If we look through the http objects, we can observer, alongside other stuff, a bunch of png images with random names. Looking through a few, we can see they are qr codes in strange colour pallets. They also have a comment with format x/69 or xx/69. From this, we conclude that there are 69 qr code images, each being a character from the flag. We also get the index of them, so we need to extract the images from the capture, convert them to black-and-white, scan them, and build the flag.

To extract the images, we can simply Save All into a folder (I made one named **files**) and delete all non-image files.

Next, to convert to black and white, I had an idea: Get the colour of the first pixel, any pixel matching that colour should be white, while any pixel not matching that colour should be black.

The final script looks like this:

```
import os
import cv2
from pyzbar.pyzbar import decode
import numpy as np
from PIL import Image

folder_path = 'files'
image_files = [f for f in os.listdir(folder_path)]
final_string = [''] * 69

def extract_comment(image_path):
    with Image.open(image_path) as img:
        metadata = img.text
        return metadata.get('Comment', 'No comment found')

def scan_qr(image_path):
    image = cv2.imread(image_path)
    first_pixel = image[0, 0]
    target_color = np.array([first_pixel[0], first_pixel[1], first_pixel[2]])
    mask = np.all(image == target_color, axis=-1)
    transformed_image = np.zeros_like(image)
    transformed_image[mask] = 255
    qr_codes = decode(transformed_image)
    comment = extract_comment(image_path)

    for qr in qr_codes:
        qr_data = qr.data.decode('utf-8')
        index = int(comment.split('/')[0]) - 1
        final_string[index] = qr_data

for image_file in image_files:
    image_path = os.path.join(folder_path, image_file)
    scan_qr(image_path)
```

```
print(f"Final string: {''.join(final_string)}")
```

Made with love by: AndreiCat