      ↪  repository/2",
     "qtip": "http://phi.resources.hucompute.org/
      ↪  repository/2<br>home
      ↪  repository"
   },
   {
     "parent":
      ↪  "http://phi.resources.hucompute.org/
      ↪  repository/1",
     "access": 4,
     "id": "3_466",
     "text": "resources",
     "leaf": false,
     "uri": "http://phi.resources.hucompute.org/
      ↪  repository/3",
     "qtip": "http://phi.resources.hucompute.org/
      ↪  repository/3<br>Resources"
   }
 ],
 "success": true
}
```

Wie im vorhergehenden Listing zu sehen, wurden im Wurzel-Repository zwei Repositoried angelegt, nämlich *home* und *resources*. Nun kann man die **/repositories**-Anfrage nochmals aufrufen, jedoch mit einer neuen **node**, welche das neue Quellrepository wird. Als **node** kann sowohl die *id* sowie die *uri* eines Repositories angegeben werden.

```
$.ajax({
  type:      "GET",
  dataType:  "json",
  url:       "https://phi.resources.hucompute.org/
  ↪  repositories",
  param:
   {
     session: SESSION_ID,
     node: "http://phi.resources.hucompute.org/
      ↪  repository/3"
   },
   // [...]
});
```

Mit diesem Verfahren kann man sich die Baumstruktur aller Repositorien ausgeben lassen. Man kann am letzten Ergebnis (s. Listing 1 auf der nächsten Seite) gut erkennen, dass die Bilder des Projekts „PHI Picturing Atlas“ im Verzeichnis `http://phi.resources.hucompute.org/repository/2530` liegen. Da sich die URIs nicht mehr verändern kann man diese als Grundlage für die gezielte Ausgabe der Ordner sowie der sich darin befindenden Dokumente verwenden.

```
{
 "total": 5,
 "data": [
   {
     "parent": "http://phi.resources.hucompute.org/repository/3",
     "access": 4,
     "id": "2530_35",
     "text": "PHI Picturing History Atlas",
     "leaf": false,
     "uri": "http://phi.resources.hucompute.org/repository/2530",
     "qtip": "http://phi.resources.hucompute.org/repository/2530<br>Some Description"
   },
   {
     "parent": "http://phi.resources.hucompute.org/repository/3",
     "access": 4,
     "id": "5_35",
     "text": "Ontologies",
     "leaf": true,
     "uri": "http://phi.resources.hucompute.org/repository/5",
     "qtip": "http://phi.resources.hucompute.org/repository/5<br>Some Description"
   },
   {
     "parent": "http://phi.resources.hucompute.org/repository/3",
     "access": 4,
     "id": "7_36",
     "text": "UploadFiles",
     "leaf": false,
     "uri": "http://phi.resources.hucompute.org/repository/7",
     "qtip": "http://phi.resources.hucompute.org/repository/7<br>Some Description"
   },
   {
     "parent": "http://phi.resources.hucompute.org/repository/3",
     "access": 4,
     "id": "219_36",
     "text": "Test",
     "leaf": true,
     "uri": "http://phi.resources.hucompute.org/repository/219",
     "qtip": "http://phi.resources.hucompute.org/repository/219<br>Some Description"
   },
   {
     "parent": "http://phi.resources.hucompute.org/repository/3",
     "access": 4,
     "id": "2510_36",
     "text": "export",
     "leaf": true,
     "uri": "http://phi.resources.hucompute.org/repository/2510",
     "qtip": "http://phi.resources.hucompute.org/repository/2510<br>export"
   }
 ],
 "success": true
}
```


Listing 1. Ergebnis der Repositorien mit der **node** 3.

3.2 Dokumente

Dokumente sind die allgemeinste Form von Dateien, die in einem Repository abgelegt werden können. Bilder sind hierbei spezielle Dokumente, auf die wir in Abschnitt 3.4 näher eingehen werden.

Um sich Dokumente in einem Repository anzeigen zu lassen, benötigt man den Aufruf **/documents**. Wichtigster Parameter ist hierbei wiederum die **node**, welcher das Repository angibt, dessen Dokumente angezeigt werden sollen:

```
$.ajax({
  type:      "GET",
  dataType:  "json",
  url:       "https://phi.resources.hucompute.org/
  ↪  documents",
  param:
   {
     session: SESSION_ID,
     node: "REPO_URI"
   },
   //[...]
});
```

Als Ergebnis erhalten wir eine Liste von Dokumenten. (Sollten keine Dokumente im Repository vorhanden sein, ist die Liste leer.)

```
{
  "success": true,
  "totalCount": 0,
  "result": [
  {
    "uri": "http://phi.resources.hucompute.org/
    ↪  document/123",
    "description": "Newspaper Article",
    "created": "Created 23:55:2384 UTC",
    "modified": "Modified 23:57:1934 UTC",
    "name": "Foo-File",
    "id": 123,
    "accessPermission": 2,
    "storageType": "FILEDOCUMENT",
    "mimeType": "application/pdf",
    "size": 128,
    "thumbnail": "http://
    ↪  phi.resources.hucompute.org/thumbnail/123",
    "fileDownload": "http://
    ↪  phi.resources.hucompute.org/file/123",
    "isLeaf": true,
    "parents": "JArray with name and uri of
    ↪  parent/s",
  }
  ]
}
```

Die Eigenschaften eines Dokuments sind selbsterklärend.

3.3 Filter und Sortieren

Die Dokumentabfrage kann durch einige Parameter spezifiziert werden. Es ist möglich, das Ergebnis durch Filter- und Sortierregeln einzuschränken. Die Filter- und Sortiereinschränkungen beziehen sich hierbei auf die Datenfelder der jeweiligen Objekte, z. B.:

```
[
 {
  "operator":"like",
  "value":"gr",
  "property":"name"
 }
]
```

Die Eigenschaft **property** definiert, nach welcher Eigenschaft gefiltert werden soll. Der Wert von **operator** bezieht sich auf den Vergleichsoperator und der Wert von **value** legt fest, womit die Eigenschaft verglichen werden soll. Alle Einträge, für welche diese Bedingungen nicht zutreffen, werden im Ergebnis aussortiert.

Es kann nach den Eigenschaften `name`, `description`, `size`, `storage`, `downloadable`, `modified`, `created` sowie nach allen Annotationen sortiert und gefiltert werden. Da man alle Dokumente im Ressourcen-Management beliebig annotieren kann, ist es an dieser Stelle nicht möglich, eine Übersicht über die vorhandenen Annotations-Properties zu geben.

Das Sortieren funktioniert nach demselben Prinzip, indem die Dokumentabfrage mit dem **sort**-Parameter aufgerufen wird. Dieses Objekt enthält den Sortierparameter sowie die Sortierrichtung.

```
[
 {
  "direction":"DESC",
  "property":"size"
 }
]
```

```
$.ajax({
  type:      "GET",
  dataType:  "json",
  url:       "https://phi.resources.hucompute.org/
  ↪  documents",
  param:
   {
     session: SESSION_ID,
     node: "http://phi.resources.hucompute.org/
     ↪  repository/547",
     filter:[
      {
  "operator":"like",
  "value":"Jahre",
  "property":"name"
      }
     ],
     sort: [
      {
  "direction":"DESC",
  "property":"name"
      }
     ]
   },
```

```
    //[...]
});
```

3.4 Namensraum von Bildern

Bilder sind spezielle Ressourcen, für die es einen eigenen Namensraum gibt. Die URI einer Ressource lautet z.B. `https://phi.resources.hucompute.org/document/950`. Hierbei ist `document` der Namensraum für Dokumente. Sobald man den Namensraum in `image` ändert, wird die Ressource als Bild behandelt. Sollte die Ressource kein Bildformat haben, erhält man einen entsprechenden Hinweis. Mit dem Namensraum für `image` hat man vielfältige Möglichkeiten.

Da alle URIs in unseren Webservices auch mit Inhalt hinterlegt sind, erhält man, wenn man die URI eines Elements aufruft, auch die dazugehörigen Informationen. Voraussetzung ist natürlich, dass man die nötigen Zugriffsrechte dazu besitzt.


```
{"result":
  {"owner":
   {
    "label":"sysop",
    "uri":"https://authority.hucompute.org/user/0"
   },
   "image":"http://phi.resources.hucompute.org/
   ↪  image/950",
   "downloadable":true,"created":"2016-04-26
   ↪  23:23:52.279",
   "description":"",
   "permission":4,
   "uri":"http://phi.resources.hucompute.org/
   ↪  document/950",
   "filedownload":"http://
   ↪  phi.resources.hucompute.org/file/950",
   "storetype":"FILEDOCUMENT",
   "size":1989242,
   "children":[],
   "name":"1000 Jahre Deutscher Geschichte (Homann
   ↪  1934)  197.png",
   "modified":"2016-04-26 23:23:52.279",
   "mimetype":"image/png",
   "information":
     {
      "transparency":3,
      "width":1680,
      "type":7,
      "height":1080
     },
   "parents":[
     {
      "label":"1000 Jahre Deutscher Geschichte
      ↪  (Homann 1934)",
      "uri":"http://phi.resources.hucompute.org/
      ↪  repository/547"}]},
      "success":true
     }
```


Der Rückgabewert in `image` gibt den Namensraum für Bilder an und die resultierende URI `http://phi.resources.hucompute.org/image/950` liefert uns nach dessen Aufruf das Bild selbst zurück. Im Beispiel betragen die Abmessungen des Bildes eine Breite von 1680 und eine Höhe von 1080 Pixeln und werden im Rückgabewert von `information` angegeben. Der Namensraum bietet aber noch viel mehr. So kann man mit einer entsprechenden Erweiterung des vorherigen Aufrufs um zwei Parameter das Bild in einer gewünschten Größe zurückgeben lassen. Durch das Anfügen von **/WIDTH/HEIGHT** an die Image-URI erhalten wir ein Bild, welches proportional auf die jeweils größte Angabe von Breie oder Höhe angepasst wird. Der Namensraum **thumbnail** liefert eine Vorschau des Bildes in der Abmessung von 200×200 Pixeln.

Die URI eines Bildes lässt sich einfach in den `src`-Teil eines `<img>`-Elements einer HTML-Seite einfügen und wird dann im Browser geladen. Es gilt zu beachten, dass auch hier die jeweilige Session mitgegeben werden muss, wie im folgenden Beispiel illustriert:

```
<html>
<body>
  <img src='http://phi.resources.hucompute.org/
   ↪  image/950?session=SESSION'
   ↪  />
  <img src='http://phi.resources.hucompute.org/
   ↪  image/950/500/200?session=SESSION'
   ↪  />
  <img src='http://phi.resources.hucompute.org/
   ↪  thumbnail/950?session=SESSION'
   ↪  />
</body>
</html>
```

Rückfragen und Kontakt

Bei Rückfragen bezüglich der Verwendung der beschriebenen API wenden Sie sich bitte an Giuseppe Abrami.[6]

[6] abrami@em.uni-frankfurt.de