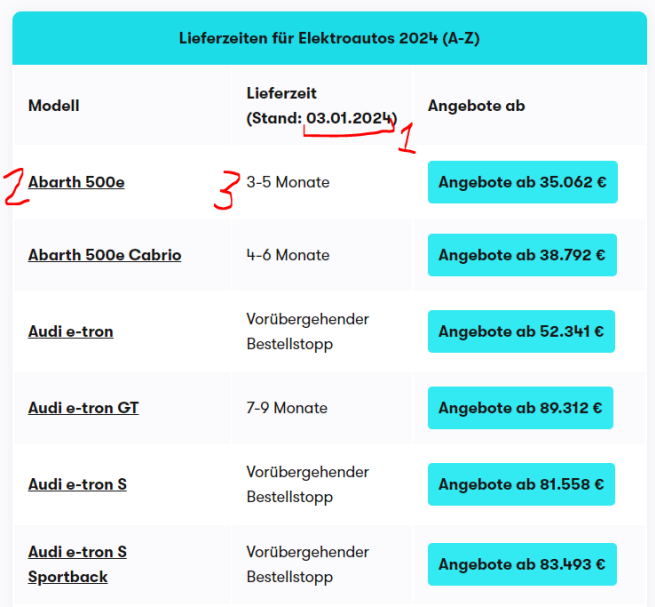
Carwow Electric Vehicle Lead Times – 14811

⚠️ **If the resource you are scraping requires you to agree to any Terms & Conditions, please do not proceed and notify your contract manager immediately. Under no circumstances should you create a false account or fake identity.**

Description:

We would like to scrape the advertised delivery times for electric vehicles as listed here:

https://www.carwow.de/ratgeber/elektroauto/lieferzeiten-elektroautos#gref



Lieferzeiten für Elektroautos 2024 (A-Z)

Modell	Lieferzeit (Stand: 03.01.2024)	Angebote ab
Abarth 500e	3-5 Monate	Angebote ab 35.062 €
Abarth 500e Cabrio	4-6 Monate	Angebote ab 38.792 €
Audi e-tron	Vorübergehender Bestellstopp	Angebote ab 52.341 €
Audi e-tron GT	7-9 Monate	Angebote ab 89.312 €
Audi e-tron S	Vorübergehender Bestellstopp	Angebote ab 81.558 €
Audi e-tron S Sportback	Vorübergehender Bestellstopp	Angebote ab 83.493 €

Simply read the table that shows order time by vehicle model. We need to collect:

- [1] The date on which the lead times were measured / reported
- [2] The model of car
- [3] The lead time

Please collect all strings in the local language of the website (German).

Root URL:

https://www.carwow.de/ratgeber/elektroauto/lieferzeiten-elektroautos#gref

Job Frequency:

Daily

Output Columns:

Column name	Datatype	Xpath/location on site/number in screenshot	Example value	Comment
scrape_datetime	*datetime*	N/A	datetime.datetime.utcnow().isoformat()	*Ensure this is a datetime in ISO-8601 format, and have one value per run, evaluated at start of the script, rather than at the time of each request*
data_url	*string*			*URL where the row is scraped from*
as_of_date	*datetime*	1	2024/01/03	*The date on which the lead times were observed; in the table header; repeat this across all rows*
model	*string*	2	Abarth 500e	*Model of car*
lead_time	string	3	3-5 Monate	*String describing the lead time*

Timeline:

You may complete this job any time and submit any required files to the linked Github repository within one week of accepting the job.

Please submit your code here: Repo to follow in chat

Submission Files:

Sample.csv for sample data

A requirement.txt

scrape/ - containing all of the source code

Main file: scrape.py that will be run with a output $filename.

Job Schema/Output Format:

You should save the output csv using these settings from a pandas DataFrame:

```
encoding="utf-8",
 line_terminator="\n",
 quotechar='"',
 quoting=csv.QUOTE_ALL,
 index=False
```

Package preferences and Scraping notes

Selenium is a last resort (only purposeful obfuscation from website owner)

Generally unnecessary to run a full browser

Please provide reasoning if using Selenium

Beautiful soup

Avoid hard codes, references should be relative

Parse json from an api rather than html

Runtime Environment:

Your code will be copied form the root to/usr/src/scrape

You should feel free to modify the requirements as you need. However, you must keep the awscli dependency

You may also upload additional binaries into the repository root and reference them there.

Please do not change the Dockerfile or shell scripts in the repository as this will cause automated test failure.

python scrape.py $filename

Page access limitations (max requests / day):

If you encounter a captcha during your scrape job, please contact the job poster before continuing.

10% of website traffic max