



Multiple Picture Storiies erstellen und auf GitHub Pages hosten

In Part 02 haben wir den Storiies Editor kennengelernt, mit dem sich sehr einfach effektvolle virtuelle Präsentationen erstellen lassen. Die basieren, wie wir inzwischen wissen, auf Annotationen von gezielt ausgewählten Canvasausschnitten. Allerdings geht das im Storiies Editor (bislang) nur mit *einem* Bild. Um Präsentationen mit mehreren Bildern zu erstellen, braucht es ein kleines Work Around, das im Folgenden vorgestellt wird. Dabei wird auch gezeigt, wie man IIIF Manifests und AnnotationLists über GitHub Pages veröffentlicht.

Inhalt

- Multiple Picture Storiies aus einzelnen Storiies zusammensetzen
- AnnotationLists und IIIF Manifests aus Storiies exportieren
- Mit einem GitHub Account und GitHub Pages IIIF Manifests und AnnotationLists hosten
- Multiple Picture Storiies anzeigen und einbetten

Vorüberlegungen

Nehmen wir an, wir wollen eine kleine virtuelle Präsentation bauen und online stellen – zum Beispiel, um auf einer Projektseite ein historisches Dokument oder Artefakt zu präsentieren. Der Storiies Editor bietet einen einfachen Weg, das zu bewerkstelligen. Allerdings müssen wir ein wenig tricksen, wenn wir mehr als ein Bild präsentieren wollen. Hier schon mal herzlichen Dank an Andy Cummins von Cogapp: Während der Storiies Editor zurzeit noch nicht darauf ausgelegt ist, mehrere Bilder zu verarbeiten, kann der Storiies Viewer Multiple Picture Storiies sehr wohl anzeigen. Wir basteln sie daher außerhalb des Editors aus drei Einzelpräsentationen zusammen und lassen sie danach vom Storiies Viewer ausgeben.

Sie können für die folgende Übung das mit dem Bodleian Manifest Editor erstellte IIIF Manifest aus **Part 04** verwenden (die Version OHNE Annotationen). Da Storiies aber auch gewöhnliche Bilddateien zu IIIF Images umwandelt, können Sie auch drei beliebige jpeg oder png Dateien verwenden – so wie ich.

Laden Sie die drei Bilder (bzw. die info.json Dateien aus dem IIIF Manifest) nacheinander im Storiies Editor hoch und annotieren Sie (siehe **Part 02**). Wenn es nur um's Ausprobieren geht, reichen ein oder zwei Annotationen pro Bild. Eine richtige Präsentation sollte im Vorfeld jedoch gut geplant sein: Welche Bilder sollen in welchen Schritten annotiert und hintereinander gezeigt werden? Später lassen sich zwar der Text der Annotationen und auch die Description noch ändern, der Ausschnitt des jeweiligen Canvas, in den pro Annotation hineingezoomt wird, aber nicht mehr so leicht. Anschauungsbeispiele gibt es hier:

<http://storiies.cogapp.com/viewer?manifest=https://storiies.cogapp.com/ycba/manifest.json&quality=native>

<https://storiies.cogapp.com/viewer?manifest=https://storiies.cogapp.com/gdl/manifest.json>

Aus 3 mach 1: Manifests und AnnotationLists herunterladen

1. Sie haben nun drei Einzelpräsentationen mit dem Storiies Editor erstellt. Um daraus eine Multiple Picture Storiie zu bauen, müssen wir zunächst die **IIIF Manifests der drei Einzelpräsentationen herunterladen**. Wie das geht, zeigt **Andy Cummins** im **Video aus Part 02** (siehe auch unten Ressourcen und Links, ca. Min. 15:20). Hier eine Anleitung:
2. Öffnen Sie die erste Ihrer Präsentationen mit Chrome im Storiies Viewer: Wenn Sie den Editor noch geöffnet haben, klicken Sie auf „Preview“. Storiies versendet auch eine Email mit einem Bearbeitungslink sowie dem Link zur Präsentation.
3. Rechtsklick und „Untersuchen“ auswählen.
4. Klicken Sie in dem sich öffnenden Fenster oben auf „Network“ und dann auf „XHR“. Aktualisieren Sie die Seite: Daraufhin sollte eine Liste angezeigt werden, darunter info.json des IIIF Image und ein IIIF Manifest.

Name	Status	Type	Initiator	Size	Time	Wat
manifestJSON?manifest=https://jqz7t23pp9.execute-a......	304	fetch	shortcode.js:502	169 B	563 ms	
collect?v=1&v=j88&a=1266594782&t=pageview&s=1...	200	xhr	analytics.js:38	21 B	22 ms	
info.json	200	xhr	openseadragon.js:2174	(disk cache)	1 ms	

Abb.: IIIF Manifest von Storiies öffnen (eigener Screenshot, bearbeitet).

- Doppelklick auf das IIIF Manifest: Die JSON Datei öffnet sich im Browser. Rechtsklick auf „@id“, „Link speichern unter“ auswählen und das Manifest speichern (z. B. in einem neuen Ordner auf dem Desktop).

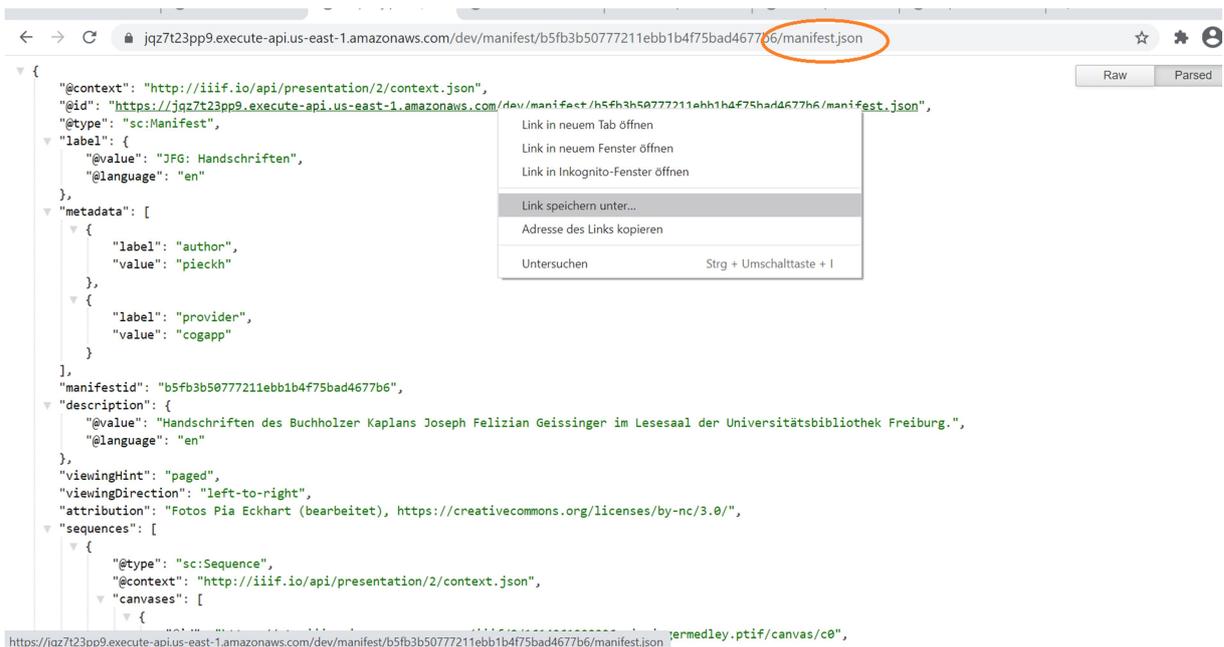


Abb.: IIIF Manifest aus dem Browser speichern (eigener Screenshot, bearbeitet).

- Kehren Sie zum IIIF Manifest im Browser zurück. Wenn Sie nach unten scrollen, findet sich nach „canvases“, den Informationen zum Bild und „other content“ die „@id“ der AnnotationList (zur **Verlinkung von AnnotationLists im IIIF Manifest** siehe **Part 05**). Klicken Sie darauf, die AnnotationList öffnet sich im Browser. Speichern Sie auch die AnnotationList (z.B. als annotations01) in Ihrem Ordner.
- Jetzt müssen wir noch alle Schritte für alle IIIF Manifests und AnnotationLists wiederholen.

>> **Hinweis:** Es kann sein, dass Sie beim Download der AnnotationLists eine Fehlermeldung erhalten (bei mir ist das jedenfalls so). Nicht sehr elegant, aber wirksam: Öffnen Sie im **Atom Editor** eine neue Datei und kopieren Sie die JSON Datei der AnnotationList aus dem Browser komplett hinein. Danach speichern Sie die neue Datei in Ihrem Ordner (**Endung .json** nicht vergessen, also z. B. annotations01.json). Sollten Sie die AnnotationList im Browser bereits geschlossen haben, finden Sie deren URL im IIIF Manifest unter „other content“, „@id“, und können Sie so im Browser öffnen.

Aus 3 mach 1: Lustiges Copy&Paste im Atom Editor

Das IIIF Manifest, das zum ersten Image der künftigen Präsentation gehört, nehmen wir als Startpunkt für die folgende Bastelei. Wir brauchen dazu nur die IIIF Manifests (nicht die AnnotationLists):

1. IIIF Manifest 01 und 02 im Atom Editor öffnen. Lokalisieren und kopieren Sie den kompletten Code für das Canvas + verlinkter AnnotationList in 02, und fügen Sie das nach dem Canvas + verlinkter AnnotationList in 01 ein.

```
File Edit View Selection Find Packages Help
JFG02_manifest.json
21 "sequences": [
22   {
23     "@type": "sc:Sequence",
24     "@context": "http://iiif.io/api/presentation/2/context.json",
25     "canvases": [
26       {
27         "@id": "https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874841.ptif/canvas/c0",
28         "label": {
29           "@value": "JFG: Muensterbuch\\n by pieckh\\n\\nFreiburg UB, Hs. 498, fol. 18v-19r\\n\\nPublic Domain",
30           "@language": "en"
31         },
32         "@context": "http://iiif.io/api/presentation/2/context.json",
33         "@type": "sc:Canvas",
34         "width": 3145,
35         "height": 2572,
36         "images": [
37           {
38             "resource": {
39               "@id": "https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874841.ptif/full/full/0/default.jpg",
40               "@type": "dctypes:Image",
41               "format": "image/jpeg",
42               "width": 3145,
43               "height": 2572,
44               "service": {
45                 "@context": "http://iiif.io/api/image/2/context.json",
46                 "profile": "http://iiif.io/api/image/1/level2.json",
47                 "@id": "https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874841.ptif"
48               }
49             },
50             "@type": "sc:Annotation",
51             "motivation": "sc:painting",
52             "on": "https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874841.ptif/canvas/c0"
53           }
54         ],
55         "otherContent": [
56           {
57             "@id": "https://ja27t2pp9.execute-api.us-east-1.amazonaws.com/dev/annotation/1fa5278076b811ebad3a59439fa21da8/annotations.json",
58             "@type": "sc:AnnotationList"
59           }
60         ]
61       }
62     ]
63   }
64 ]
```

Abb.: Aus IIIF Manifest 02 wird Canvas + verlinkte AnnotationList kopiert: Die öffnende geschweifte Klammer nach „**canvases**“: [und die von Atom hervorgehobene schließende geschweifte Klammer umfassen den gesamten zum Canvas gehörigen Code (eigener Screenshot, bearbeitet).

Dabei gibt es einen einfachen Trick: Wir müssen nicht wirklich etwas von JSON Syntax verstehen, aber darauf achten, dass **alle sich öffnende Klammern [{ auch wieder in der richtigen Reihenfolge schließen }]**. Klickt man eine Klammer an, hebt der Atom Editor das Gegenstück optisch hervor.

2. Genauso lässt sich auch die richtige Stelle lokalisieren, an der der **kopierte Codeschnipsel in das IIIF Manifest 01 eingefügt** werden soll. Wichtig: Nach der letzten schließenden Klammer des bestehenden Canvas **muss ein Komma** eingefügt werden, bevor das neue Canvas eingefügt wird.

```

File Edit View Selection Find Packages Help
Welcome Guide JFG01_manifest.json JFG02_manifest.json
44 "service":{
45   "@context":"http://iiif.io/api/image/2/context.json",
46   "profile":"http://iiif.io/api/image/1/level2.json",
47   "@id":"https://storiiies-images.cogapp.com/iiif/2/1614261900096geissingermedley.ptif"
48 }
49 },
50 "@type":"oa:Annotation",
51 "motivation":"sc:painting",
52 "on":"https://storiiies-images.cogapp.com/iiif/2/1614261900096geissingermedley.ptif/canvas/c0"
53 }
54 },
55 "otherContent":[
56 {
57   "@id":"https://jgz7k23pp9.execute-api.us-east-1.amazonaws.com/dev/annotation/b5fb3b50777211eb1b4f75bad4677b6/annotations.json",
58   "@type":"sc:AnnotationList"
59 }
60 ],
61 },
62 {
63   "@id":"https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874041.ptif/canvas/c0",
64   "label":{
65     "@value":"JFG: Muensterbuch\n by pieckh\n nFreiburg UB, Hs. 498, fol. 18v-19r\n n\n nPublic Domain",
66     "@language":"en"
67   },
68   "@context":"http://iiif.io/api/presentation/2/context.json",
69   "@type":"sc:Canvas",
70   "width":3145,
71   "height":2572,
72   "images":{
73     {
74       "resource":{
75         "@id":"https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874041.ptif/full/full/0/default.jpg",
76         "@type":"dctypes:Image",
77         "format":"image/jpeg",
78         "width":3145,
79         "height":2572,
80         "service":{
81           "@context":"http://iiif.io/api/image/2/context.json",
82           "profile":"http://iiif.io/api/image/1/level2.json",
83           "@id":"https://storiiies-images.cogapp.com/iiif/2/1614181634927geissinger17874041.ptif"
84         }
85       },
86       "@type":"oa:Annotation",
87

```

Abb.: Zwischen Canvas 01 und Canvas 02 muss ein Komma stehen (eigener Screenshot, bearbeitet).

3. Zur Sicherheit am besten die gesamte Datei mit **JSONLint** prüfen. Ist die Datei **valide**, direkt das dritte Canvas + verlinkter AnnotationList aus IIIF Manifest 03 einfügen! Danach das neue IIIF Manifest mit drei Canvases unter einem neuen Namen speichern.

Wir haben jetzt ein feines neues IIIF Manifest, das alle Bilder und Annotationen unserer Multiple Picture Storiie enthält. Um die **Präsentation vom Storiies Viewer anzeigen** zu lassen, müssen wir das IIIF Manifest und die zugehörigen AnnotationLists **online hosten**, damit Storiies Viewer darauf zugreifen kann (die IIIF Images wird sich Storiies weiter vom eigenen IIIF Image Server holen, aber das IIIF Manifest haben wir ja verändert). Jetzt kommen **GitHub** und **GitHub Pages** ins Spiel.

Mit GitHub und GitHub Pages IIIF Manifests hosten

GitHub ist ein netzbasierter Dienst zur Versionsverwaltung für Software-Entwicklungsprojekte. GitHub Pages ist ein Service für statische Webseiten, der vom eigenen GitHub Account sehr leicht aktiviert werden kann. Wir benutzen GitHub, um unsere IIIF Manifests und AnnotationLists online zu hosten, damit der Storiies Viewer darauf zugreifen kann (vom lokalen Web Server funktioniert das nicht).

1. Falls nötig, erstellen Sie einen GitHub Account. Anleitungen gibt es hier: <https://pages.github.com> und ausführlich mit Screenshots von Glen Robson; er benutzt GitHub in dieser Übung um IIIF Images zu hosten, die Grundidee ist aber dieselbe: <https://training.iiif.io/iiif-1-day-workshop/image-api/level0-github-hosting.html>

- >> **Hinweis:** Der gewählte **Username** wird im Folgenden Bestandteil aller URLs werden!
- Legen Sie ein neues „repository“ an mit dem Namen „username.github.io“ (Username = GitHub Account Username). In meinem Fall: testiiif.github.io

github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

Great repository names are short and memorable. Need inspiration? How about [studious-train?](#)

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

- Jetzt aktivieren wir **GitHub Pages**: Klicken Sie auf „create new file“. Es öffnet sich eine Eingabezeile: Nach „username.github.io“ ergänzen Sie „index.html“.

Search or jump to... Pull requests Issues Marketplace Explore

testiiif / testiiif.github.io Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

or

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# testiiif.github.io" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/testiiif/testiiif.github.io.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/testiiif/testiiif.github.io.git
git branch -M main
git push -u origin main
```

<https://github.com/testiiif/testiiif.github.io/new/main>

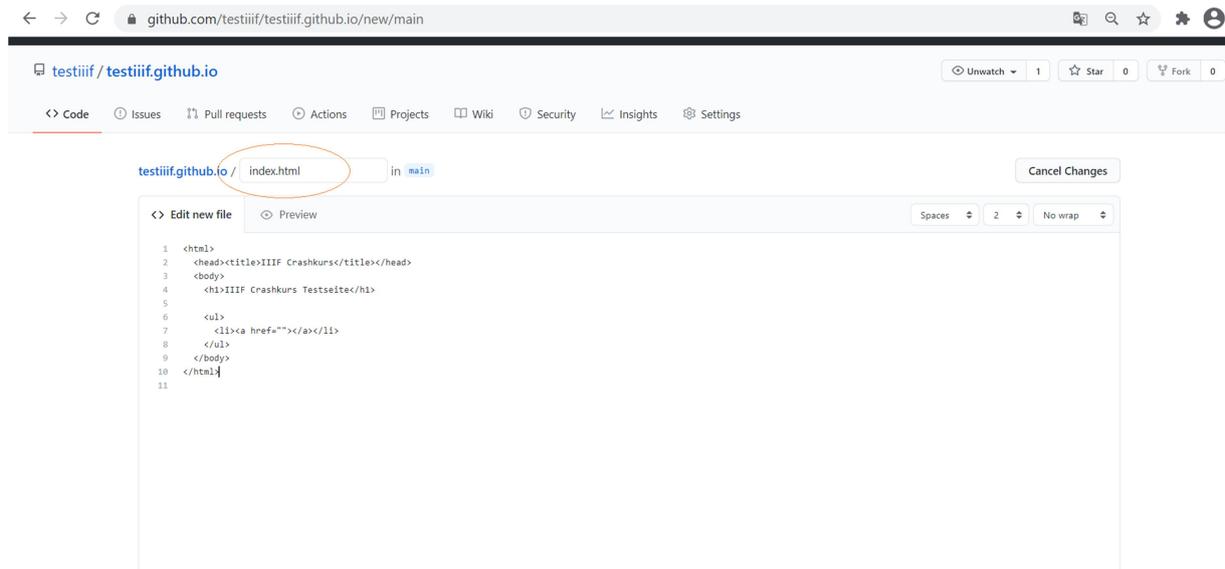
- Kopieren Sie folgenden html Code in den Editor. Wer mehr von html versteht als ich, kann das natürlich auch schicker machen:

```

<html>
  <head><title>IIIF Crashkurs</title></head>
  <body>
    <h1>IIIF Crashkurs Testseite</h1>

    <ul>
      <li><a href=""></a></li>
    </ul>
  </body>
</html>

```



5. Scrollen Sie ans Seitenende und klicken Sie auf „Commit new file“.
6. Im Browser sollten Sie jetzt unter <https://username.github.io> eine Seite erreichen, die „IIIF Crashkurs Testseite“ anzeigt (das dauert manchmal ein paar Minuten). Bei mir sieht das so aus: <https://testiif.github.io>
7. Jetzt laden wir das IIIF Manifest unserer Multiple Picture Storiie hoch: Gehen Sie zurück zu Ihrem Repository <https://github.com/username/username.github.io>
8. Klicken Sie auf „Add file“, „Upload file“. Sie können das IIIF Manifest jetzt mit drag&drop ins Repository laden.

Jetzt wird's ernst: Öffnen Sie Ihre Präsentation im Storiies Editor, indem Sie folgende Seite aufrufen:

`https://storiies.cogapp.com/viewer?manifest=https://username.github.io/{dateiname Manifest}.json`

Im gelben Abschnitt rufen wir ein Manifest im Storiies Viewer auf, der blaue Abschnitt verweist den Viewer auf die URL unseres auf GitHub Pages gehosteten IIIF Manifest an. In meinem Fall sieht das so aus:

https://storiies.cogapp.com/viewer?manifest=https://testiif.github.io/JFG_manifest.json

That's it...but.

AnnotationLists über GitHubPages hosten

Unsere Multiple Picture Storiies sind online, wir können Sie per Link teilen oder in eine Website einbetten (z. B. gleich auf unserer neuen GitHub Page, siehe unten).

Wer elegant Annotationen nachträglich ändern, eventuell übersehene (Tipp)Fehler korrigieren möchte usw., kann auch die zugehörigen AnnotationLists in das GitHub Repository laden. Logischerweise müssen die AnnotationLists dann neue URLs im IIIF Manifest bekommen, damit der Storiies Viewer auf sie zugreifen kann (ausführlicher erklärt in **Part 06**):

1. Im GitHub Repository „Add file“, „Upload file“ auswählen und mit drag&drop die AnnotationLists hochladen.
2. Die URLs der hochgeladenen AnnotationLists sind folgendermaßen aufgebaut: `https://username.github.io/{dateiname AnnotationList}.json`. In meinem Fall: https://testiiif.github.io/annotations_JFG01.json
3. Die neuen URLs der AnnotationLists im IIIF Manifest verlinken: Praktischerweise lassen sich die Änderungen direkt im GitHub Editor vornehmen (ggf. mit JSONLint validieren). Also, IIIF Manifest im Repository anklicken, auf das kleine Stift-Symbol rechts klicken („edit this file“) und die alten URLs unter „other content“, „@id“ mit den neuen GitHub Pages URLs austauschen.
4. Jetzt lassen sich auch **die AnnotationLists im GitHub Editor bearbeiten**, z. B. wenn man Änderungen am Text einer Annotation vornehmen möchte.

War da noch was? Multiple Picture Storiies in GitHub Pages einbetten

Falls Sie die Präsentation auf Ihrer GitHub Page zeigen wollen, steuern Sie in Ihrem GitHub Repository **index.html** an und klicken Sie auf das Stift-Symbol. Fügen Sie folgenden Code vor `</body>` ein:

```
<html>
  <head><title>IIIF Crashkurs</title></head>
  <body>
    <h1>IIIF Crashkurs Testseite</h1>

    <ul>
      <li><a href=""></a></li>
    </ul>

    <h2>Titel <!--Hier den Titel der eigenen Präsentation angeben-->. Eine Multiple Picture Storiie
    powered by
    <a href="https://www.cogapp.com" target="_blank">Cogapp</a> und
    <a href="https://storiies-editor.cogapp.com/" target="_blank">Storiies Editor</a></h2>
    <br/>
    <br/>

    <iframe width="100%" height="600" src=https://storiies.cogapp.com/viewer?manifest=<!--hier URL IIIF
    Manifest angeben-->?embed=true title="<!--Hier den Titel der eigenen Präsentation angeben--
    >"></iframe>

    </body>
  </html>
```

Auf „Commit changes“ klicken und etwas Geduld aufbringen – Es dauert mitunter eine Weile, bis die Änderungen auf <https://username.github.io> angezeigt werden.

Danke Andy Cummins, Cogapp, Storiies, Glen Robson und GitHub (Pages)!

Ressourcen und Links

Cogapp <https://cogapp.com>

Storiies Editor <https://storiies-editor.cogapp.com/>

GitHub <https://github.com>

Offizielle Anleitung zu GitHub Pages <https://pages.github.com>

Glen Robsons ausführliche Anleitung zur Erstellung eines GitHub Accounts und der Nutzung von GitHub Pages <https://training.iif.io/iif-1-day-workshop/image-api/level0-github-hosting.html>

Andy Cummins präsentiert den Storiies Editor auf Youtube <https://youtu.be/u4GC9ULypls>