

Automated Data Collection

Collecting data with Application Programming
Interfaces (APIs)

Jakob Jünger & Chantal Gärtner



Data Collection with APIs

User Interface (Browser)



HTML (Webscraping)

```

1 <!DOCTYPE html>
2 <html lang="de" class="no-js logged-in client-root">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=
6
7     <title>
8 Al Jazeera English (@aljazeeraenglish) • Instagram-Foto
9 </title>
10
11
12     <meta name="robots" content="noimageindex, noar
13     <meta name="apple-mobile-web-app-status-bar-sty
14     <meta name="mobile-web-app-capable" content="ye
15     <meta name="theme-color" content="#ffffff">
16     <meta id="viewport" name="viewport" content="wi
17     <link rel="manifest" href="/data/manifest.json"
18
19     <link rel="preload" href="/static/bundles/es6/C
20     <link rel="preload" href="/static/bundles/es6/Consumer.
21     <link rel="preload" href="/static/bundles/es6/ProfilePa
22     <link rel="preload" href="/static/bundles/es6/Vendor.js
23     <link rel="preload" href="/static/bundles/es6/de_DE.js/
24     <link rel="preload" href="/static/bundles/es6/ConsumerL
25     <link rel="preload" href="/static/bundles/es6/ConsumerL

```

JSON (API)

```

{
  "graphql": {
    "user": {
      "biography": "Your Voice. Your Story. Your Platform.",
      "blocked_by_viewer": false,
      "restricted_by_viewer": false,
      "country_block": false,
      "external_url": "https://linkin.bio/aljazeeraenglish",
      "external_url_linkshimmed": "https://l.instagram.com/?u=https%3A%2F%2Flinkin.bio%2Faljazeeraenglish%2Fh&e=ATOSdXQPc_AJxLYM-XyDhXUkd-6URdU6A7Tdb1_vKIKr4JFgrbVRp_pYSC1a5TJ00ekDes4cG-xdKCApES5gquNECq-nuOUSVKGegA&s=1",
      "edge_followed_by": {
        "count": 2409554
      },
      "fbid": "17841400896010580",
      "followed_by_viewer": false,
    }
  }
}

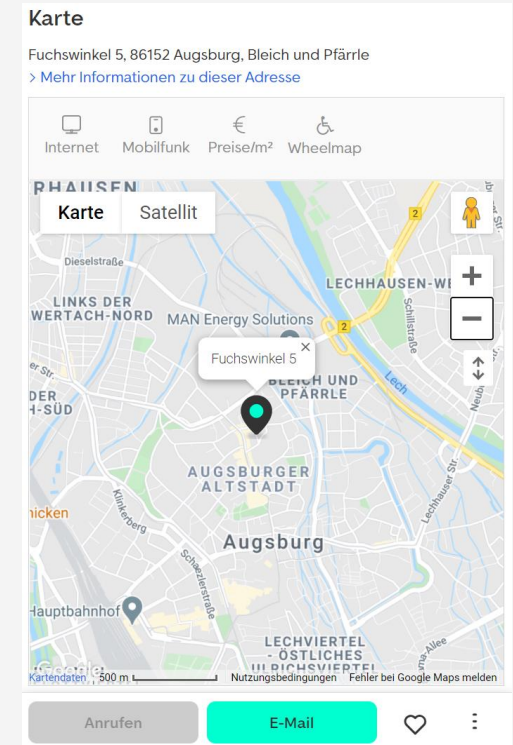
```

Application Programming Interfaces (APIs)

- APIs allow software programs to interact with each other -> Buttons, maps etc. (Jacobson, Brail & Woods 2012: 5).
- Web-APIs are provided by website operators
- Use case for science: pre-structured data access (usually JSON-format)
- Access restrictions (authentication, rate limits)



Source: aljazeera.com



Source: immobilien Scout24.de

Example services and platforms providing APIs

Social Media

- Facebook (<https://developers.facebook.com/>)
- Twitter (<https://developer.twitter.com/en/docs>)
- YouTube (<https://developers.google.com/youtube/v3>)
- Flickr (<https://www.flickr.com/services/api/>)
- Reddit (<https://www.reddit.com/dev/api/>)
- LinkedIn (<https://www.linkedin.com/developers/>)

Messenger: Telegram, WhatsApp, Threema, Skype, Discord

Streaming: Spotify, Apple Music, Vimeo, Twitch

Other Services: Google Maps, Amazon, Wikipedia

Facebook page of Al Jazeera in the browser and as JSON



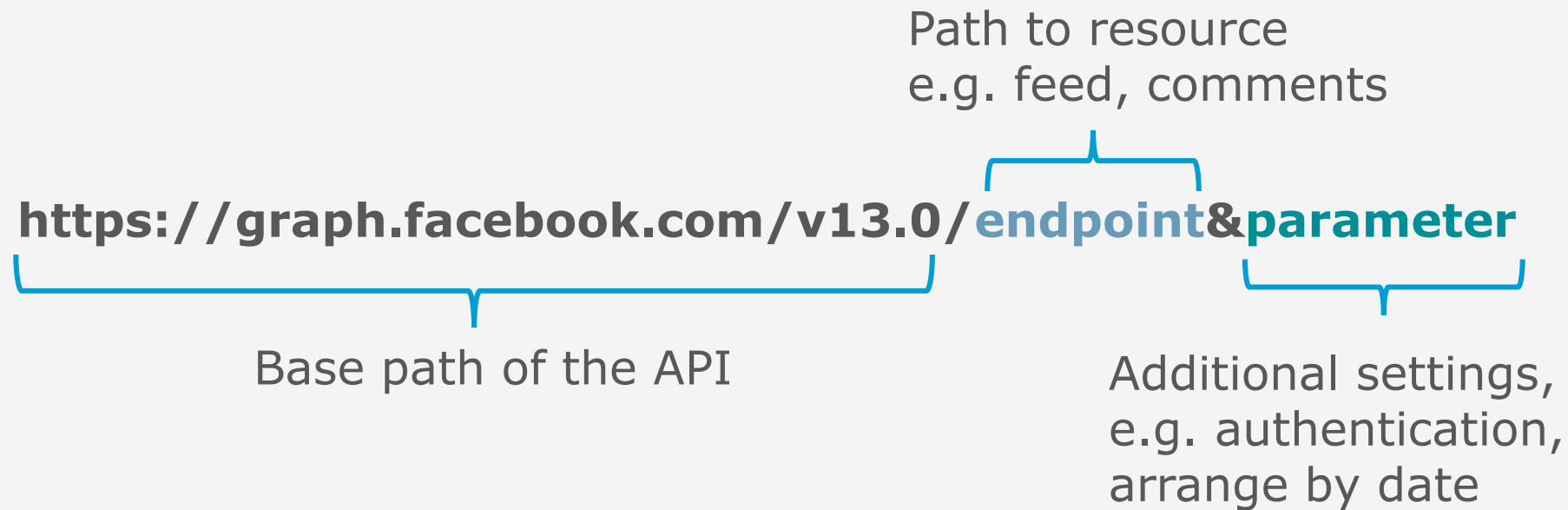
```
{  
  "message": "Accounting firm Mazars said it could  
              no longer stand behind annual  
              financial statements it prepared for  
              the Trump Organization.",  
  "from": {  
    "name": "Al Jazeera English",  
    "id": "7382473689"  
  },  
  "created_time": "2022-02-15T12:19:44+0000",  
  "updated_time": "2022-02-15T15:57:07+0000",  
  "id": "7382473689_10160599026218690"  
}
```

From JSON to tabular data

```
"data": [  
  {  
    "message": "Are things really this bad with Trump? Yikes",  
    "created_time": "2022-02-15T12:21:09+0000",  
    "comment_count": "18",  
    "like_count": "0"  
  },  
  {  
    "message": "I thought the signature does.",  
    "created_time": "2022-02-15T12:22:55+0000",  
    "comment_count": "0",  
    "like_count": "0"  
  },  
  {  
    "message": "He needs a bailout again",  
    "created_time": "2022-02-15T13:15:17+0000",  
    "comment_count": "0",  
    "like_count": "0"  
  }  
]
```

Message	Comment Count	Like Count
Are things really this bad with Trump? Yikes	18	0
I thought the signature does.	0	0
He needs a bailout again	0	0

Using the Facebook API – URL construction



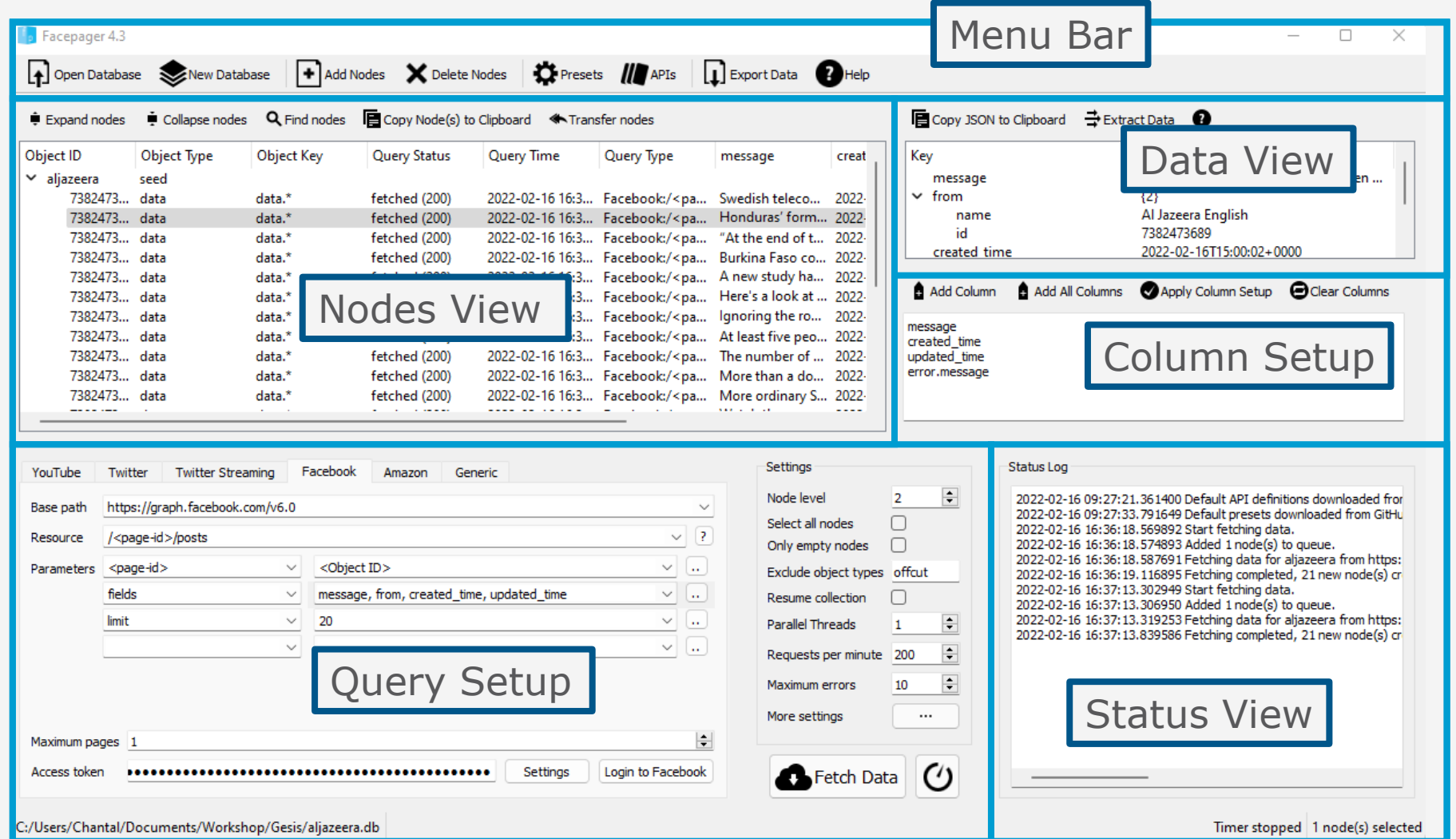
Automation --> compose and query many URLs at the same time



Automated data collection with Facepager

- **Automated data collection** using APIs and web scraping without programming
- **Developed** since 2012 by Till Keyling (Munich) and Jakob Jünger (Münster)
- **Citations:** about 280 published papers
- **Open Source Project** on GitHub with installers for Windows and Mac:
<https://github.com/strohne/Facepager>
- **API Modules:** YouTube, Twitter, Facebook, Amazon, Generic. Query parameters can be customized
- **Presets** for documentation, sharing settings, and getting started quickly
- **Export** as CSV file or via clipboard
- **Help** via built-in API documentation, wiki and Facebook group

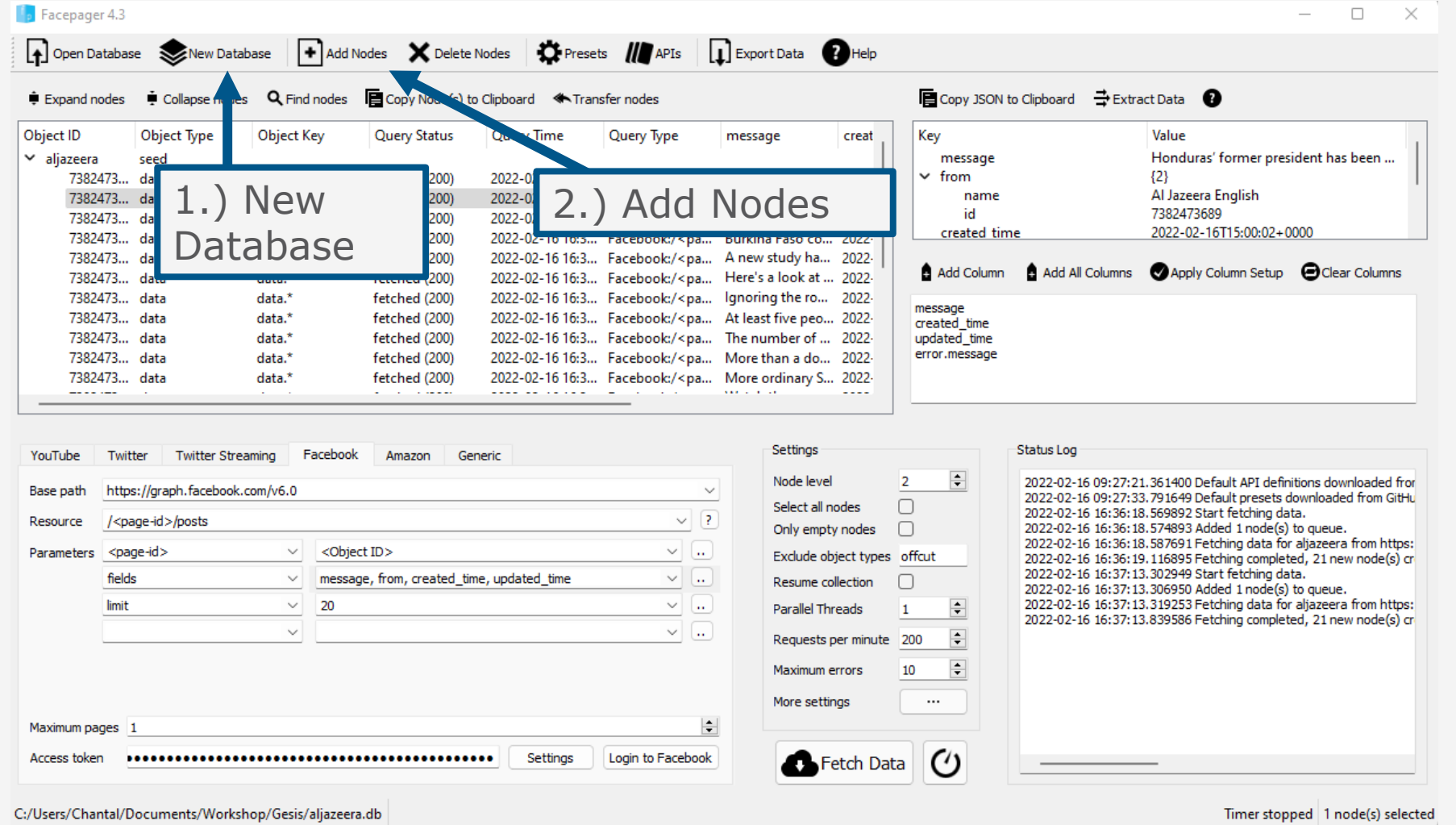
Facepager's Layout



The screenshot shows the Facepager 4.3 application interface. The layout is divided into several sections:

- Menu Bar:** Located at the top right, containing standard window controls (minimize, maximize, close) and the application title "Facepager 4.3".
- Nodes View:** A table displaying a list of nodes. The table has columns: Object ID, Object Type, Object Key, Query Status, Query Time, Query Type, message, and creat. The data is filtered by "alazeera" and shows a "seed" node and several "data" nodes.
- Data View:** A panel on the right showing the details of a selected node. It includes a "Key" section with a tree view (message, from, name, id, created time) and a "Value" section showing the corresponding data.
- Column Setup:** A panel on the right showing a list of columns (message, created_time, updated_time, error.message) and buttons for "Add Column", "Add All Columns", "Apply Column Setup", and "Clear Columns".
- Query Setup:** A panel at the bottom left for configuring the data source and query. It includes tabs for "YouTube", "Twitter", "Twitter Streaming", "Facebook", "Amazon", and "Generic". The "Facebook" tab is selected, showing fields for "Base path", "Resource", "Parameters", "fields", "limit", and "Maximum pages".
- Status View:** A panel at the bottom right showing a "Status Log" with a list of events and a "Timer stopped" indicator.
- Settings:** A panel on the right side of the bottom section containing various configuration options like "Node level", "Select all nodes", "Only empty nodes", "Exclude object types", "Resume collection", "Parallel Threads", "Requests per minute", "Maximum errors", and "More settings".
- Fetch Data:** A button at the bottom right of the settings panel.
- Access token:** A field at the bottom left of the query setup panel for entering a Facebook access token.
- Buttons:** "Settings" and "Login to Facebook" buttons are located at the bottom of the query setup panel.

Facepager's Layout

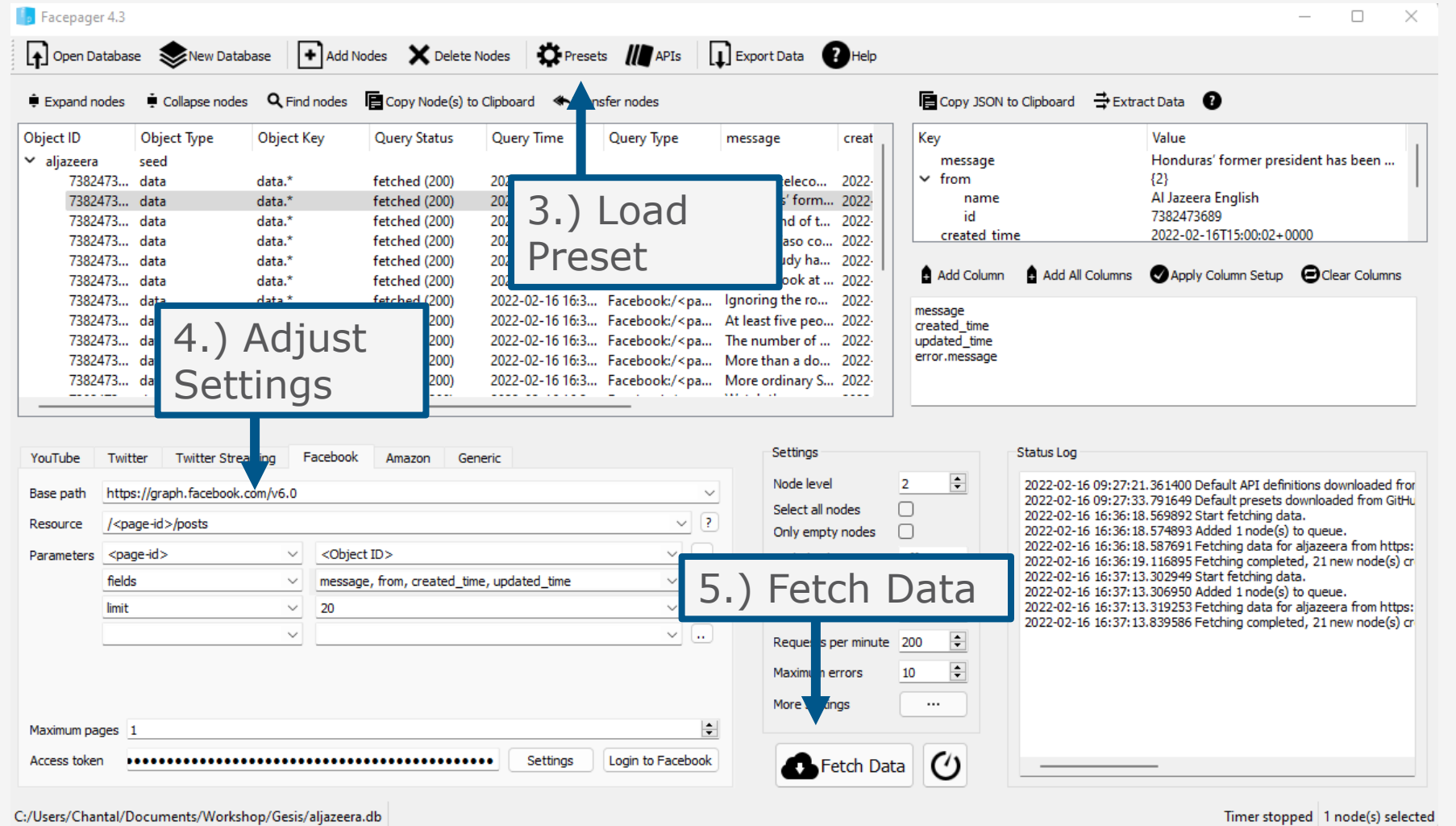


The screenshot shows the Facepager 4.3 application window. Two blue arrows point to specific buttons in the top toolbar:

- 1.) New Database**: Points to the 'New Database' button (represented by a database icon).
- 2.) Add Nodes**: Points to the 'Add Nodes' button (represented by a plus icon).

The main interface displays a table of fetched data for the 'aljazeera' database. The table has columns: Object ID, Object Type, Object Key, Query Status, Query Time, Query Type, message, and created_time. Below the table, there are settings for the data source (YouTube, Twitter, Twitter Streaming, Facebook, Amazon, Generic) and various parameters like Base path, Resource, Parameters, fields, limit, and Maximum pages. A 'Fetch Data' button is visible at the bottom right. The Status Log on the right shows the progress of the data fetching process.

Facepager's Layout



The screenshot shows the Facepager 4.3 application interface. The main window displays a table of data with columns: Object ID, Object Type, Object Key, Query Status, Query Time, Query Type, message, and creat. The data is organized into a tree structure under 'aljazeera'. A blue box labeled '3.) Load Preset' points to the 'Presets' button in the top toolbar. Another blue box labeled '4.) Adjust Settings' points to the 'Settings' button in the bottom toolbar. A third blue box labeled '5.) Fetch Data' points to the 'Fetch Data' button in the bottom toolbar. The interface also includes a 'Status Log' on the right side showing a timeline of events.

3.) Load Preset

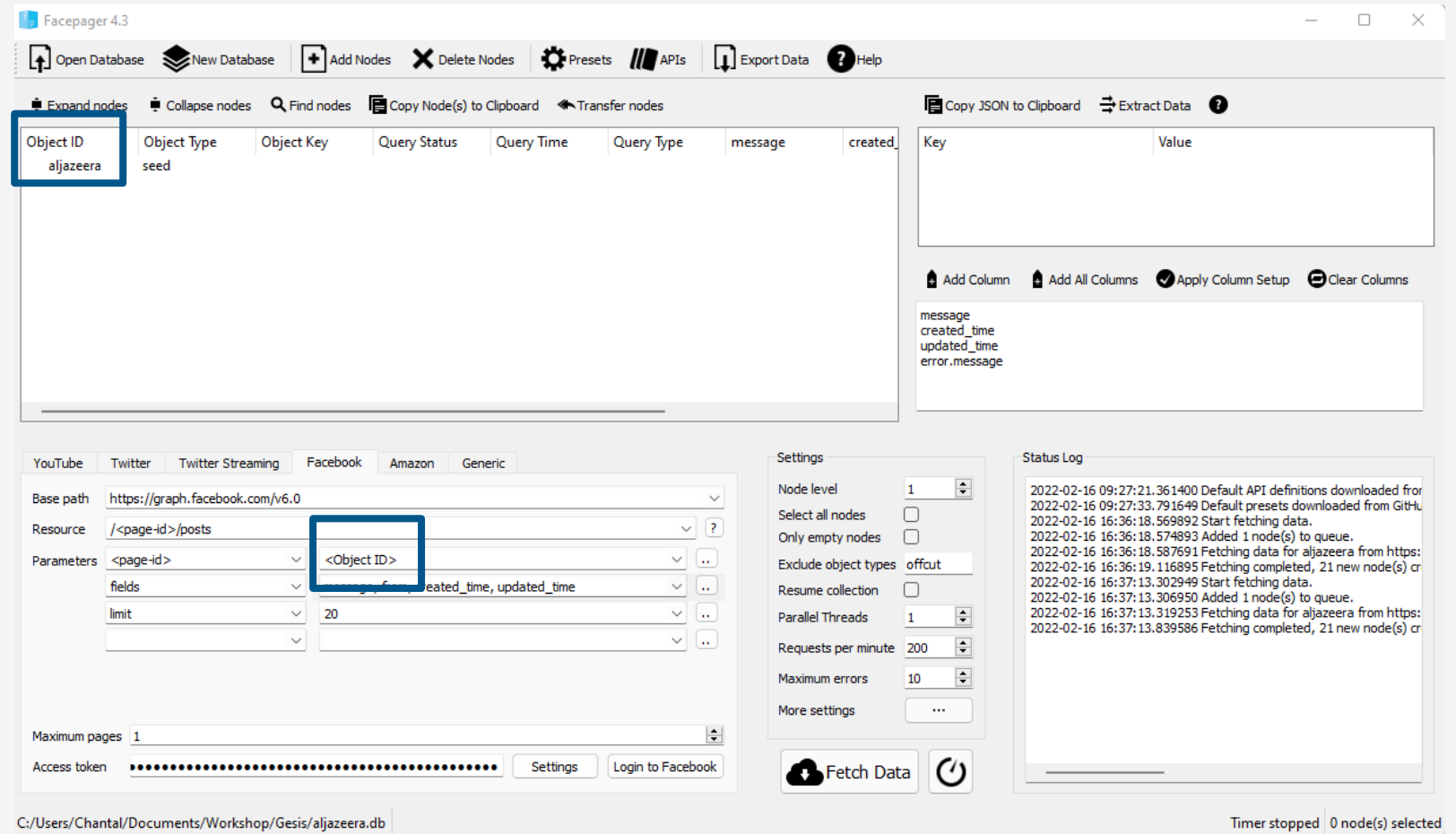
4.) Adjust Settings

5.) Fetch Data

Timer stopped | 1 node(s) selected

Facepager's Layout

<https://graph.facebook.com/v13.0/aljazeera/posts>



Facepager 4.3

Open Database New Database Add Nodes Delete Nodes Presets APIs Export Data Help

Expand nodes Collapse nodes Find nodes Copy Node(s) to Clipboard Transfer nodes

Object ID	Object Type	Object Key	Query Status	Query Time	Query Type	message	created
aljazeera	seed						

Copy JSON to Clipboard Extract Data

Add Column Add All Columns Apply Column Setup Clear Columns

message
created_time
updated_time
error.message

YouTube Twitter Twitter Streaming Facebook Amazon Generic

Base path

Resource

Parameters

Parameter	Value
<page-id>	<Object ID>
fields	message, created_time, updated_time
limit	20

Maximum pages

Access token

Settings

Node level

Select all nodes ☐

Only empty nodes ☐

Exclude object types

Resume collection ☐

Parallel Threads

Requests per minute

Maximum errors

More settings

Fetch Data

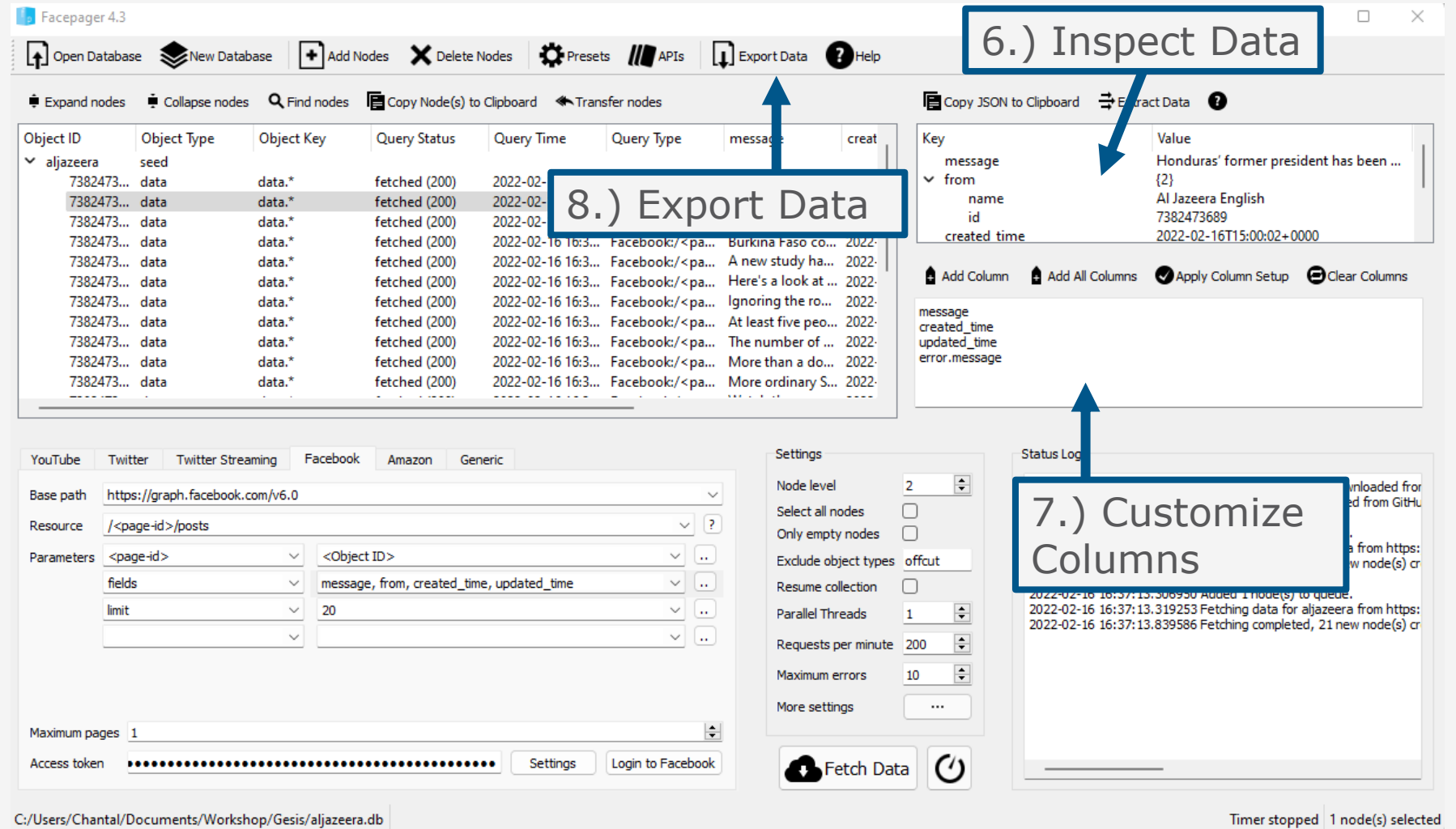
Status Log

2022-02-16 09:27:21.361400 Default API definitions downloaded from
2022-02-16 09:27:33.791649 Default presets downloaded from GitHub
2022-02-16 16:36:18.569892 Start fetching data.
2022-02-16 16:36:18.574893 Added 1 node(s) to queue.
2022-02-16 16:36:18.587691 Fetching data for aljazeera from https:
2022-02-16 16:36:19.116895 Fetching completed, 21 new node(s) cr
2022-02-16 16:37:13.302949 Start fetching data.
2022-02-16 16:37:13.306950 Added 1 node(s) to queue.
2022-02-16 16:37:13.319253 Fetching data for aljazeera from https:
2022-02-16 16:37:13.839586 Fetching completed, 21 new node(s) cr

C:/Users/Chantal/Documents/Workshop/Gesis/aljazeera.db

Timer stopped 0 node(s) selected

Facepager's layout



The screenshot shows the Facepager 4.3 application window. The interface is divided into several sections:

- Top Bar:** Contains buttons for 'Open Database', 'New Database', 'Add Nodes', 'Delete Nodes', 'Presets', 'APIs', 'Export Data', and 'Help'.
- Left Panel:** A tree view showing the database structure. Under 'aljazeera', there is a 'seed' node and a list of 'data' nodes. One 'data' node is selected.
- Center Panel:** A table displaying query results. The columns are: Object ID, Object Type, Object Key, Query Status, Query Time, Query Type, message, and creat. The table shows multiple rows of data fetched from Facebook.
- Right Panel:** A 'Key-Value' view showing details for the selected node. It lists keys like 'message', 'from', 'name', 'id', and 'created time' with their corresponding values. Below this is a section for 'message' with fields like 'created_time', 'updated_time', and 'error.message'.
- Bottom Left Panel:** Configuration settings for data sources. It includes tabs for 'YouTube', 'Twitter', 'Twitter Streaming', 'Facebook', 'Amazon', and 'Generic'. The 'Facebook' tab is active, showing fields for 'Base path', 'Resource', 'Parameters', 'fields', 'limit', and 'Maximum pages'. There is also an 'Access token' field and buttons for 'Settings' and 'Login to Facebook'.
- Bottom Right Panel:** A 'Settings' panel with options for 'Node level', 'Select all nodes', 'Only empty nodes', 'Exclude object types', 'Resume collection', 'Parallel Threads', 'Requests per minute', 'Maximum errors', and 'More settings'. Below these are 'Fetch Data' and 'Refresh' buttons.
- Status Log:** A panel on the far right showing a log of activities, including 'Added 1 node(s) to queue' and 'Fetching completed, 21 new node(s) cr'.

Annotations with arrows point to specific features:

- 6.) Inspect Data:** Points to the 'Key-Value' view on the right.
- 8.) Export Data:** Points to the 'Export Data' button in the top bar.
- 7.) Customize Columns:** Points to the 'message' section in the right panel, which allows selecting specific fields to display.

At the bottom of the window, the file path 'C:/Users/Chantal/Documents/Workshop/Gesis/aljazeera.db' is shown on the left, and 'Timer stopped | 1 node(s) selected' is shown on the right.

Practical session with Facepager

The Facepager CSV file (shortened)

level	id	parent_id	object_id	object_type	message	created_time
0	107	None	aljazeera	seed		
1	108	107	7382473689_10160611919253690	data	Drone force using ...	2022-02-21T23:30:11+0000
1	109	107	7382473689_10160611872573690	data	More than 14,000 people ...	2022-02-21T23:00:36+0000
1	128	107	aljazeera	offcut		
2	213	108	7382473689_10160611919253690	data	We want war, we want ...	2022-02-21T23:33:16+0000
2	214	108	7382473689_10160611919253690	data	If they are telling the ...	2022-02-21T23:40:25+0000
2	223	108	7382473689_10160611919253690	offcut		

```
import pandas as pd
df = pd.read_csv(
    "facebooknews.csv",
    sep=";"
)

display(df)
```

```
df = df.loc[ : , : ]
```

rows

```
df.object_type == 'data'
```

columns

```
['object_id',
 'created_time', 'comments']
```

Python – our toolbox part one

Data wrangling

```
import pandas as pd
```

```
pd.read_csv
```

```
df.value_counts()
```

```
df.describe()
```

```
df.sort_values()
```

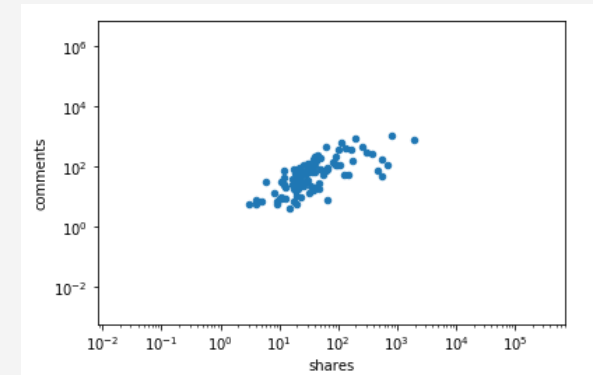
Plots

```
import matplotlib.pyplot as plt
```

```
df.hist()
```

```
plt.hist()
```

```
df.plot.scatter
```



Literature

Jünger, Jakob (2018): **Mapping the Field of Automated Data Collection on the Web**. Data Types, Collection Approaches and their Research Logic. In: Stützer, Cathleen / Welker, Martin / Egger, Marc (Hg). Computational Social Science in the Age of Big Data. Concepts, Methodologies, Tools, and Applications. *Neue Schriften zur Online-Forschung* der Deutschen Gesellschaft für Online-Forschung (DGOF). Köln: Halem-Verlag, S. 104-130.

Jünger, Jakob / Keyling, Till (2020). **Facepager**. An application for automated data collection on the web. Source code and releases available at <https://github.com/strohne/Facepager>.

Russell, A. Matthew (2011): **Mining the Social Web**: Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites. O'Reilly Media.