University of Basel

RISE Crash Course: "Practical Basics of LLMs over APIs"

Sorin Marti, Lea Kasper (RISE & EIB), 05.03.2025

Who is Research and Infrastructure Support (RISE)?

“We support researchers in the humanities and social sciences at the University of Basel in the conception of computer-based research, the creation, analysis and user-oriented presentation of digital data, as well as in sustainable and open methods of data dissemination.”

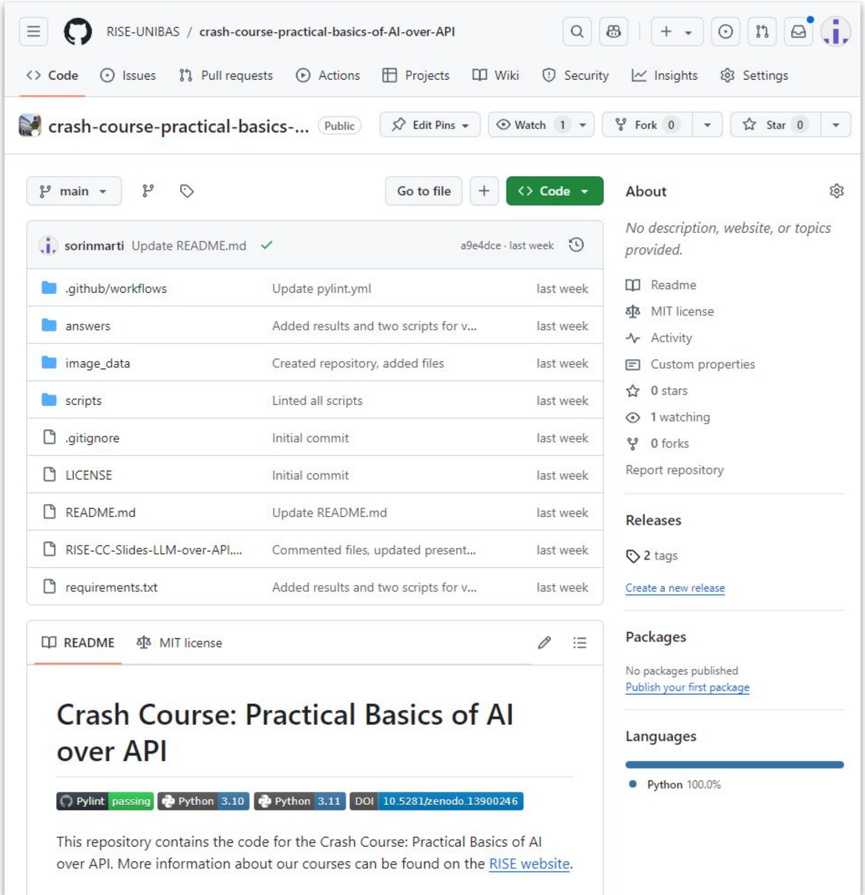
rise@unibas.ch

Agenda

1. Prepare
   a. Conceptualize a workflow → What data will be processed how in which steps?
   b. Test your prompt and source material → Does it work in a browser?
   c. Choose a Provider, Get API Access → Which provider is best for your needs?
   d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
   a. Write requests and pass context → How to write a simple python script which does X requests?
   b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
   a. Output formatting → How do I tell the LLM to format output data?
   b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Course Materials

- All examples, scripts and the slides are available in a GitHub repository and over Zenodo.
- There is a step-by-step tutorial on how to try the example by yourself or adapt it for your own research.
- You will have to create your own API keys in order to use the examples. Be aware that all prices and ratios mentioned in this presentation are changing constantly.



sorinmarti Update README.md ✓ a9e4dce · last week

.github/workflows	Update pylint.yml	last week
answers	Added results and two scripts for v...	last week
image_data	Created repository, added files	last week
scripts	Linted all scripts	last week
.gitignore	Initial commit	last week
LICENSE	Initial commit	last week
README.md	Update README.md	last week
RISE-CC-Slides-LLM-over-API....	Commented files, updated present...	last week
requirements.txt	Added results and two scripts for v...	last week

Crash Course: Practical Basics of AI over API

This repository contains the code for the Crash Course: Practical Basics of AI over API. More information about our courses can be found on the RISE website.

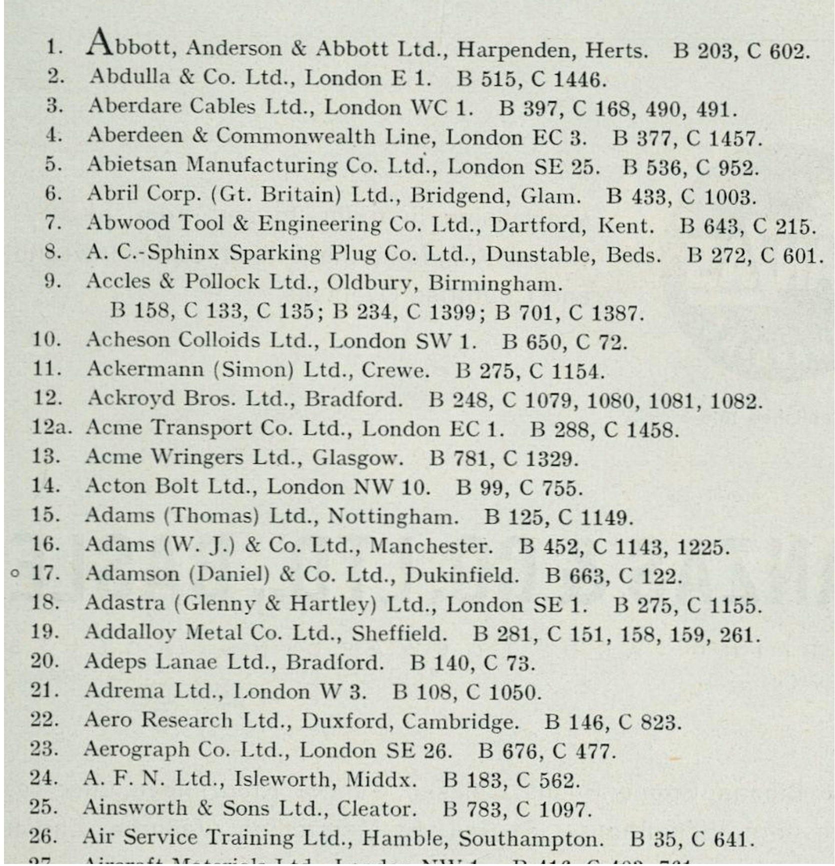
Agenda

1. Prepare
    a. **Conceptualize a workflow** → **What data will be processed how in which steps?**
    b. Test your prompt and source material → Does it work in a browser?
    c. Choose a Provider, Get API Access → Which provider is best for your needs?
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to format output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Example Data

- Source: *Index of Swiss agents representing United Kingdom firms* published by the British Chamber of Commerce for Switzerland in 1951
- Well formatted list-entries in three sections: A,B,C. A: British Firms, B: Swiss Firms, C: Goods Categories
- Identifiers at the end of the line: connections to other firms or goods categories
- This data forms a network

https://doi.org/10.7891/e-manuscripta-136029



1. Abbott, Anderson & Abbott Ltd., Harpenden, Herts. B 203, C 602.
2. Abdulla & Co. Ltd., London E 1. B 515, C 1446.
3. Aberdare Cables Ltd., London WC 1. B 397, C 168, 490, 491.
4. Aberdeen & Commonwealth Line, London EC 3. B 377, C 1457.
5. Abietsan Manufacturing Co. Ltd., London SE 25. B 536, C 952.
6. Abril Corp. (Gt. Britain) Ltd., Bridgend, Glam. B 433, C 1003.
7. Abwood Tool & Engineering Co. Ltd., Dartford, Kent. B 643, C 215.
8. A. C.-Sphinx Sparking Plug Co. Ltd., Dunstable, Beds. B 272, C 601.
9. Accles & Pollock Ltd., Oldbury, Birmingham.
   B 158, C 133, C 135; B 234, C 1399; B 701, C 1387.
10. Acheson Colloids Ltd., London SW 1. B 650, C 72.
11. Ackermann (Simon) Ltd., Crewe. B 275, C 1154.
12. Ackroyd Bros. Ltd., Bradford. B 248, C 1079, 1080, 1081, 1082.

12a. Acme Transport Co. Ltd., London EC 1. B 288, C 1458.

13. Acme Wringers Ltd., Glasgow. B 781, C 1329.
14. Acton Bolt Ltd., London NW 10. B 99, C 755.
15. Adams (Thomas) Ltd., Nottingham. B 125, C 1149.
16. Adams (W. J.) & Co. Ltd., Manchester. B 452, C 1143, 1225.

○ 17. Adamson (Daniel) & Co. Ltd., Dukinfield. B 663, C 122.

18. Adastra (Glenny & Hartley) Ltd., London SE 1. B 275, C 1155.
19. Addalloy Metal Co. Ltd., Sheffield. B 281, C 151, 158, 159, 261.
20. Adeps Lanae Ltd., Bradford. B 140, C 73.
21. Adrema Ltd., London W 3. B 108, C 1050.
22. Aero Research Ltd., Duxford, Cambridge. B 146, C 823.
23. Aerograph Co. Ltd., London SE 26. B 676, C 477.
24. A. F. N. Ltd., Isleworth, Middx. B 183, C 562.
25. Ainsworth & Sons Ltd., Cleator. B 783, C 1097.
26. Air Service Training Ltd., Hamble, Southampton. B 35, C 641.

Conceptualize a Workflow

- What data do you have?

  “197 jpg images of a printed book. They contain lists of members of a chamber of commerce. Some of the images are not part of the list but are advertisements and such. The book is in English, it is from 1951”.

- What information do you want to extract from this data?

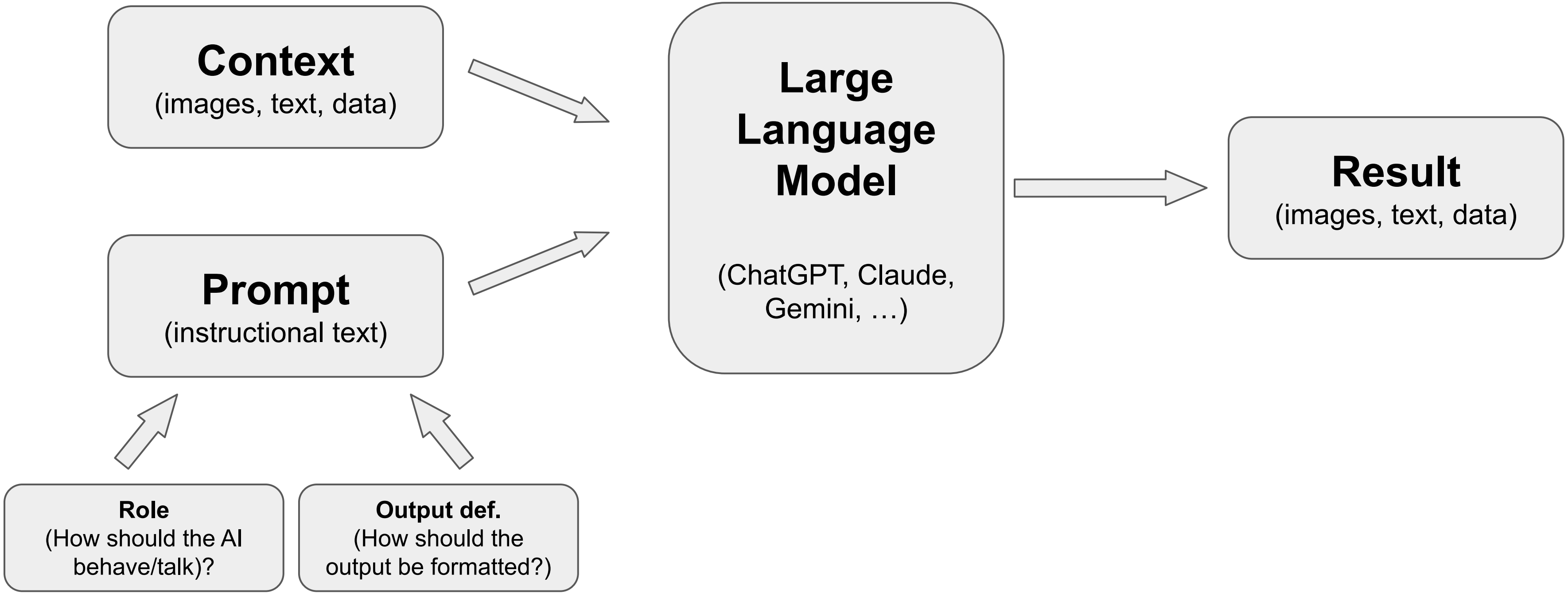
  “Identifiers, Company names, Location names, Network information.”

- How should a response be structured?

  “A list of standardized JSON entries to create networks from.”

rise@unibas.ch

Parts of an LLM request



Context
(images, text, data)

Prompt
(instructional text)

Role
(How should the AI behave/talk)?

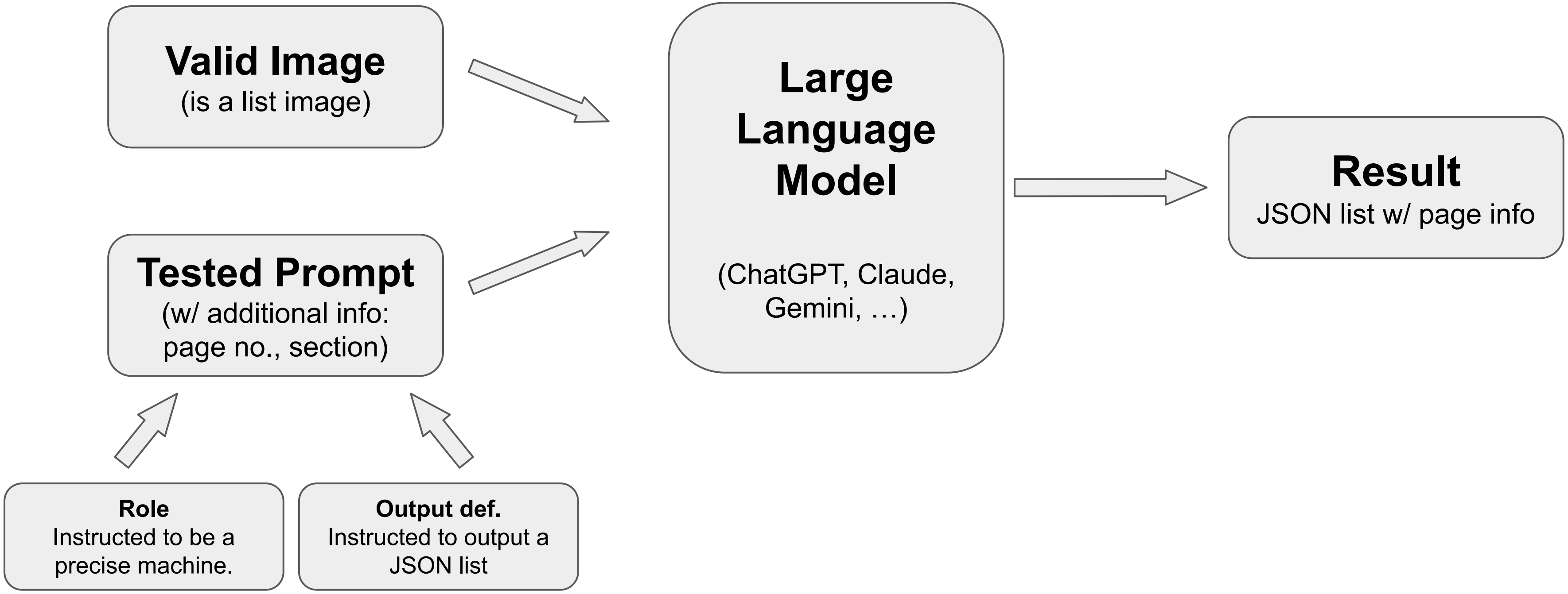
Output def.
(How should the output be formatted?)

Large Language Model

(ChatGPT, Claude, Gemini, …)

Result
(images, text, data)

Example LLM request



Valid Image
(is a list image)

Tested Prompt
(w/ additional info:
page no., section)

Role
Instructed to be a
precise machine.

Output def.
Instructed to output a
JSON list

Large Language Model

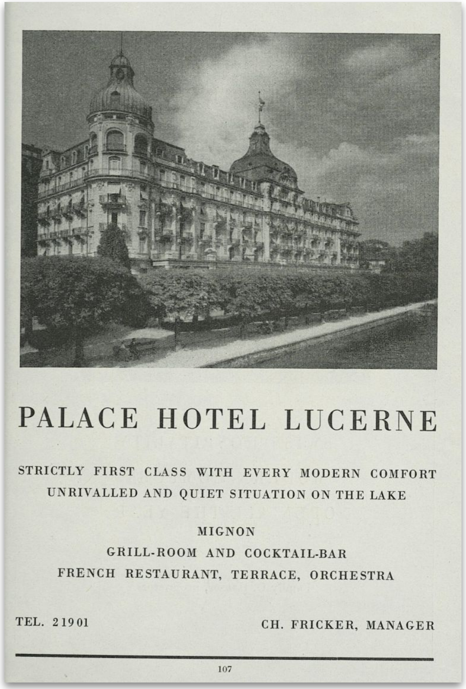
(ChatGPT, Claude,
Gemini, …)

Result
JSON list w/ page info

rise@unibas.ch

Revisit the Workflow: Tasks

- Download all the images from e-manuscripta
- Sort out images which are not part of the list
- Engineer a prompt to extract the needed information
- For each valid image:
  - Send a request to an LLM with the image as context
  - Save the response in a meaningful way
- Merge all the information into a dataset
- Random sampling to assure accuracy



PALACE HOTEL LUCERNE

STRICTLY FIRST CLASS WITH EVERY MODERN COMFORT
UNRIVALLED AND QUIET SITUATION ON THE LAKE

MIGNON
GRILL-ROOM AND COCKTAIL-BAR
FRENCH RESTAURANT, TERRACE, ORCHESTRA

TEL. 21901 CH. FRICKER, MANAGER

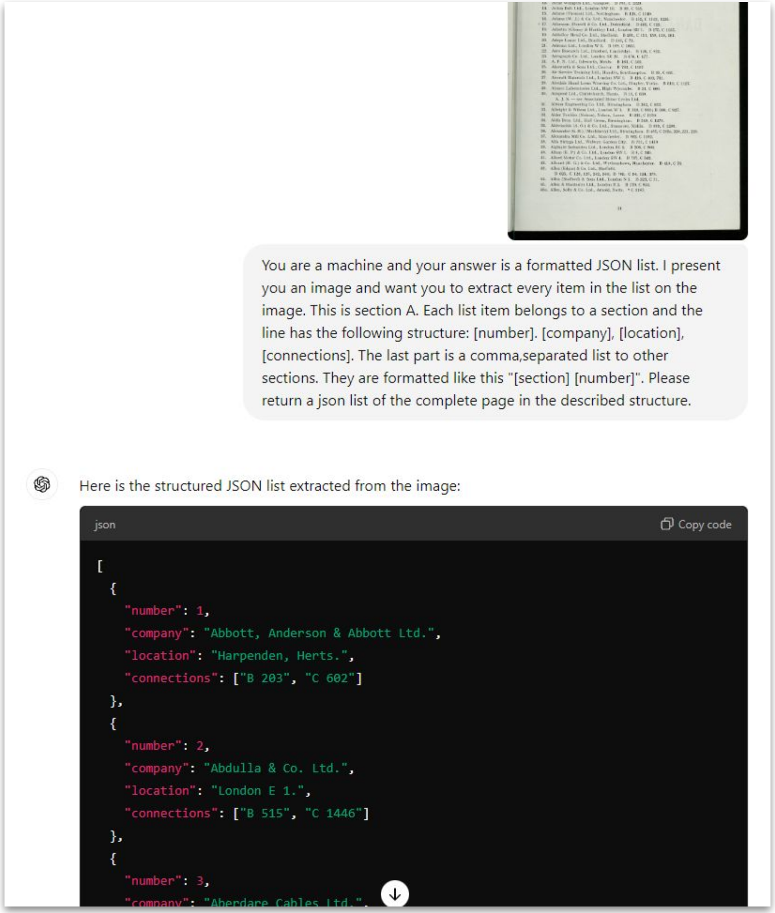
107

Agenda

1. Prepare
    a. Conceptualize a workflow → What data will be processed how in which steps?
    b. Test your prompt and source material → Does it work in a browser?
    c. Choose a Provider, Get API Access → Which provider is best for your needs?
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Test in Browser: ChatGPT

- Present one single image and explain the task, i.e. design the prompt.
- This lets you work on your prompt and test results.
- Good prompt engineering is crucial!



You are a machine and your answer is a formatted JSON list. I present you an image and want you to extract every item in the list on the image. This is section A. Each list item belongs to a section and the line has the following structure: [number]. [company], [location], [connections]. The last part is a comma,separated list to other sections. They are formatted like this "[section] [number]". Please return a json list of the complete page in the described structure.

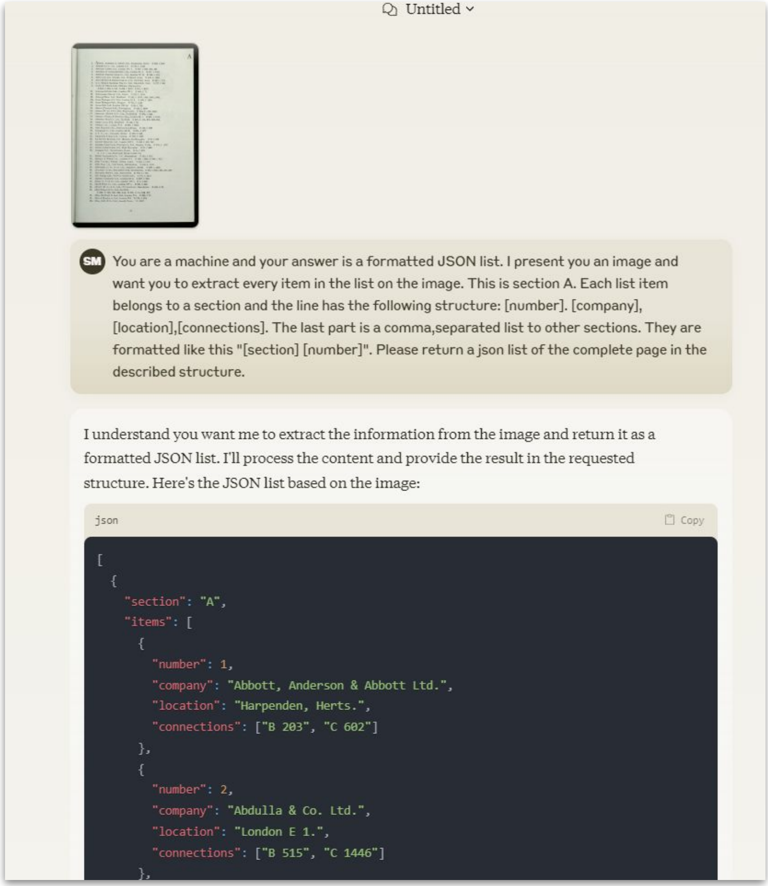
Here is the structured JSON list extracted from the image:

```json
[
  {
    "number": 1,
    "company": "Abbott, Anderson & Abbott Ltd.",
    "location": "Harpenden, Herts.",
    "connections": ["B 203", "C 602"]
  },
  {
    "number": 2,
    "company": "Abdulla & Co. Ltd.",
    "location": "London E 1.",
    "connections": ["B 515", "C 1446"]
  },
  {
    "number": 3,
    "company": "Aberdare Cables Ltd.",
```

ChatGPT 4o, 06.10.24

Test in Browser: Claude

- You might experience character limit issues, input format restrictions etc. The API can deal with that.
- You can ask the ChatBot if it stopped b/c of limitations.



You are a machine and your answer is a formatted JSON list. I present you an image and want you to extract every item in the list on the image. This is section A. Each list item belongs to a section and the line has the following structure: [number]. [company], [location],[connections]. The last part is a comma,separated list to other sections. They are formatted like this "[section] [number]". Please return a json list of the complete page in the described structure.

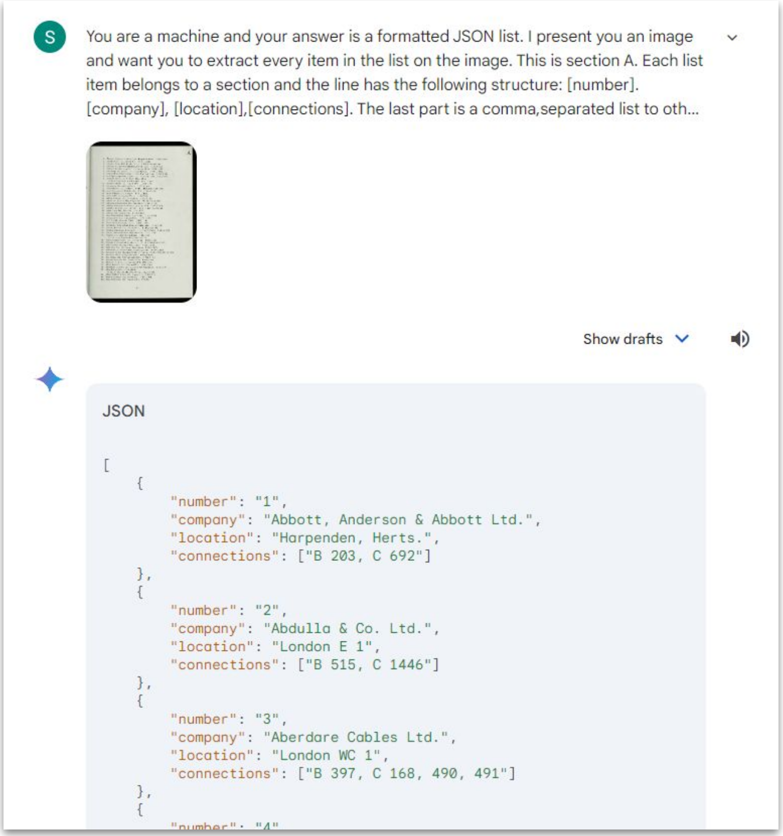
I understand you want me to extract the information from the image and return it as a formatted JSON list. I'll process the content and provide the result in the requested structure. Here's the JSON list based on the image:

```json
[
  {
    "section": "A",
    "items": [
      {
        "number": 1,
        "company": "Abbott, Anderson & Abbott Ltd.",
        "location": "Harpenden, Herts.",
        "connections": ["B 203", "C 602"]
      },
      {
        "number": 2,
        "company": "Abdulla & Co. Ltd.",
        "location": "London E 1.",
        "connections": ["B 515", "C 1446"]
      },
```

Claude 3.5 Sonnet, 06.10.24

Test in Browser: Gemini

- You can also compare LLMs using a comparison Tool
- Msty, LM-Studio
- Formulate one prompt, upload one image and send it to all wanted models.
- Works with local and online providers!



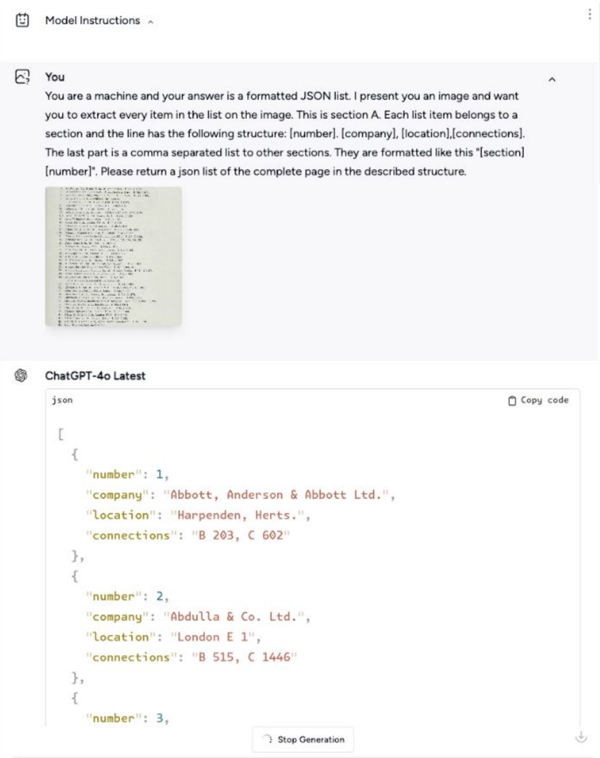
You are a machine and your answer is a formatted JSON list. I present you an image and want you to extract every item in the list on the image. This is section A. Each list item belongs to a section and the line has the following structure: [number]. [company], [location],[connections]. The last part is a comma,separated list to oth...

JSON

```json
[
    {
        "number": "1",
        "company": "Abbott, Anderson & Abbott Ltd.",
        "location": "Harpenden, Herts.",
        "connections": ["B 203, C 692"]
    },
    {
        "number": "2",
        "company": "Abdulla & Co. Ltd.",
        "location": "London E 1",
        "connections": ["B 515, C 1446"]
    },
    {
        "number": "3",
        "company": "Aberdare Cables Ltd.",
        "location": "London WC 1",
        "connections": ["B 397, C 168, 490, 491"]
    },
    {
        "number": "4"
```

Gemini 1.5 Flash, 06.10.24

Test Models: Msty

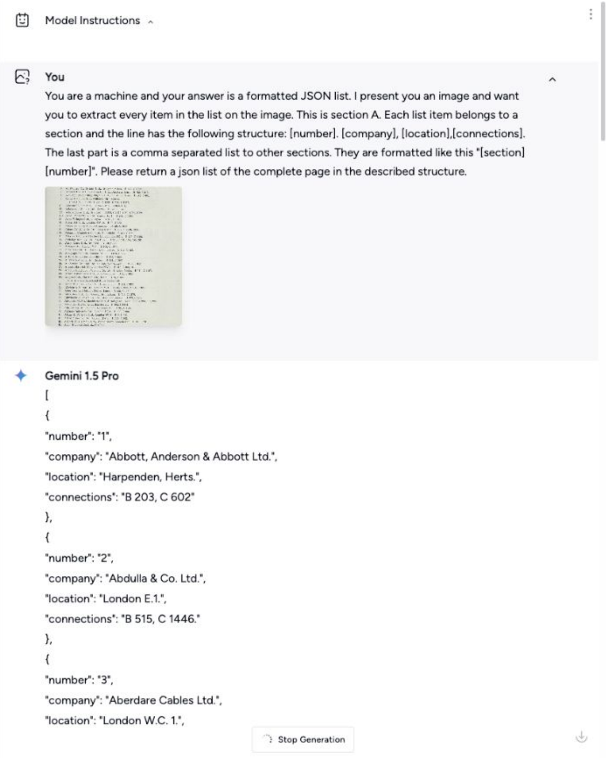


You

You are a machine and your answer is a formatted JSON list. I present you an image and want you to extract every item in the list on the image. This is section A. Each list item belongs to a section and the line has the following structure: [number]. [company], [location],[connections]. The last part is a comma separated list to other sections. They are formatted like this "[section] [number]". Please return a json list of the complete page in the described structure.

ChatGPT-4o Latest

```json
[
  {
    "number": 1,
    "company": "Abbott, Anderson & Abbott Ltd.",
    "location": "Harpenden, Herts.",
    "connections": "B 203, C 602"
  },
  {
    "number": 2,
    "company": "Abdulla & Co. Ltd.",
    "location": "London E 1",
    "connections": "B 515, C 1446"
  },
  {
    "number": 3,
```

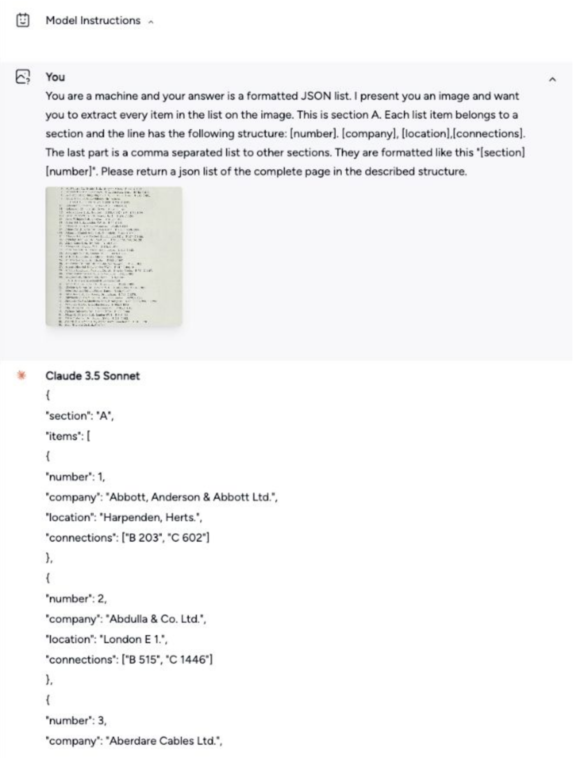


You

You are a machine and your answer is a formatted JSON list. I present you an image and want you to extract every item in the list on the image. This is section A. Each list item belongs to a section and the line has the following structure: [number]. [company], [location],[connections]. The last part is a comma separated list to other sections. They are formatted like this "[section] [number]". Please return a json list of the complete page in the described structure.

Gemini 1.5 Pro

```
[
{
"number": "1",
"company": "Abbott, Anderson & Abbott Ltd.",
"location": "Harpenden, Herts.",
"connections": "B 203, C 602"
},
{
"number": "2",
"company": "Abdulla & Co. Ltd.",
"location": "London E.1.",
"connections": "B 515, C 1446."
},
{
"number": "3",
"company": "Aberdare Cables Ltd.",
"location": "London W.C. 1.",
```



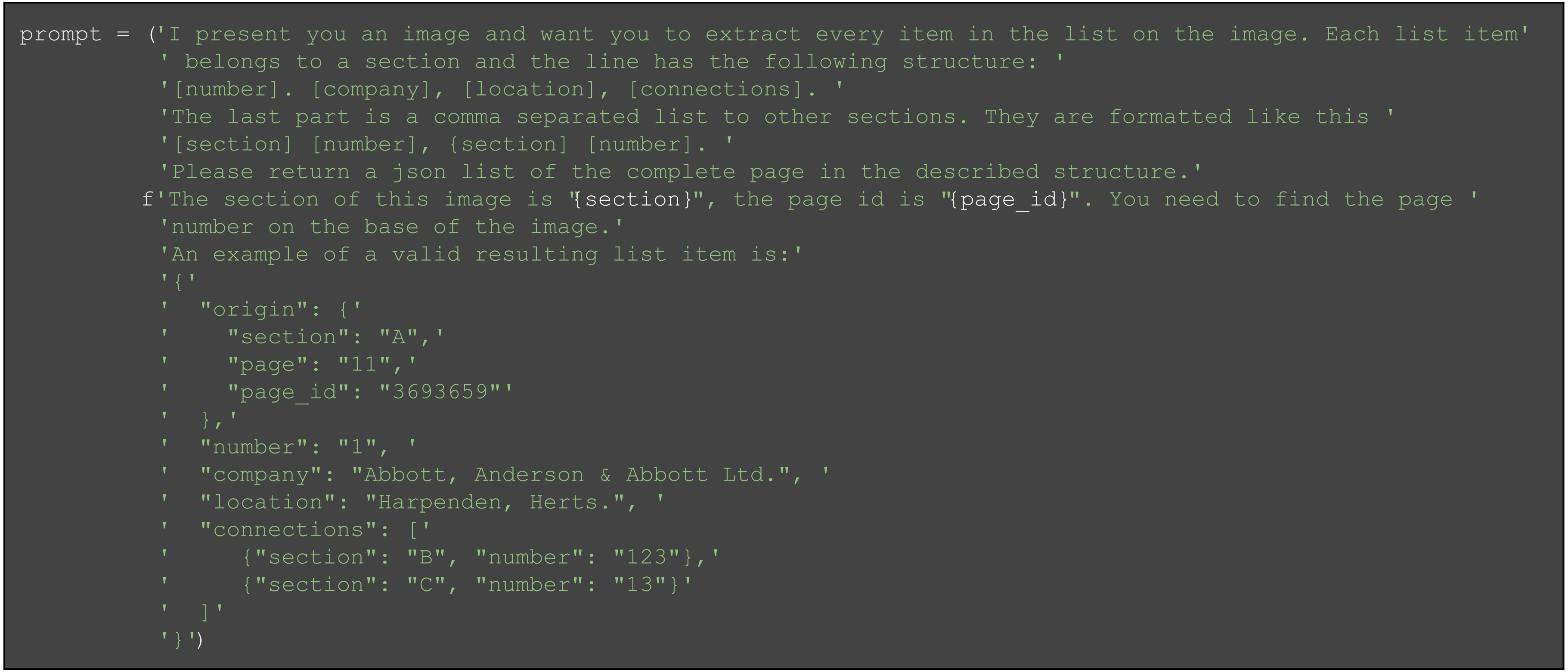
You

You are a machine and your answer is a formatted JSON list. I present you an image and want you to extract every item in the list on the image. This is section A. Each list item belongs to a section and the line has the following structure: [number]. [company], [location],[connections]. The last part is a comma separated list to other sections. They are formatted like this "[section] [number]". Please return a json list of the complete page in the described structure.

Claude 3.5 Sonnet

```
{
"section": "A",
"items": [
{
"number": 1,
"company": "Abbott, Anderson & Abbott Ltd.",
"location": "Harpenden, Herts.",
"connections": ["B 203", "C 602"]
},
{
"number": 2,
"company": "Abdulla & Co. Ltd.",
"location": "London E 1.",
"connections": ["B 515", "C 1446"]
},
{
"number": 3,
"company": "Aberdare Cables Ltd.",
```

Example Prompt



```
prompt = ('I present you an image and want you to extract every item in the list on the image. Each list item'
          ' belongs to a section and the line has the following structure: '
          '[number]. [company], [location], [connections]. '
          'The last part is a comma separated list to other sections. They are formatted like this '
          '[section] [number], {section] [number]. '
          'Please return a json list of the complete page in the described structure.'
         f'The section of this image is "{section}", the page id is "{page_id}". You need to find the page '
          'number on the base of the image.'
          'An example of a valid resulting list item is:'
          '{'
          '  "origin": {'
          '    "section": "A",'
          '    "page": "11",'
          '    "page_id": "3693659"'
          '  },'
          '  "number": "1", '
          '  "company": "Abbott, Anderson & Abbott Ltd.", '
          '  "location": "Harpenden, Herts.", '
          '  "connections": ['
          '     {"section": "B", "number": "123"},'
          '     {"section": "C", "number": "13"}'
          '  ]'
          '}')
```

Agenda

1. Prepare
    a. Conceptualize a workflow → What data will be processed how in which steps?
    b. Test your prompt and source material → Does it work in a browser?
    c. **Choose a Provider, Get API Access** → **Which provider is best for your needs?**
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to format output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Choose a provider

Claude

BY ANTHROP\C

Gemini

GitHub Copilot

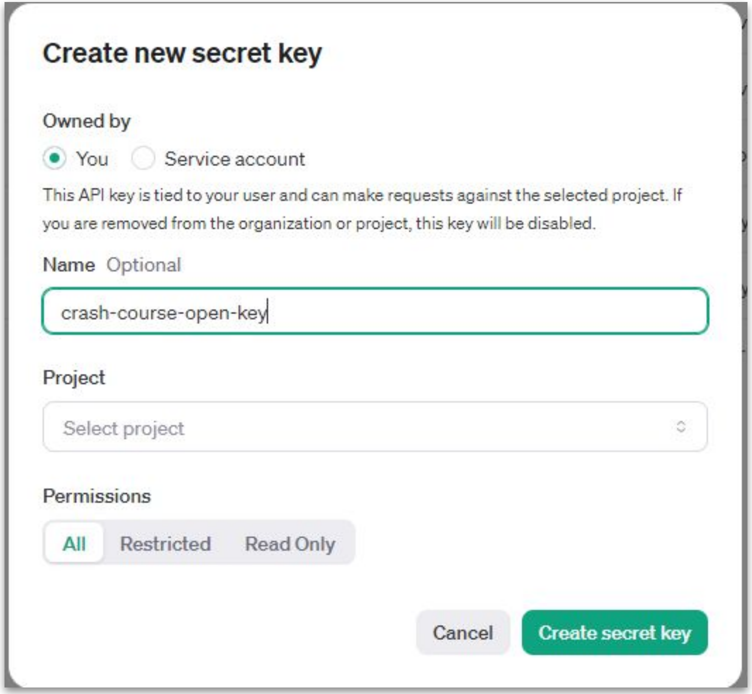
deepseek

Llama 3.1

Hugging Face

Grok

Get API Access: ChatGPT



Create new secret key

Owned by

● You ○ Service account

This API key is tied to your user and can make requests against the selected project. If you are removed from the organization or project, this key will be disabled.

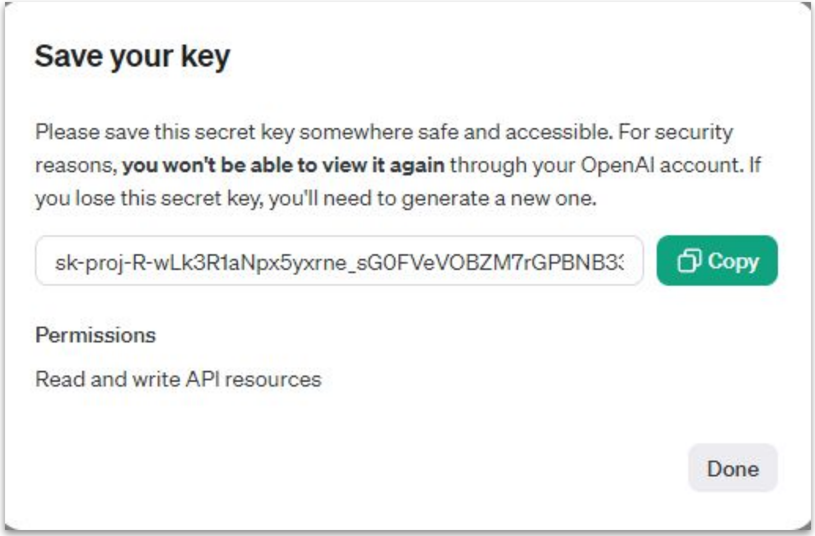
Name Optional

crash-course-open-key

Project

Select project

Permissions



Save your key

Please save this secret key somewhere safe and accessible. For security reasons, **you won't be able to view it again** through your OpenAI account. If you lose this secret key, you'll need to generate a new one.

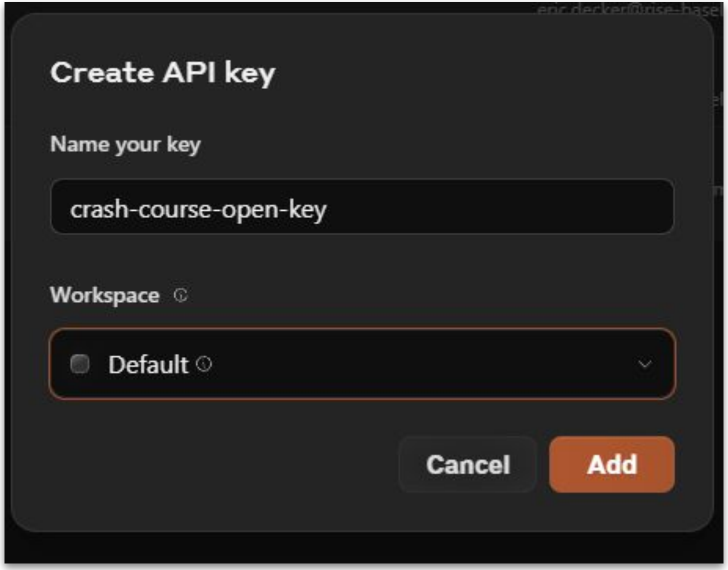
sk-proj-R-wLk3R1aNpx5yxrne_sG0FVeVOBZM7rGPBNB3

Permissions

Read and write API resources

https://platform.openai.com/

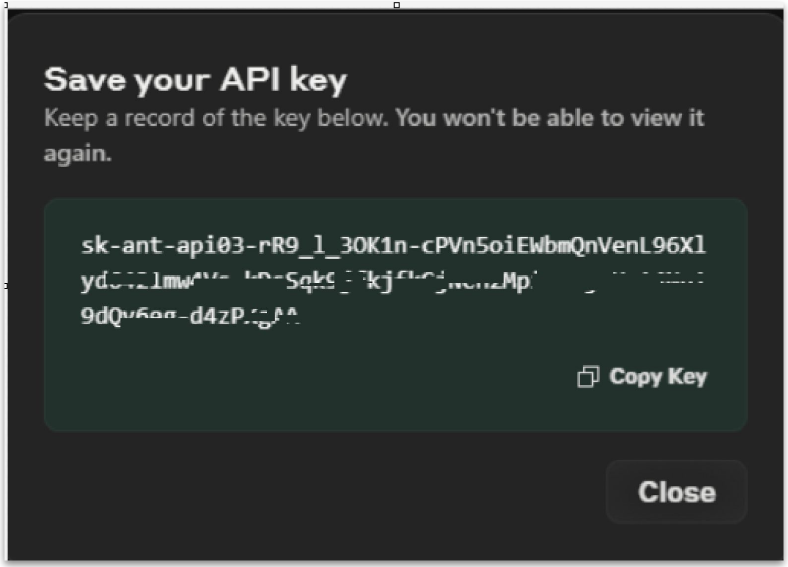
Get API Access: Claude



Create API key

Name your key: crash-course-open-key

Workspace: Default



Save your API key

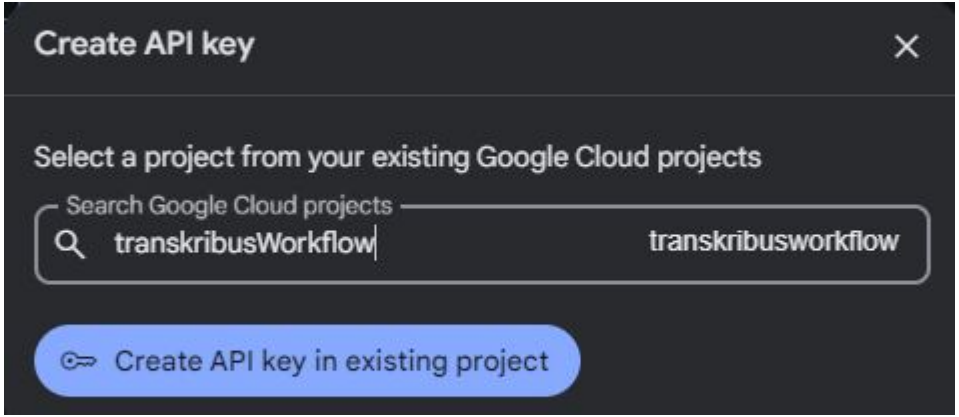
Keep a record of the key below. You won't be able to view it again.

sk-ant-api03-rR9_l_3OK1n-cPVn5oiEWbmQnVenL96Xl [redacted]

https://console.anthropic.com/

Get API Access: Gemini

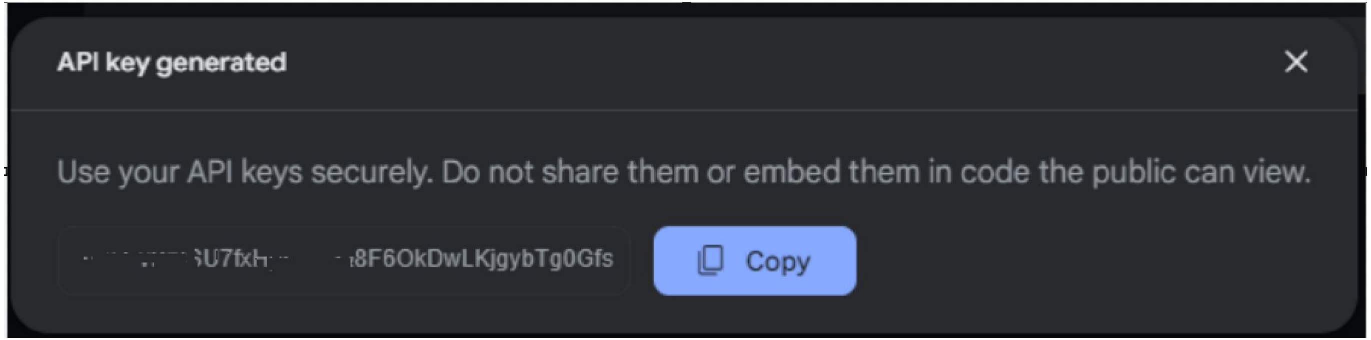
https://aistudio.google.com/app/apikey



Create API key

Select a project from your existing Google Cloud projects

Search Google Cloud projects: transkribusWorkflow



API key generated

Use your API keys securely. Do not share them or embed them in code the public can view.

[redacted]U7fxh[redacted]8F6OkDwLKjgybTg0Gfs

Agenda

1. Prepare
    a. Conceptualize a workflow → What data will be processed how in which steps?
    b. Test your prompt and source material → Does it work in a browser?
    c. Choose a Provider, Get API Access → Which provider is best for your needs?
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to format output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Manage and Calculate Cost

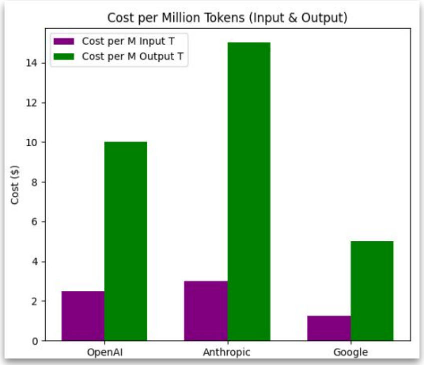
- Costs are calculated based on the input tokens, output tokens and the models used.
- Each provider has its own pricing model; they change often.
- Most providers have general limits (monthly limit, pay as you go, …)
- Tokens?
  - Smaller units of text, that may be words, parts of words, or even punctuation.
  - Rough estimate: 1 token ≈ 4 characters (including spaces; in english language)

Manage and Calculate Cost

- Input Tokens (What is sent to the LLM?)
  - Prompt → 1200 chars / 4 ≈ 300 Tokens
  - Number of Tokens for the image* → (1’024 px * 1’492 px)/750 ≈ 2’037 Tokens
  - Total input tokens → 300 + 2’037 = **2’337 Tokens**
- Output Tokens (What is received from the LLM?)
  - Expected length of output → ~45 lines * 400 chars. ≈ **4’500 Tokens**
- Total Cost:
  - ~ 200 * 2’337 Input Tokens → ~467’400 Tokens
  - ~ 200 * 4’500 Output Tokens → ~900’000 Tokens

*This varies from API to API. Some have a fixed amount. The example is based on Claude AI

rise@unibas.ch



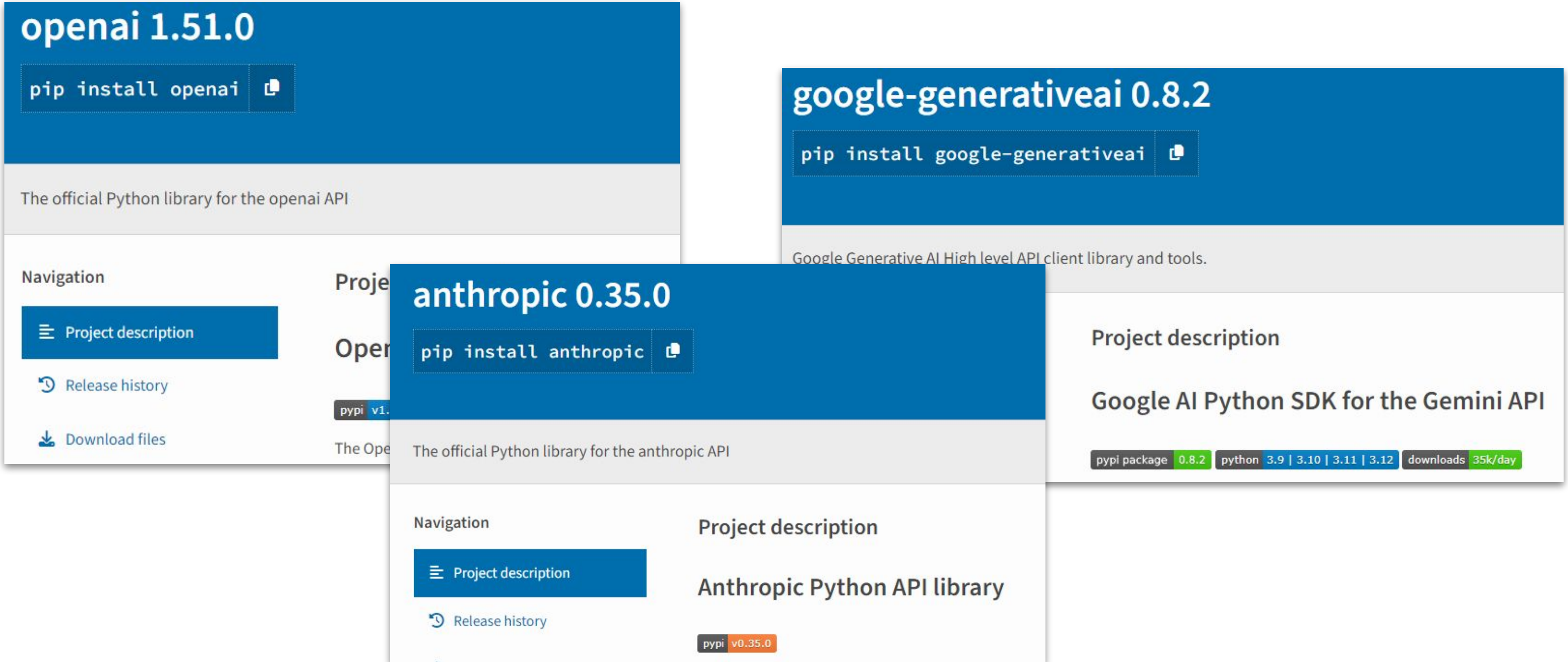
Manage and Calculate Cost

- GPT-4o *https://openai.com/api/pricing/*
  - $2.50 / 1M Input Tokens → ~467’400 Tokens = $ 1.17
  - $10 / 1M Output Tokens → ~900’000 Tokens = $ 9.00 **$10.17**
- 3.7 Sonnet *https://www.anthropic.com/pricing*
  - $3 / 1M Input Tokens → ~467’400 Tokens = $ 1.40
  - $15 / 1M Output Tokens → ~900’000 Tokens = $13.50 **$14.90**
- Gemini 2.0 Flash: *https://ai.google.dev/pricing*
  - $1.25 / 1M Input Tokens → ~467’400 Tokens = $ 0.58
  - $5 / 1M Output Tokens → ~900’000 Tokens = $ 4.50 **$5.09**

Agenda

1. Prepare
    a. Conceptualize a workflow → What data will be processed how in which steps?
    b. Test your prompt and source material → Does it work in a browser?
    c. Choose a Provider, Get API Access → Which provider is best for your needs?
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to format output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Create A Script: Use existing libraries



openai 1.51.0

```
pip install openai
```

The official Python library for the openai API

Navigation

- Project description
- Release history
- Download files

Proje

Oper

pypi v1.

The Ope

google-generativeai 0.8.2

```
pip install google-generativeai
```

Google Generative AI High level API client library and tools.

Project description

Google AI Python SDK for the Gemini API

anthropic 0.35.0

```
pip install anthropic
```

The official Python library for the anthropic API

Navigation

- Project description
- Release history

Project description

Anthropic Python API library

Create A Script: They all work alike

```
api_key = "4-v3ry-l0ng-4nd-s3cr3t-4p1-k3y"
prompt = "[...]"
```

```
from openai import OpenAI

client = OpenAI(api_key=api_key)

workload = [...]

answer = client.chat.completions.\
         create(
            messages=workload,
            model="gpt-4o")




text =
answer.choices[0].message.content
```

```
from anthropic import Anthropic

client = Anthropic(api_key=api_key)

answer =  client.messages.create(
            max_tokens=2048,
            messages=[{
              "role": "user",
              "content": prompt,
            }],
            model="claude-3-opus")



text = answer.content[0].text
```

```
import google.generativeai as genai

genai.configure(api_key=api_key)
model = genai.GenerativeModel\
                (“gemini-flash1.5”)

answer =  model.generate_content\
                         ([prompt])







text = answer.text
```

Create A Script

```python
from openai import OpenAI
from variables import prompt, base64_image

client = OpenAI(api_key="sk-proj-R-wLk...xyz")

workload = [
    {
      "role": "user",
      "content": [
        {"type": "text", "text": prompt},
        {"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{base64_image}"}}
      ]
    },
    {
      "role": "system",
      "content": "You are a precise list-reading machine and your answers are plain JSON."
    }
  ]

answer = client.chat.completions.create(messages=workload,
                                        model="gpt-4o",
                                        temperature=0.5)
```

Add additional data to your prompt

- Add metadata such as the page number or a page id.

```
page_number = 1
for root, _, filenames in os.walk("image_data"):
    for filename in filenames:
        if filename.endswith(".jpg"):
            image_id = filename.split(".")[0]
            prompt = "[...]. The id of the image is {image_id} and the page number is {page_number}."

            # Do the AI request with the adjusted prompt.
```

Agenda

1. Prepare
   a. Conceptualize a workflow → What data will be processed how in which steps?
   b. Test your prompt and source material → Does it work in a browser?
   c. Choose a Provider, Get API Access → Which provider is best for your needs?
   d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
   a. Write requests and pass context → How to write a simple python script which does X requests?
   b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
   a. Output formatting → How do I tell the LLM to format output data?
   b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

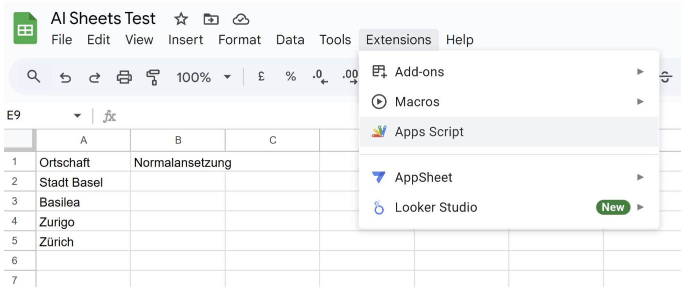
Extended Options / Features

These features differ from API to API

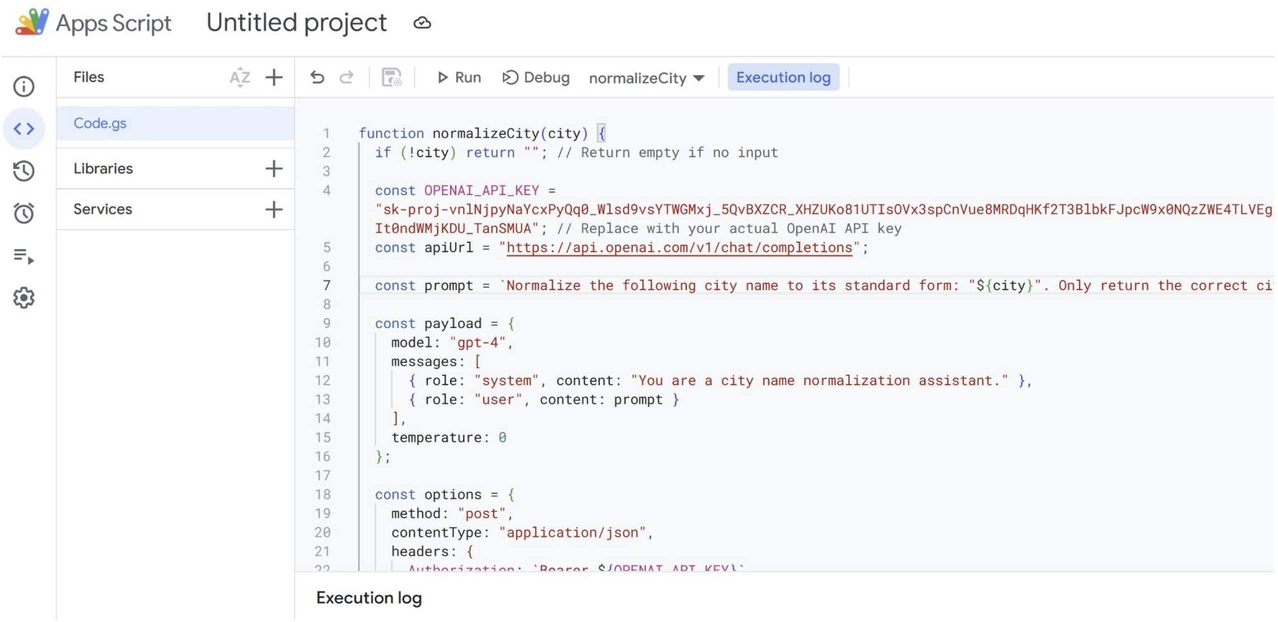
- Maximum Tokens, Temperature, Frequency Penalty, Presence Penalty
- Streaming results, Delaying results
- Additional media types
  - Audio
  - Other file types (PDF, …)

rise@unibas.ch

Prompt from Google Sheets



	A	B	C
1	Ortschaft	Normalansetzung	
2	Stadt Basel		
3	Basilea		
4	Zurigo		
5	Zürich		
6			
7			



```
function normalizeCity(city) {
  if (!city) return ""; // Return empty if no input

  const OPENAI_API_KEY =
  "sk-proj-vnlNjpyNaYcxPyQq0_Wlsd9vsYTWGMxj_5QvBXZCR_XHZUKo81UTIsOVx3spCnVue8MRDqHKf2T3BlbkFJpcW9x0NQzZWE4TLVEg
  It0ndWMjKDU_TanSMUA"; // Replace with your actual OpenAI API key
  const apiUrl = "https://api.openai.com/v1/chat/completions";

  const prompt = `Normalize the following city name to its standard form: "${city}". Only return the correct ci

  const payload = {
    model: "gpt-4",
    messages: [
      { role: "system", content: "You are a city name normalization assistant." },
      { role: "user", content: prompt }
    ],
    temperature: 0
  };

  const options = {
    method: "post",
    contentType: "application/json",
    headers: {
      Authorization: `Bearer ${OPENAI_API_KEY}`
```

Execution log

Prompt from Google Sheets

Agenda

1. Prepare
    a. Conceptualize a workflow → What data will be processed how in which steps?
    b. Test your prompt and source material → Does it work in a browser?
    c. Choose a Provider, Get API Access → Which provider is best for your needs?
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to format output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Output Formatting

```
from pydantic import BaseModel
from openai import OpenAI

client = OpenAI()

class ResearchPaperExtraction(BaseModel):
    title: str
    authors: list[str]
    abstract: str
    keywords: list[str]

completion = client.beta.chat.completions.parse(
    model="gpt-4o-2024-08-06",
    messages=[
        {"role": "system", "content": "You are an expert at structured data extraction. You will be given
unstructured text from a research paper and should convert it into the given structure}.,"
        {"role": "user", "content": "..."}
    ],
    response_format=ResearchPaperExtraction,
)

research_paper = completion.choices[0].message.parsed
```

Agenda

1. Prepare
    a. Conceptualize a workflow → What data will be processed how in which steps?
    b. Test your prompt and source material → Does it work in a browser?
    c. Choose a Provider, Get API Access → Which provider is best for your needs?
    d. Manage/Calculate Cost → How can you limit and control costs?
2. Use the APIs
    a. Write requests and pass context → How to write a simple python script which does X requests?
    b. Extended options/features → How do I set token limits, temperature, etc.?
3. Save output for further use
    a. Output formatting → How do I tell the LLM to format output data?
    b. Saving to data structures → How to save the LLMs outputs to files?
4. Discussion

Saving Data

```
json_string = answer_string.split("```json")[1].split("```")[0]
with open(f"page_{page_number}.json", "w") as f:
    f.write(json_string)
```

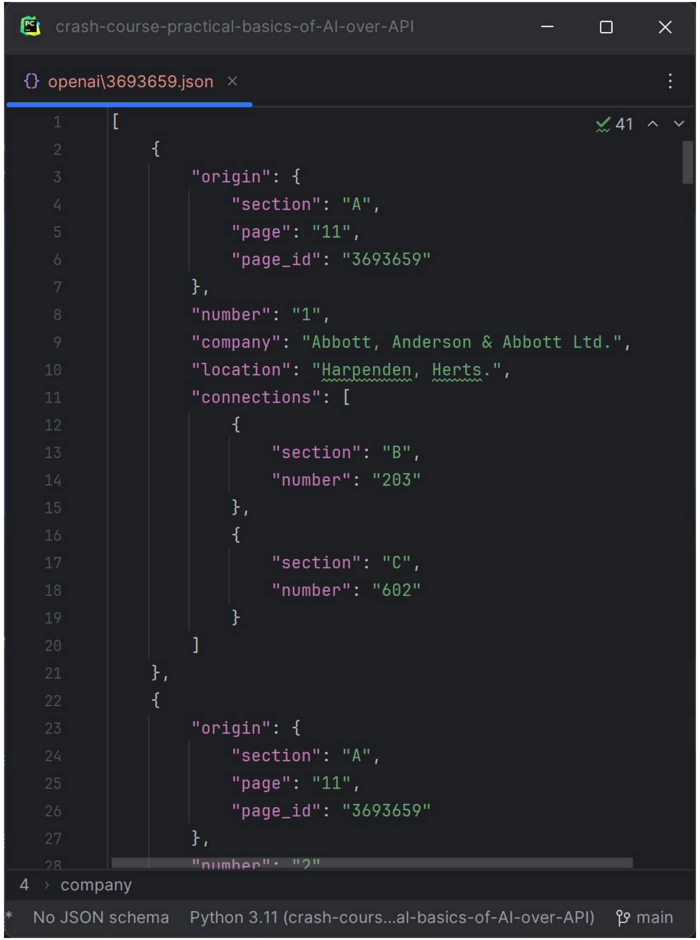
- Save as json files
- Name the files appropriately

Putting it all together

For each image in a directory:

- Resize the image and create a string representation from it
- Create a prompt containing the image ID.
- Formulate an API request with the prompt and image-string and send it.
- Extract the JSON from the answer and save it to a file.

→ https://github.com/RISE-UNIBAS/crash-course-practical-basics-of-AI-over-API

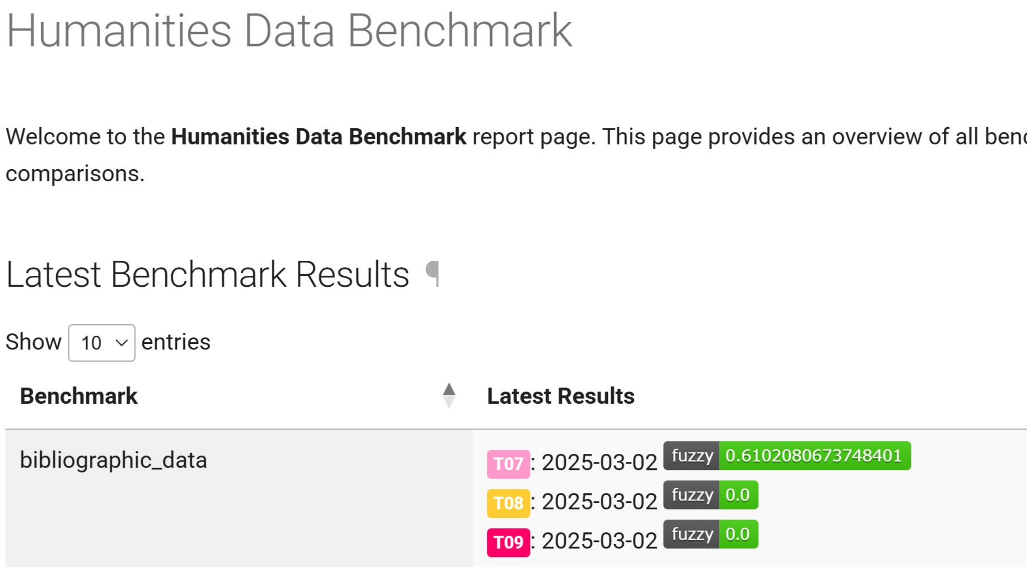

crash-course-practical-basics-of-AI-over-API
openai\3693659.json
41
4 › company
No JSON schema
Python 3.11 (crash-cours...al-basics-of-AI-over-API)
main

```json
[
    {
        "origin": {
            "section": "A",
            "page": "11",
            "page_id": "3693659"
        },
        "number": "1",
        "company": "Abbott, Anderson & Abbott Ltd.",
        "location": "Harpenden, Herts.",
        "connections": [
            {
                "section": "B",
                "number": "203"
            },
            {
                "section": "C",
                "number": "602"
            }
        ]
    },
    {
        "origin": {
            "section": "A",
            "page": "11",
            "page_id": "3693659"
        },
        "number": "2"
```

Further Steps

In the example workflow:

- Run sample
- Quality Control
- Run full batch



Humanities Data Benchmark

Welcome to the **Humanities Data Benchmark** report page. This page provides an overview of all benc comparisons.

Latest Benchmark Results ¶

Show 10 entries

Benchmark	Latest Results
bibliographic_data	T07: 2025-03-02 fuzzy 0.6102080673748401 T08: 2025-03-02 fuzzy 0.0 T09: 2025-03-02 fuzzy 0.0

If You want to use it:

- Get yourself an API key for one of the providers
- Go to The GitHub repository
- Download the code and follow the instructions

Conclusions

- Creating a precise workflow is crucial.
- Test your prompts and source materials to choose the best provider.
- The LLMs provide programming libraries. Use them.
- Adapt your prompts to include metadata of the presented source image.

- With the materials that come with this presentation, you can recreate the example or create your own workflow.
- Be aware that all prices and ratios mentioned in this presentation are changing constantly.

University
of Basel

Thank you for Listening!

Questions?

University
of Basel

More crash courses and workshops about AI:

https://rise.unibas.ch/en/news-events/

19.03.2025	14:00-16:00	AI Benchmarking
26.03.2025	14:00-16:00	Information Extraction from Images
02.04.2025	14:00-16:00	Coding with AI
09.04.2025	13.00–17.00	Workshop: “Bring Your Own Data”