

IIIF Image Conversion Training

- [1 What is IIIF Image Conversion?](#)
 - [1.1 Who is the training for image conversion aimed at?](#)
 - [1.2 What will you learn in this training?](#)
 - [1.3 How will you learn this?](#)
 - [1.4 What will you need?](#)
- [2 Image Conversion](#)
- [3 Memory consumption](#)

1 What is IIIF Image Conversion? [↗](#)

IIIF image conversion is the process of changing your original images to a multi-resolution format (e.g. TIFFs) to make them IIIF compatible and at the same time store them on an IIIF image server. The goal is to implement the IIIF APIs as described on the [IIIF website](#) and thus enhance the possibilities of working with your images in an interoperable way.

1.1 Who is the training for image conversion aimed at? [↗](#)

To take advantage of the IIIF framework, your images need to be available in multi-resolution format on a web server and a IIIF-compliant image server.

Generally, content providers have web servers where their images can be accessed on the Internet. To implement the IIIF standard for these images, it is necessary to install an IIIF compatible image server and to convert the images to multi-resolution format.

Europeana aggregators may support their content providers in adapting to the IIIF framework. They can do this by guiding their content partners through the process of image preparation with the help of this guide. Most importantly, they can make the content providers aware of the two scripts available [here](#).

1.2 What will you learn in this training? [↗](#)

In this training we will go through the process of image conversion to IIIF format as described in the [IIIF Image Conversion Guide](#). For providing the images on the Internet we use the [Apache web server](#) and the IIIF-compliant [IIPImage server](#).

1.3 How will you learn this? [↗](#)

The workflow of image conversion is described step by step and the successful execution of each step can be verified using the included screenshots in this training.

1.4 What will you need? [↗](#)

For this training you will need a set of images (e.g. jpeg images) that you want to convert to multi-resolution format.

You could use your own images or download a test set of 10 images from here: [ConversionGuide_Images_TestSet.zip](#).

We will use [Apache web server](#) and [IIPImage server](#) in this training.

To follow all steps with your own images or the test set images, you must have successfully completed the installation of the Apache web server and the IIPImage Server. You can easily perform these two steps by running the [iif-install.sh](#) script with administrator permissions. (see also [IIIF Image Conversion Guide](#))

At first you have to make the script executable:

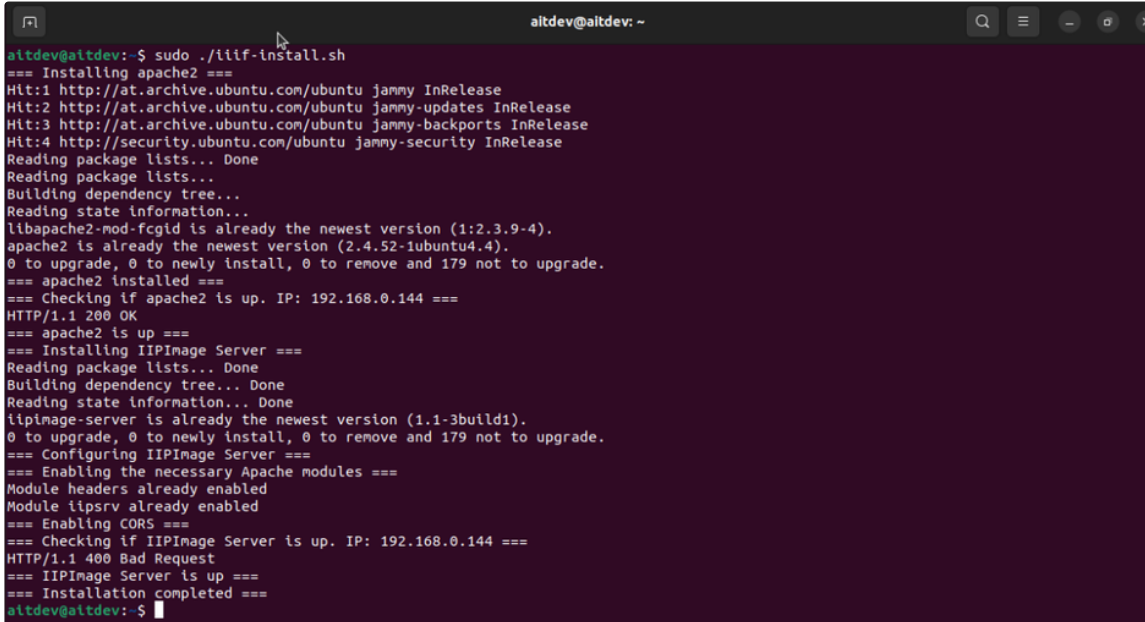
```
sudo chmod +x iif-install.sh
```

```
sudo chmod +x iiii-install.sh
```

```
@ubuntu:~$ sudo chmod +x iiii-install.sh  
[sudo] password for [REDACTED]  
@ubuntu:~$
```

Then you have to run the script in your terminal by simply typing:

```
1 sudo ./iiif-install.sh
```

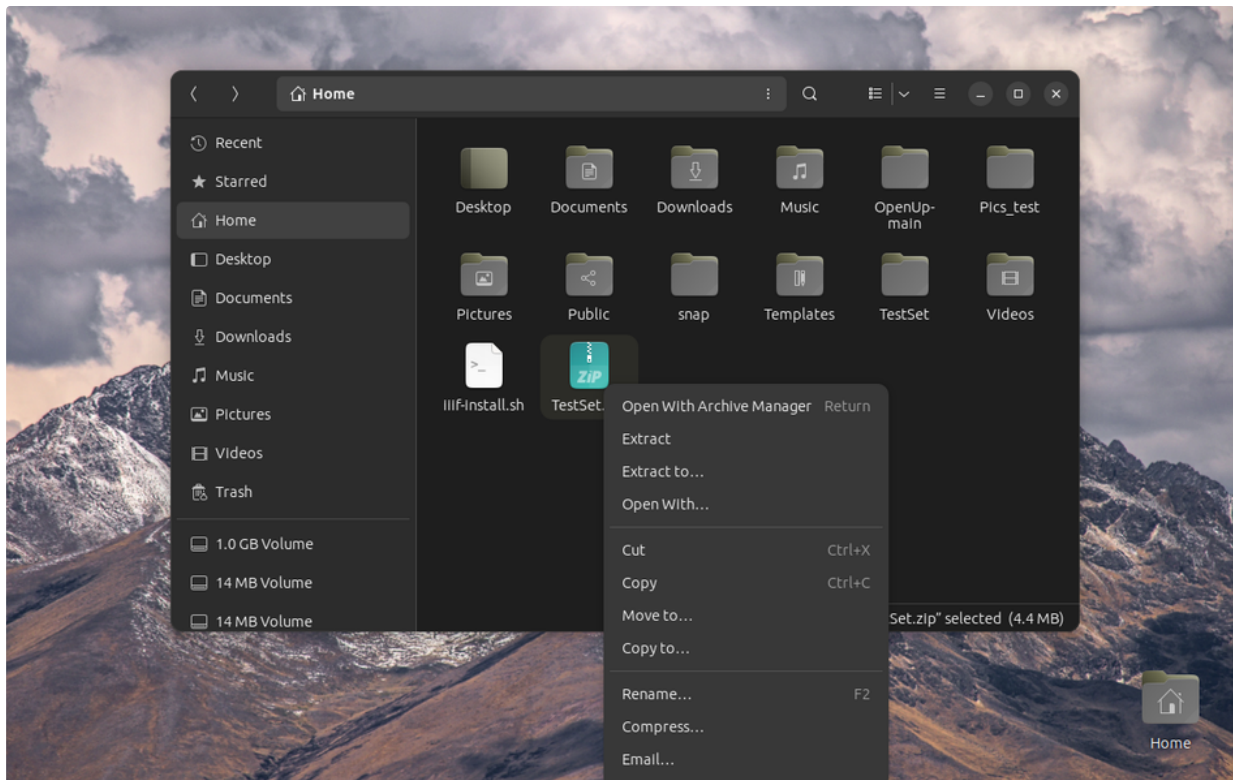


```
aitdev@aitdev: ~  
aitdev@aitdev:~$ sudo ./iiif-install.sh  
=== Installing apache2 ===  
Hit:1 http://at.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://at.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://at.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done  
Reading package lists...  
Building dependency tree...  
Reading state information...  
libapache2-mod-fcgid is already the newest version (1:2.3.9-4).  
apache2 is already the newest version (2.4.52-1ubuntu4.4).  
0 to upgrade, 0 to newly install, 0 to remove and 179 not to upgrade.  
=== apache2 installed ===  
=== Checking if apache2 is up. IP: 192.168.0.144 ===  
HTTP/1.1 200 OK  
=== apache2 is up ===  
=== Installing IIPImage Server ===  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
iipimage-server is already the newest version (1.1-3build1).  
0 to upgrade, 0 to newly install, 0 to remove and 179 not to upgrade.  
=== Configuring IIPImage Server ===  
=== Enabling the necessary Apache modules ===  
Module headers already enabled  
Module iipsrv already enabled  
=== Enabling CORS ===  
=== Checking if IIPImage Server is up. IP: 192.168.0.144 ===  
HTTP/1.1 400 Bad Request  
=== IIPImage Server is up ===  
=== Installation completed ===  
aitdev@aitdev:~$
```

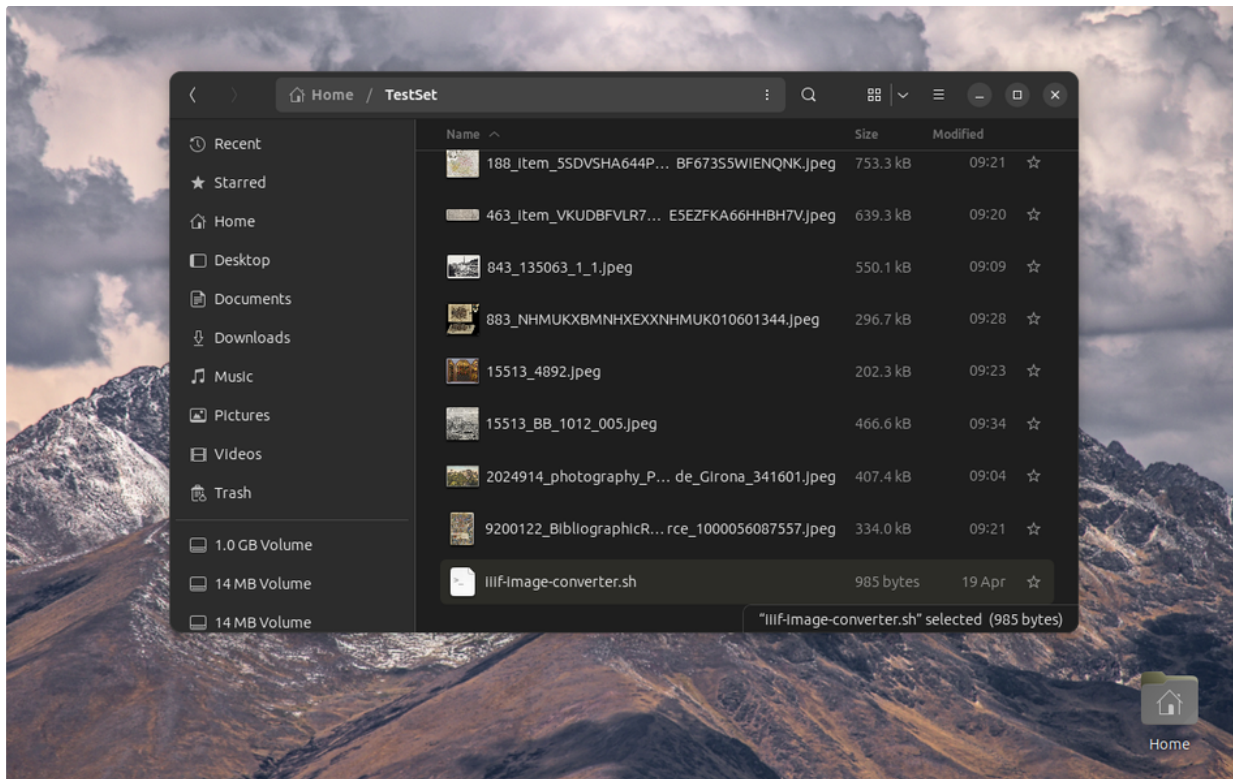
Figure 1 - Output of the command "sudo ./iiif-install.sh"

2 Image Conversion [↗](#)

To facilitate transport, the images are saved in a zip file so that they are not transferred individually, but all together. Your next step should be decompressing the zip file in your Home directory.



After that, you have to move the [iiif-image-converter.sh](#) file into the folder with the decompressed images.



Make the script executable with:

```
1 sudo chmod +x iiif-image-converter.sh
```

```
@ubuntu:~/TestSet$ sudo chmod +x iiif-image-converter.sh
[sudo] password for :
@ubuntu:~/TestSet$
```

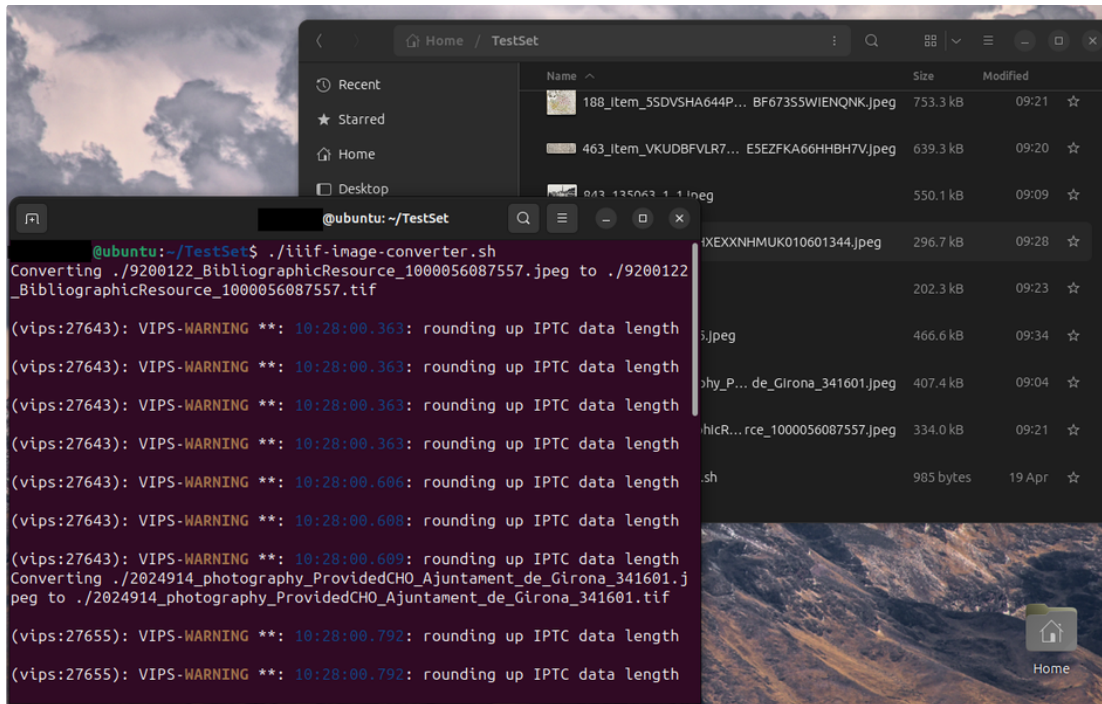
In your terminal, open the TestSet folder where your images and the script are located (`cd TestSet` if you are not already in that folder).

```
@ubuntu:~$ cd TestSet
@ubuntu:~/TestSet$
```

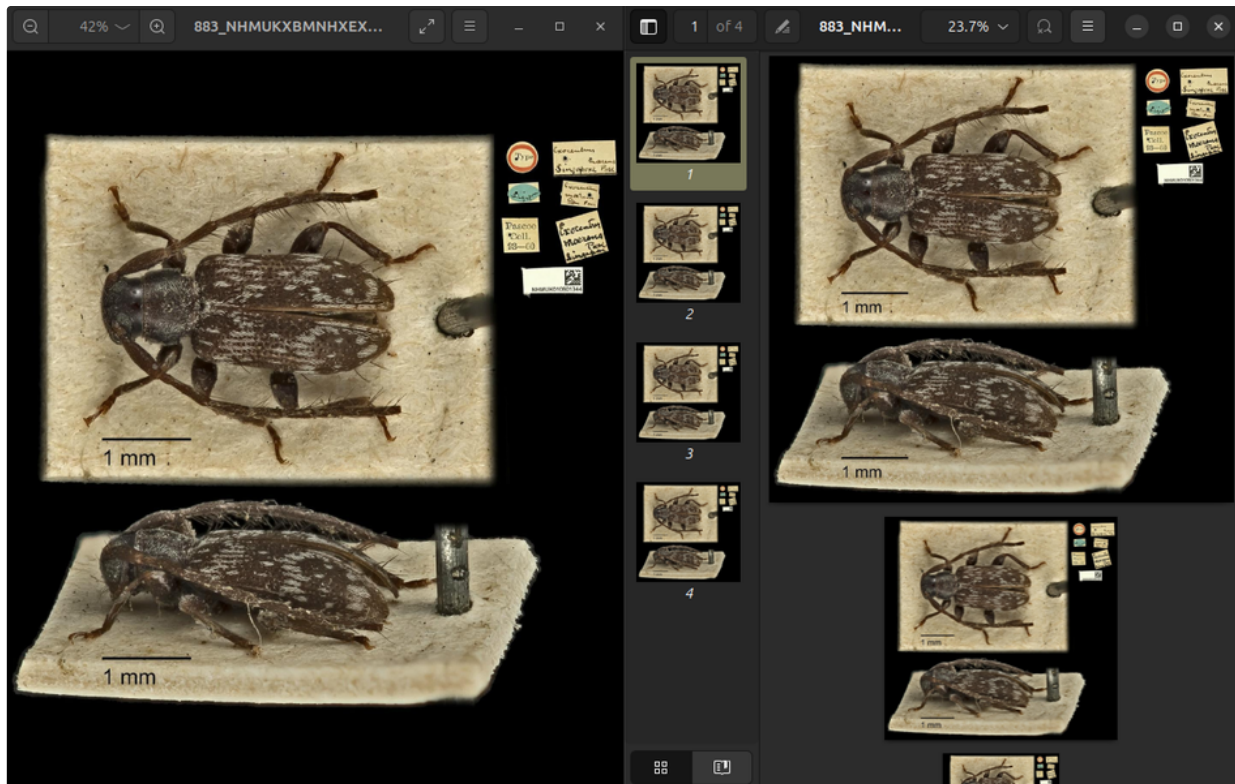
And run the script from there by simply typing:

```
1 ./iiif-image-converter.sh.
```

In the following screenshot you can see that during the execution of this script there is a corresponding TIFF image generated for every original image.



A better view example of the original versus corresponding tiff image is shown in the next figure. There you can see that the TIFF image consists of 3 layers that represent different resolutions of the original image.



At the end of execution the script demands sudo password in order to move the generated TIFF images to the image server folder.

```
Converting ./188_item_5GD7L7NNL2S05C27YVJOPTNFVRUWWDJK.jpeg to ./188_item_5GD7L7NNL2S05C27YVJOPTNFVRUWWDJK.tif
Converting ./117__3C8E3DE3_1FBD_4BA3_96D3_822F790CFAD0.jpeg to ./117__3C8E3DE3_1FBD_4BA3_96D3_822F790CFAD0.tif
[sudo] password for  :
@ubuntu:~/TestSet$
```

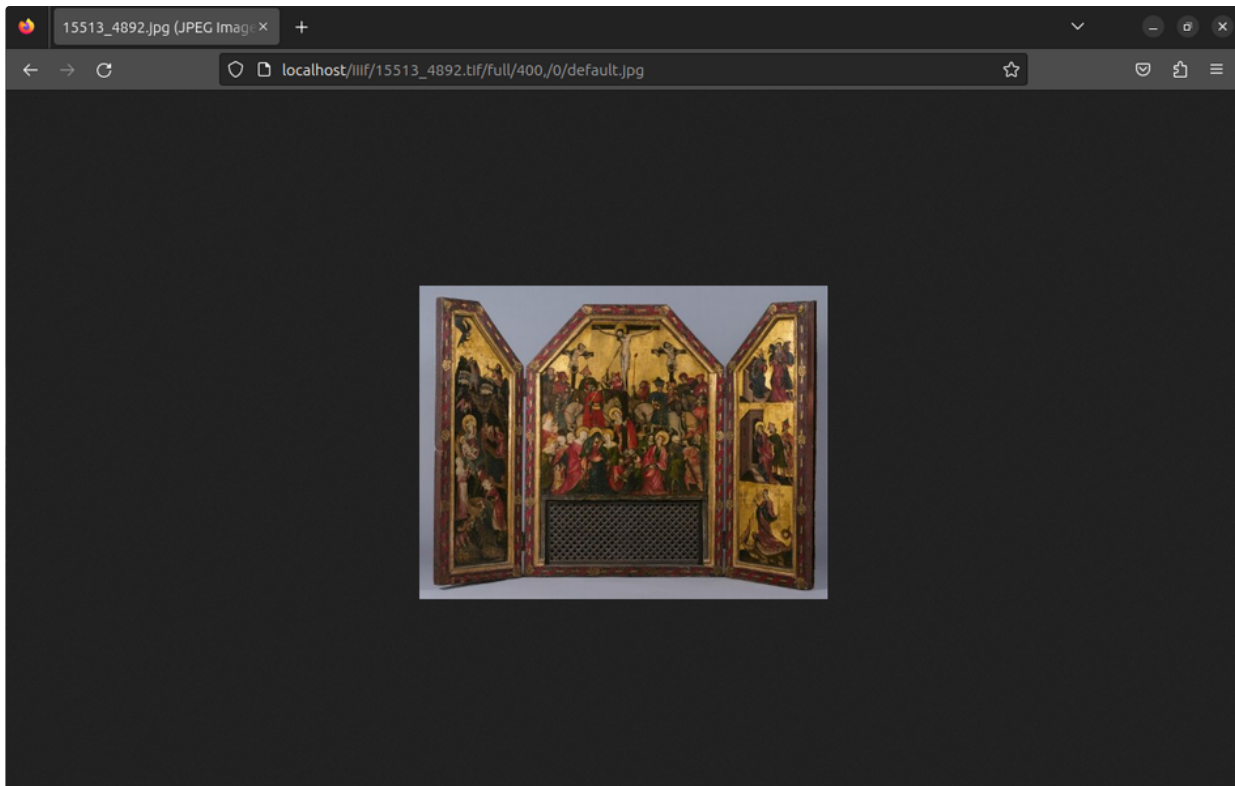
After correctly entering the sudo password the script ends with no outputs in the terminal and the images are moved to the `/var/www/iipimage-server` folder.

To check if the images are really there, go to this folder and type `ls` into your terminal. The following output should be obtained.

```
@ubuntu:~$ cd /var/www/iipimage-server/
@ubuntu:/var/www/iipimage-server$ ls
117__3C8E3DE3_1FBD_4BA3_96D3_822F790CFAD0.tif
15513_4892.tif
15513_BB_1012_005.tif
188_item_5GD7L7NNL2S05C27YVJOPTNFVRUWWDJK.tif
188_item_5SDVSHA644P7YM4ASLBF673S5WIENQNK.tif
2024914_photography_ProvidedCH0_Ajuntament_de_Girona_341601.tif
463_item_VKUDBFVLR7X4PM67UE5EZFKA66HHB7V.tif
843_135063_1_1.tif
883_NHMUKXB...XNHMUK010601344.tif
9200122_BibliographicResource_1000056087557.tif
```

Now, you can access the images from the browser with a URL.

For example, you can type into the browsers address bar: http://localhost/iiif/15513_4892.tif/full/400,/0/default.jpg

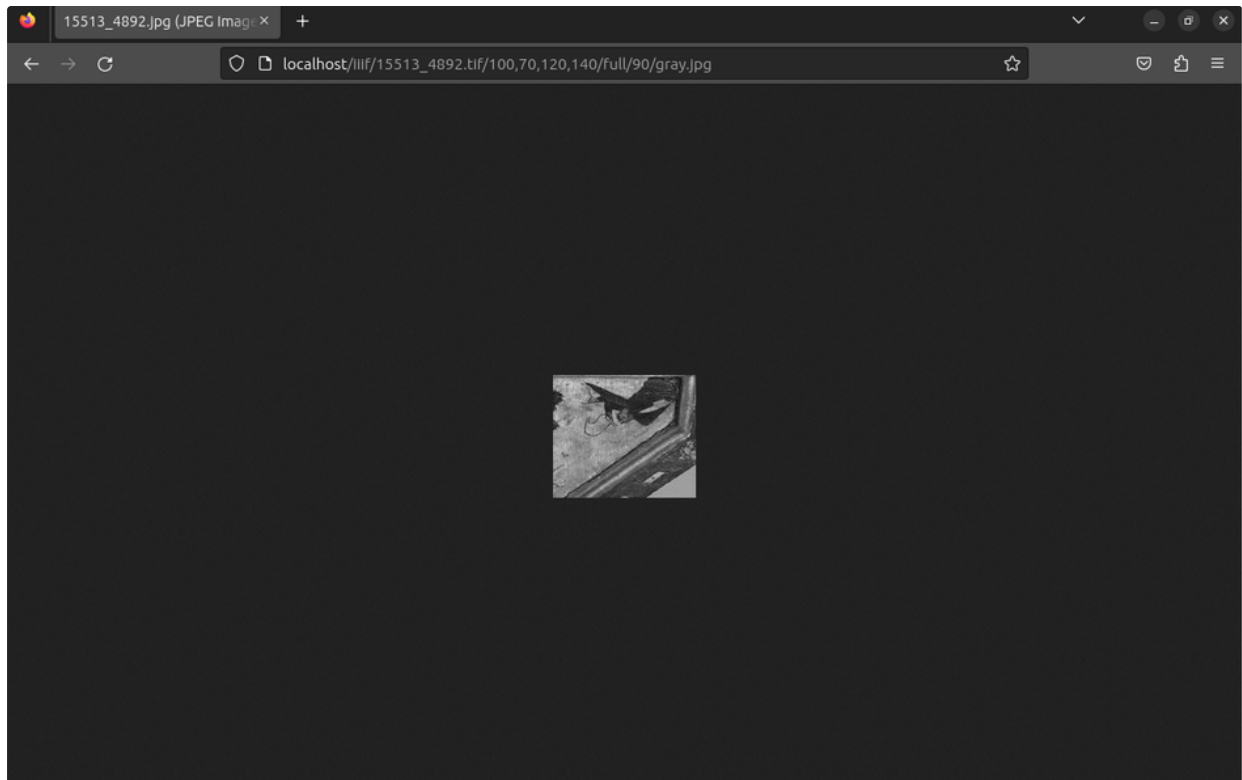


The identifiers after .tif image in the URL represent the required image request that must be in the following order: region, size, rotation and quality. "The order of the parameters is also intended as a mnemonic for the order of the operations by which the service should manipulate the image content. Thus, the requested image content is first extracted as a region of the complete image, then scaled to the requested size, mirrored and/or rotated, and finally transformed into the colour quality and format. This resulting image content is returned as the representation for the URI. Image and region dimensions in pixels are always given as integer numbers. Intermediate calculations may use floating point numbers and the rounding method is implementation specific. Some parameters, notably percentages, may be specified with floating point numbers. These should have at most 10 decimal digits and consist only of decimal digits and "." with a leading zero if less than 1.0."

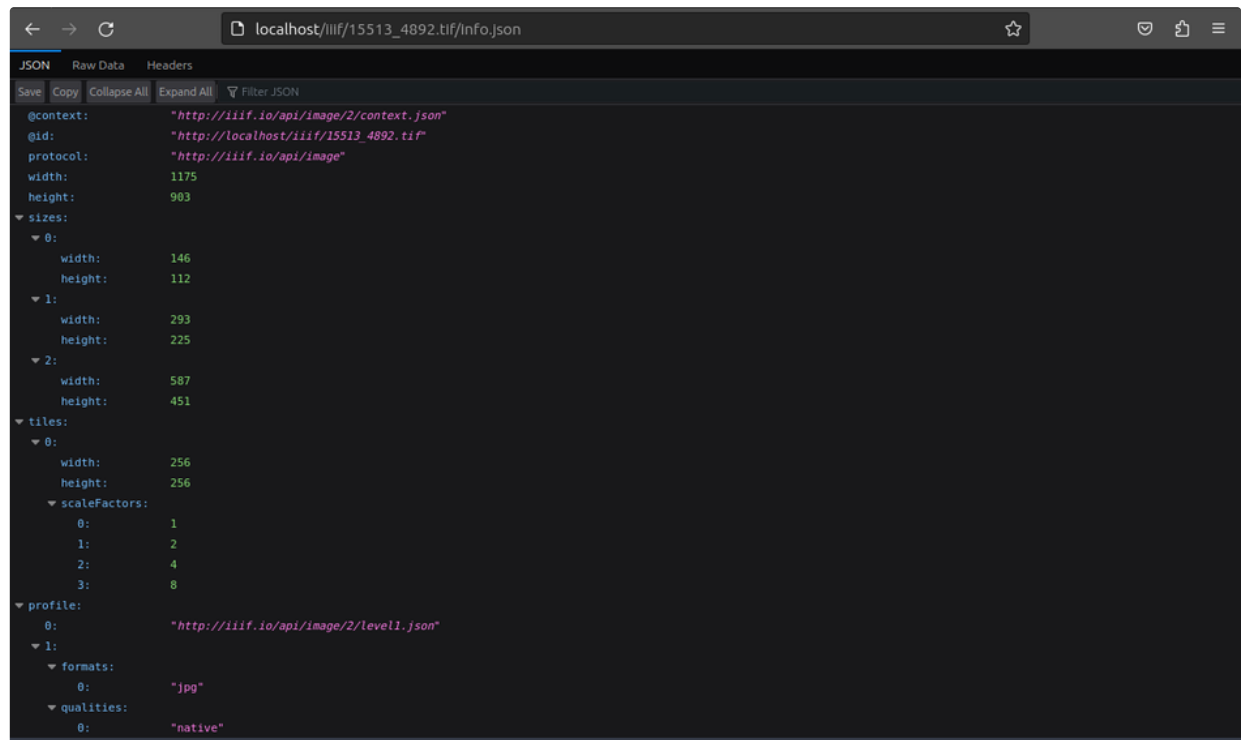
In accordance with these guidelines, the image representation on your computer can be as follows

http://localhost/iiif/15513_4892.tif/full/100,70,120,140/90/gray.jpg

With this url the region of the image consisting of pixels that have x coordinate between 100 and 140 and y coordinate between 70 and 120 is extracted. The extracted region is not scaled, and is returned at its full size (from size=full in url), rotated 90 degrees and in grayscale quality. The result of this url can be seen in the following figure.



For all TIFF images the image server creates an info.json file that can be accessed via the browser address bar and contains all necessary information (image width and height, available qualities, formats etc.) about the corresponding image.



With the adequate manifest you are able to access and view these images further with IIIF-compliant image viewers like [Mirador](#), [Universal Viewer](#), [OpenSeadragon](#) etc.

3 Memory consumption [↗](#)

The size of the resulting TIFF image depends on the size of the original image and the tiling parameter specified for the conversion. Our training test set includes 10 images, which together are 4.4 MB in size. The resulting TIFF images require significantly more disk space. In our case, each resulting TIFF image has a size between 1.9 and 7.8 MB, which adds up to a total of 52.8 MB. This is an increase of 1200%.