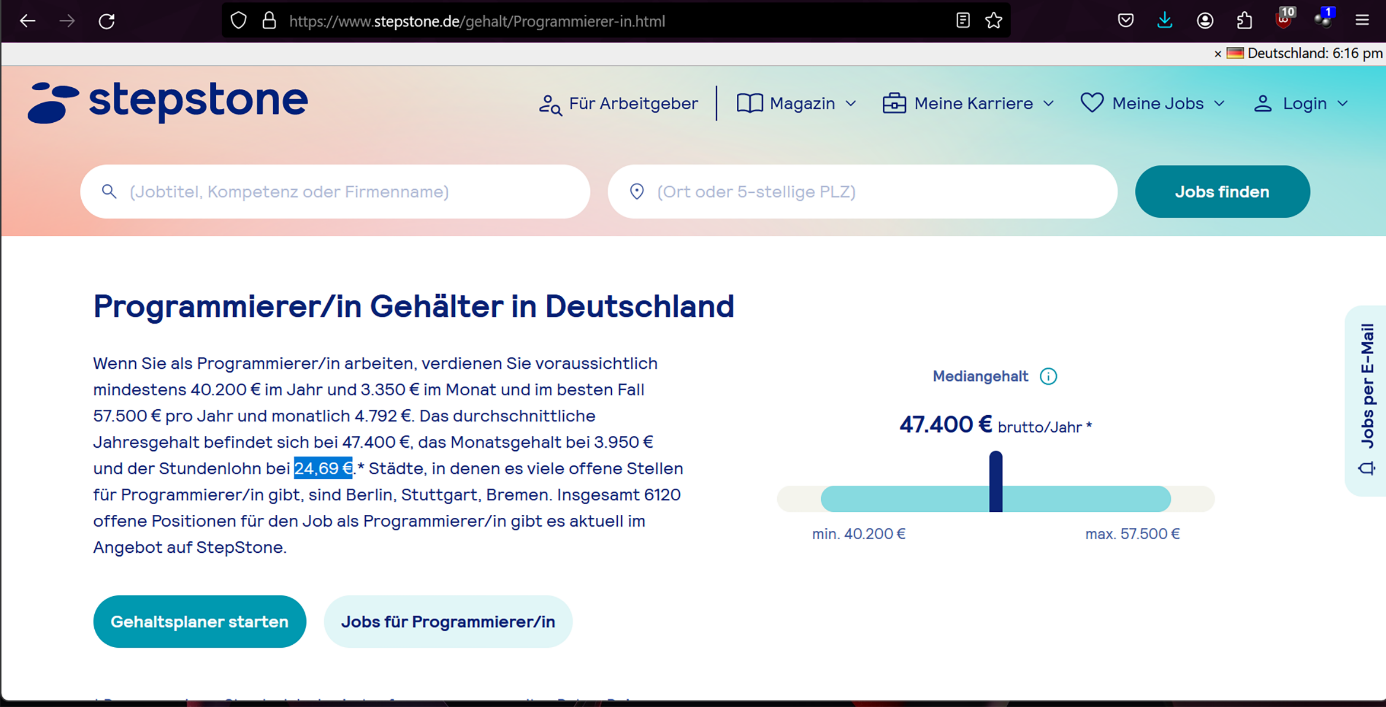
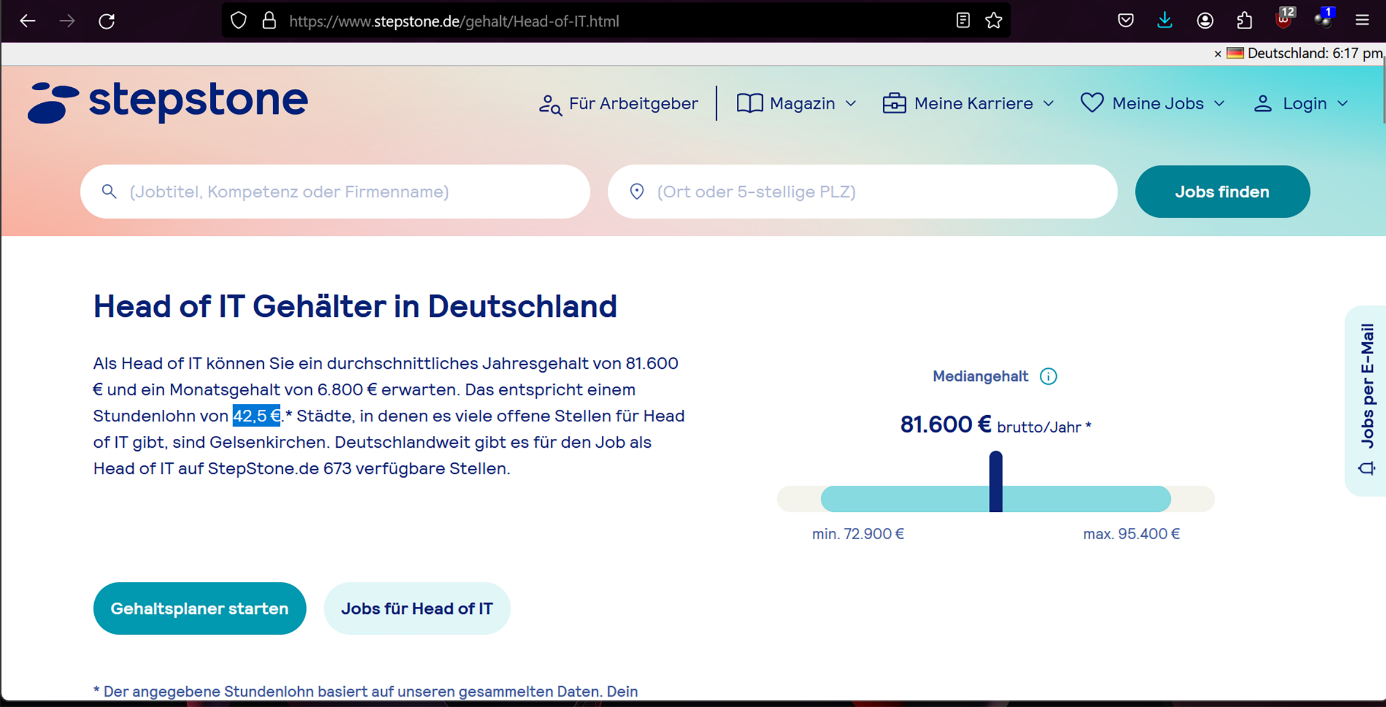
# Abbildung 1: Aktivität Diagramm

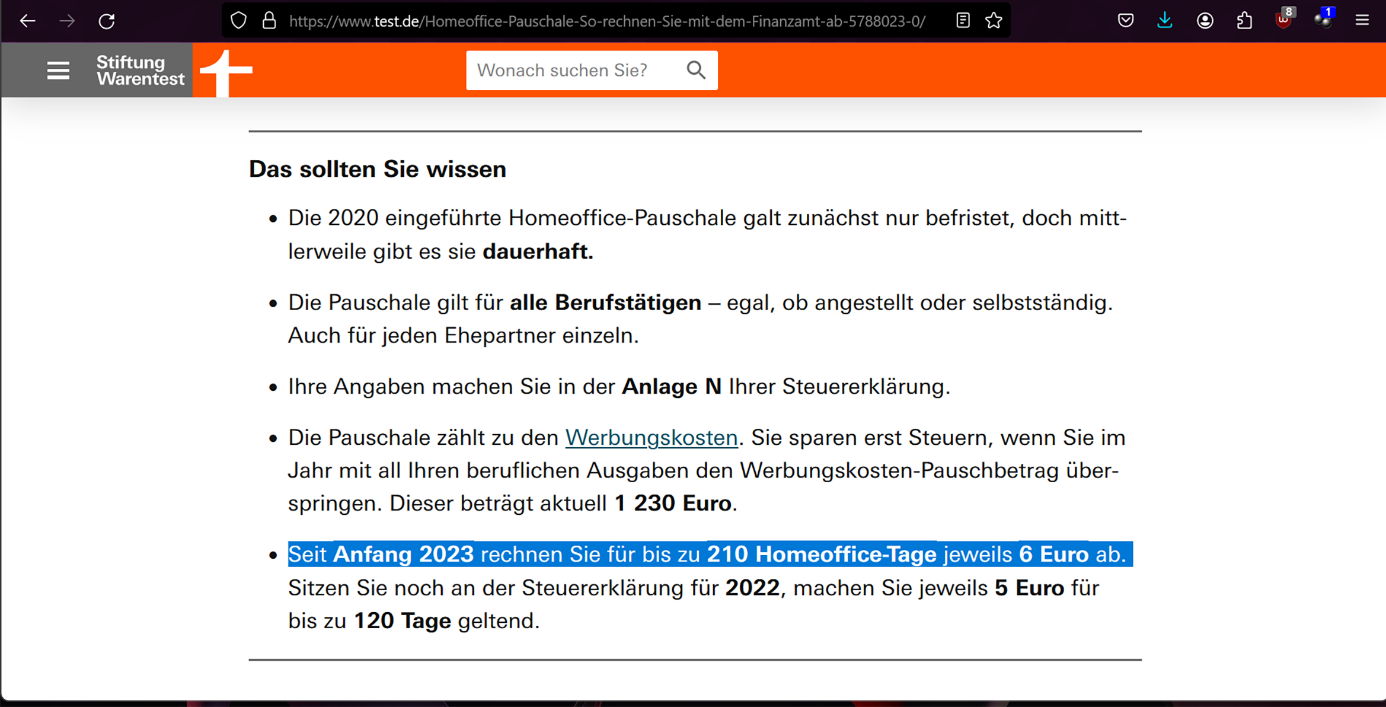
# Abbildung 2: Programmierer Stundenlohn



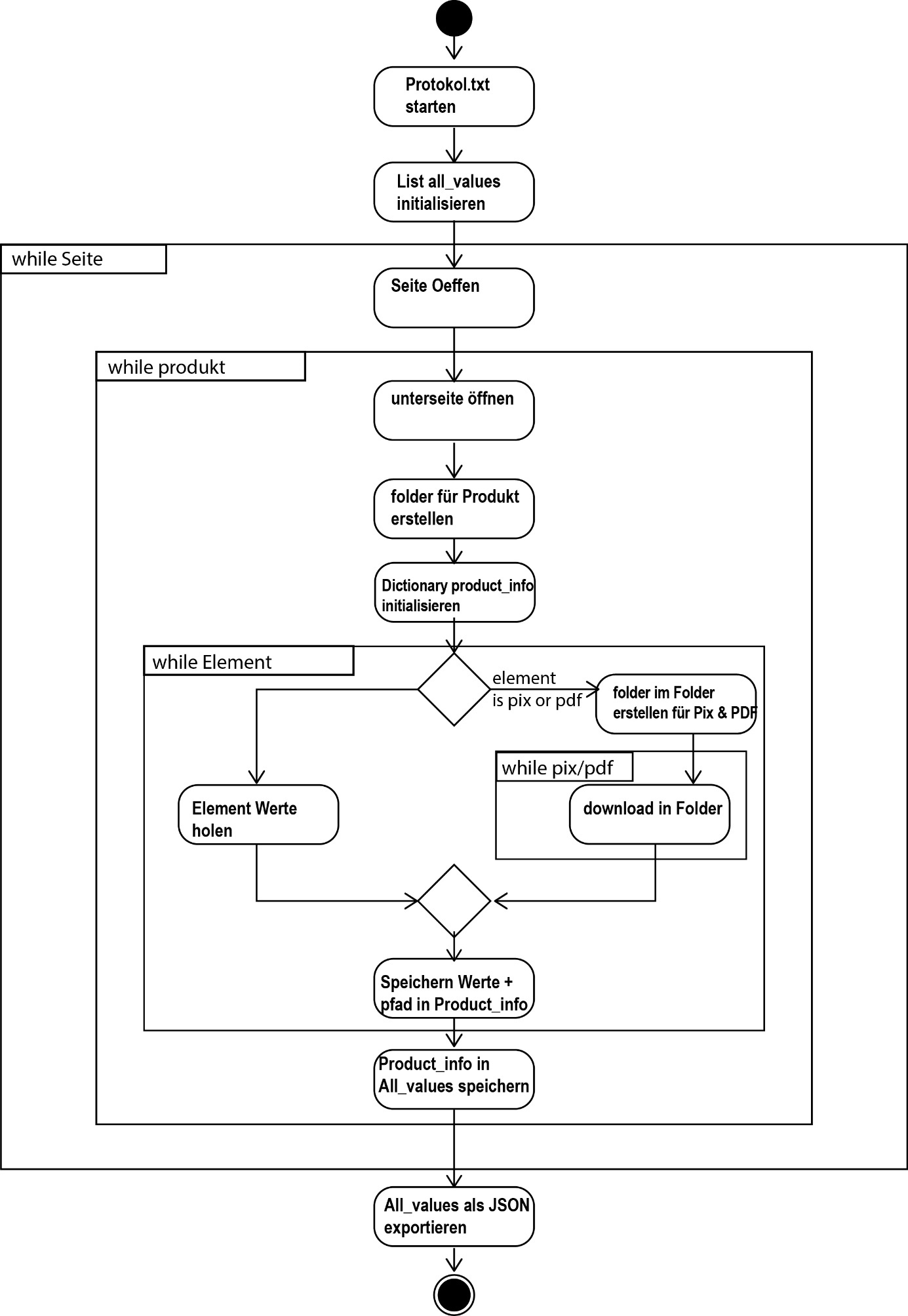
# Abbildung 3: Head of IT-Stundenlohn



# Abbildung 4: Home­office-Pauschale



# Abbildung 5: Aktivität Diagramm

****

# Abbildung 6: URL-Programmierung

#             \_

#            | |

#  \_   \_ \_ \_\_| |\_\_\_

# | | | | '\_\_| / \_\_|

# | |\_| | |  | \\_\_ \

#  \\_\_,\_|\_|  |\_|\_\_\_/

urls = [f'https://shop.buzil.de/anwendungsbereich/?p={i}&f=' for i in range(1, 14)]

all\_values = []

# Open the log file in append mode to avoid overwriting previous logs

with open(log\_file\_name, 'a') as log\_file:

    # Define a custom print function that writes to both console and log file

    def custom\_print(\*args, \*\*kwargs):

        # Convert the arguments to strings

        messages = [str(arg) for arg in args]

        # Concatenate the strings with a space separator

        message = ' '.join(messages)

        print(message, \*\*kwargs) # Print to console

        log\_file.write(message + '\n') # Write to log file

    n = 0

    subpage = 0

    for url in urls:

        n += 1

        custom\_print(f'main page {n} from 13')

        custom\_print("----------------------------------")

        custom\_print("url:", url)

        # Make an HTTP GET request to the website

        response = requests.get(url)

        # Check if the request was successful

        if response.status\_code == 200:

            # Parse the HTML content

            soup = BeautifulSoup(response.text, 'html.parser')

            # Find all <div> elements with the specified class

            item = soup.find('div', class\_='tile-view-phone')

            # todo Loop Produkts

            a\_tags = item.find\_all('a', class\_='product--link')

            for a\_tag in a\_tags:

                subpage += 1

                custom\_print('')

                custom\_print(f'page {subpage} from 145')

                product\_info = {}  # Create a new dictionary for each a\_tag

                product\_info['a'] = a\_tag['href']

                custom\_print(f'- found {product\_info["a"]}')

                custom\_print('')

# Abbildung 7: Log Output

main page 1 from 13

----------------------------------

url: https://shop.buzil.de/anwendungsbereich/?p=1&f=

page 1 from 145

- found https://shop.buzil.de/spezial/air-provence-g-565t

page 2 from 145

- found https://shop.buzil.de/kuechen-/lebensmittelhygiene/bistro-g-435

page 3 from 145

- found https://shop.buzil.de/sanitaer/bucasan-clear-g-463

page 4 from 145

- found https://shop.buzil.de/sanitaer/bucasan-fresh-p-920

page 5 from 145

- found https://shop.buzil.de/sanitaer/bucasan-hit-p-921

page 6 from 145

- found https://shop.buzil.de/sanitaer/bucasan-sanibond-g-457

page 7 from 145

- found https://shop.buzil.de/sanitaer/bucasan+-intense-p-922

page 8 from 145

- found https://shop.buzil.de/spezial/buz-air-ambra-g-564

page 9 from 145

- found https://shop.buzil.de/kuechen-/lebensmittelhygiene/buz-calc-ex-g-437

page 10 from 145

- found https://shop.buzil.de/spezial/buz-fresh-magic-g-567

page 11 from 145

- found https://shop.buzil.de/oberflaeche/buz-glass-p-912

…

# Abbildung 8: Erstellen von Ordnern für Produkte, PDFs und Bilder Programmierung

# create folder with that name

invalid\_chars = ['<', '>', ':', '"', '/', '\\', '|', '?', '\*']

# Extract the name from the href

folder\_name = os.path.basename(product\_info['a'])

# Ersetze ungültige Zeichen durch Unterstriche

for char in invalid\_chars:

    folder\_name = folder\_name.replace(char, '\_')

# Create a folder with the name from the href

folder\_path = os.path.join(os.getcwd(), folder\_name)

os.makedirs(folder\_path, exist\_ok=True)

custom\_print(f'- - making folder {folder\_path}')

# Abbildung 9: Folders Anzahl



# Abbildung 10: Download Programmierung

#... rest of code

# todo linked page

# Make an HTTP GET request to the linked page

response2 = requests.get(a\_tag['href'])

# Check if the request was successful

if response2.status\_code == 200:

# Parse the HTML content

    soup2 = BeautifulSoup(response2.text, 'html.parser')

    # Extract all <div> elements with the specified ID

    section = soup2.find('div', id='contentwrap')

    #  \_     \_\_

    # | |   /  |

    # | |\_\_ `| |

    # | '\_ \ | |

    # | | | || |\_

    # |\_| |\_\\_\_\_/

    # Extract the h1 element within the section

    h1\_element = section.find('h1')

    if h1\_element:

        h1\_text = h1\_element.get\_text().strip()

        # strip is trimming empty space

        product\_info['h1'] = h1\_text

        custom\_print(f'- - FOUND H1')

    else:

        product\_info['h1'] = ''

        custom\_print(f'+ + h1 is missing ?')

# ----------------------------------------

    # write to all values

    all\_values.append(product\_info)

#    \_

#   (\_)

#    \_ \_\_\_  \_\_\_  \_ \_\_

#   | / \_\_|/ \_ \| '\_ \

#   | \\_\_ \ (\_) | | | |

#   | |\_\_\_/\\_\_\_/|\_| |\_|

#  \_/ |

# |\_\_/

# Specify the directory path where the JSON file is located

directory\_path = ''

# Specify the file names

json\_file\_name = f'buzil\_{current\_datetime}.json'

# Save all\_values as a JSON file for each URL

with open(json\_file\_name, 'w', encoding='utf-8') as json\_file:

    json.dump(all\_values, json\_file, ensure\_ascii=False, indent=4)

print("JSON file page created.")

json\_file\_path = os.path.join(directory\_path, json\_file\_name)

# Abbildung 11: JSON-Output

[

    {

        "a": "https://shop.buzil.de/spezial/air-provence-g-565t",

        "h1": "Air Provence"

    }

]

# Abbildung 12: Protokoll

main page 1 from 13

----------------------------------

url: https://shop.buzil.de/anwendungsbereich/?p=1&f=

page 1 from 145

- found https://shop.buzil.de/spezial/air-provence-g-565t

- - making folder C:\Users\greg\Documents\sites\project\buzil\air-provence-g-565t

- - - FOUND H1

# Abbildung 13: JSON

[

    {

        "a": "https://shop.buzil.de/spezial/air-provence-g-565t",

        "h1": "Air Provence"

    },

    {

        "a": "https://shop.buzil.de/kuechen-/lebensmittelhygiene/bistro-g-435",

        "h1": "Bistro"

    },

    {

        "a": "https://shop.buzil.de/sanitaer/bucasan-clear-g-463",

        "h1": "Bucasan® Clear"

    },

    {

        "a": "https://shop.buzil.de/sanitaer/bucasan-fresh-p-920",

        "h1": "Bucasan® Fresh"

    },

…

# Abbildung 14: Protokoll

main page 1 from 13

----------------------------------

url: https://shop.buzil.de/anwendungsbereich/?p=1&f=

page 1 from 145

- found https://shop.buzil.de/spezial/air-provence-g-565t

- - making folder C:\Users\greg\Documents\sites\project\buzil\air-provence-g-565t

- - - FOUND H1

page 2 from 145

- found https://shop.buzil.de/kuechen-/lebensmittelhygiene/bistro-g-435

- - making folder C:\Users\greg\Documents\sites\project\buzil\bistro-g-435

- - - FOUND H1

page 3 from 145

- found https://shop.buzil.de/sanitaer/bucasan-clear-g-463

- - making folder C:\Users\greg\Documents\sites\project\buzil\bucasan-clear-g-463

- - - FOUND H1

page 4 from 145

- found https://shop.buzil.de/sanitaer/bucasan-fresh-p-920

- - making folder C:\Users\greg\Documents\sites\project\buzil\bucasan-fresh-p-920

- - - FOUND H1

page 5 from 145

- found https://shop.buzil.de/sanitaer/bucasan-hit-p-921

- - making folder C:\Users\greg\Documents\sites\project\buzil\bucasan-hit-p-921

- - - FOUND H1

…

# Abbildung 15: JSON-Output

[

    {

        "a": "https://shop.buzil.de/spezial/air-provence-g-565t",

        "h1": "Air Provence G 565T",

        "image\_urls": [

            "https://shop.buzil.de/media/image/0a/1a/ca/2QOLn7huZ4R9dO1\_1280x1280.jpg",

            "https://shop.buzil.de/media/image/c8/8b/07/2QOLn7huZ4R9dO1\_200x200.jpg",

            "https://shop.buzil.de/media/image/7e/d6/56/2QOLn7huZ4R9dO1.jpg",

            "https://shop.buzil.de/media/image/5d/0e/77/2QOLn7huZ4R9dO1\_600x600.jpg",

            "https://shop.buzil.de/media/image/3b/f1/21/2QOLn7huZ4R9dO1\_600x600@2x.jpg"

        ],

        "image\_local": [

            "air-provence-g-565t/img/2QOLn7huZ4R9dO1\_1280x1280.jpg",

            "air-provence-g-565t/img/2QOLn7huZ4R9dO1\_200x200.jpg",

            "air-provence-g-565t/img/2QOLn7huZ4R9dO1.jpg",

            "air-provence-g-565t/img/2QOLn7huZ4R9dO1\_600x600.jpg",

            "air-provence-g-565t/img/2QOLn7huZ4R9dO1\_600x600@2x.jpg"

        ],

        "product\_detail\_description\_section": [

            {

                "header": "Gebrauchsfertiger Raumspray mit Aktiv-Geruchsblocker",

                "content": "In Wohn- und Geschäftsräumen, öffentlichen Einrichtungen wie Krankenhäusern, Altenheimen sowie in der Gastronomie, Sanitär- und WC-Anlagen, Aufenthalts- und Umkleideräumen, Sport- und Wellnessbereichen etc.; für die schnelle und langanhaltende Geruchsvernichtung und Luftverbesserung"

            },

            {

                "header": "Produktvorteile",

                "content": [

                    "Aromatischer Duft, der an frisch geernteten Lavendel erinnert sorgt für eine angenehme Raumatmosphäre",

                    "Neutralisiert schnell unangenehme Gerüche mit Aktiv-Geruchsblocker",

                    "Hinterlässt keine Flecken",

                    "Umweltfreundlicher Zerstäuber ohne Treibgas"

                ]

            }

        ],

…

# Abbildung 16: Protokoll

main page 1 from 13

----------------------------------

url: https://shop.buzil.de/anwendungsbereich/?p=1&f=

page 1 from 145

- found https://shop.buzil.de/spezial/air-provence-g-565t

- - making folder C:\Users\greg\Documents\sites\project\buzil\air-provence-g-565t

- - - FOUND H1

- - - DESCRIPTION-DATA

- - - - downloading description\_data: 1

- - - - downloading description\_data: 2

- - - ADDITIONAL DATA

- - - - downloading additionail detail: 1

- - - - - downloading additionail detail

- - - - downloading additionail detail: 2

- - - - - downloading additionail detail

- - - HINT

- - - DOWNLOADING PRODUKKT DOWNLOAD TITLE 1

- - - PRODUKT DOWNLOAD CONTENT

- - - - product\_download\_content found

- - - - - product\_download\_content li found: 1

- - - - - - product\_download li

- - - - - - - downloading text and pdf for Betriebsanweisungen

- - - - - - - - downloading PDF for Betriebsanweisungen

- - - - - product\_download\_content li found: 2

…

Abbildung 17: Produkt CSV Import Schema A white paper with black text

Description automatically generated

A white sheet of paper with black text

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Abbildung 18: Aktivität Diagramm

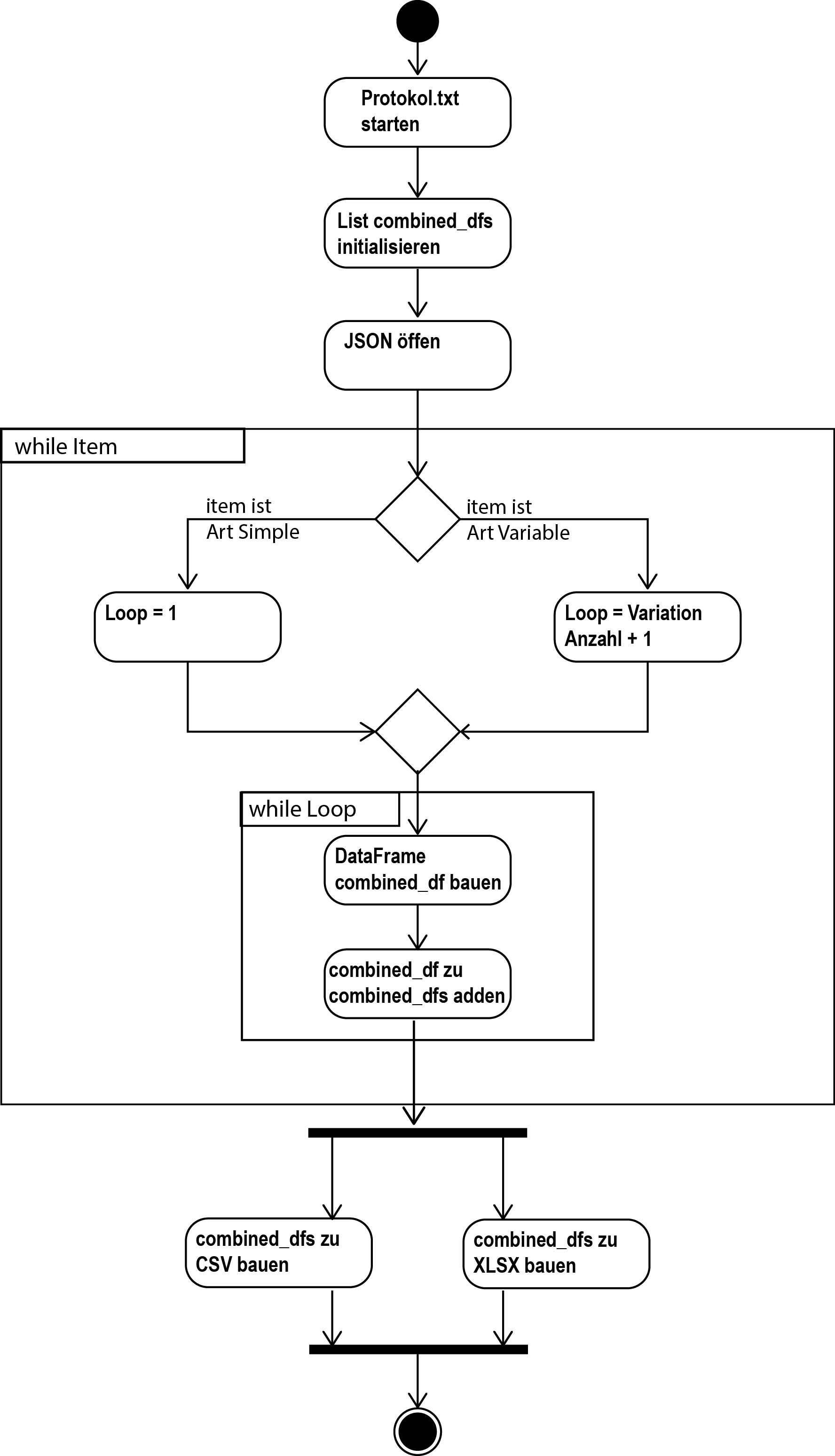


Abbildung 19: CSV Output

A screen shot of a computer

Description automatically generated

Abbildung 20: WooCommerce Import GUI 1

**A screenshot of a computer

Description automatically generated**

Abbildung 21: WooCommerce Import GUI 2

**A screenshot of a computer

Description automatically generated**

Abbildung 22: WooCommerce Import GUI 3

**A screenshot of a computer

Description automatically generated**

Abbildung 23: XLSX zu CSV

import pandas as pd

# this will reformat xlsx to an csv file

# it s well suited for german, and will respect leading zeros for article numbers

# .\_\_                 \_\_                        \_\_  .\_\_

# |\_\_| \_\_\_\_   \_\_\_\_\_\_\_/  |\_\_\_\_\_\_\_\_ \_\_ \_\_   \_\_\_\_\_/  |\_|\_\_| \_\_\_\_   \_\_\_\_   \_\_\_\_\_\_

# |  |/    \ /  \_\_\_/\   \_\_\\_  \_\_ \  |  \\_/ \_\_\_\   \_\_\  |/  \_ \ /    \ /  \_\_\_/

# |  |   |  \\\_\_\_ \  |  |  |  | \/  |  /\  \\_\_\_|  | |  (  <\_> )   |  \\\_\_\_ \

# |\_\_|\_\_\_|  /\_\_\_\_  > |\_\_|  |\_\_|  |\_\_\_\_/  \\_\_\_  >\_\_| |\_\_|\\_\_\_\_/|\_\_\_|  /\_\_\_\_  >

#         \/     \/                          \/                    \/     \/

# 1 put name of input File xlsx oder ods

# 2 put name of output csv file

# 1 Read the Excel file

df = pd.read\_excel('file\_name.xlsx', dtype=str) # Replace with your file name/path

# Keep only the 'Beschreibung' 'Kurzbeschreibung' columns or column needed

df = df[['Artikelnummer', 'Beschreibung', 'Kurzbeschreibung']]

# 3 Write to a CSV file

df.to\_csv('file\_name.csv', index=False, encoding='utf-8')

# Adjust the output file name as needed

Abbildung 24: WooCommerce Fehler Meldung A screenshot of a computer

Description automatically generated

Contents

**No table of contents entries found.**