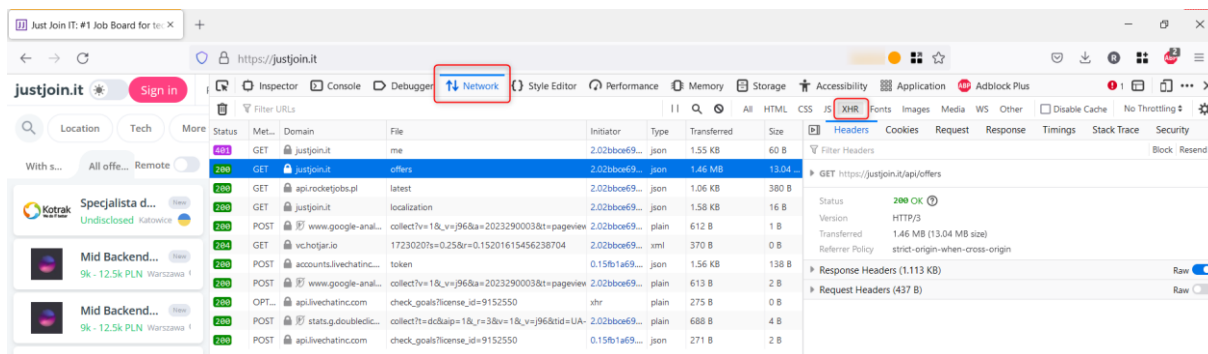


## Jak znaleźć API Endpoint

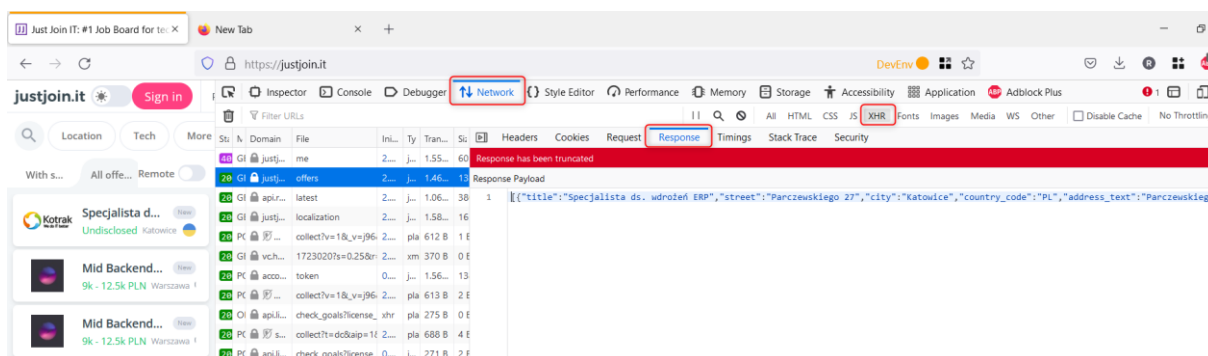
W przeglądarce (screeny pochodzą z Firefox ale w Chrome wygląda to podobnie) odpal *Developer Tools* (np. F12 lub z menu przeglądarki, zazwyczaj w opcji *More Tools*) i wpisz adres interesującej Cię strony. Chcesz zobaczyć requesty jakie są wysyłane i pobierane przez daną stronę. Interesuje nas zakładka *Network* a w niej głównie *XHR* (XMLHttpRequest) – opcje zaznaczone poniżej.



Jak wyłuskać ten właściwy request?

W gąszczu requestów, które pojawią się w zakładce *Network* interesują nas oczywiście tylko te, które pochodzą z oryginalnej strony (zwróć uwagę na kolumnę *Domain*).

W *Headers* zakładce *XHR* sprawdzamy, czy mamy jakiś sensowny URL i czy dany request zwraca jakieś dane ustrukturyzowane:

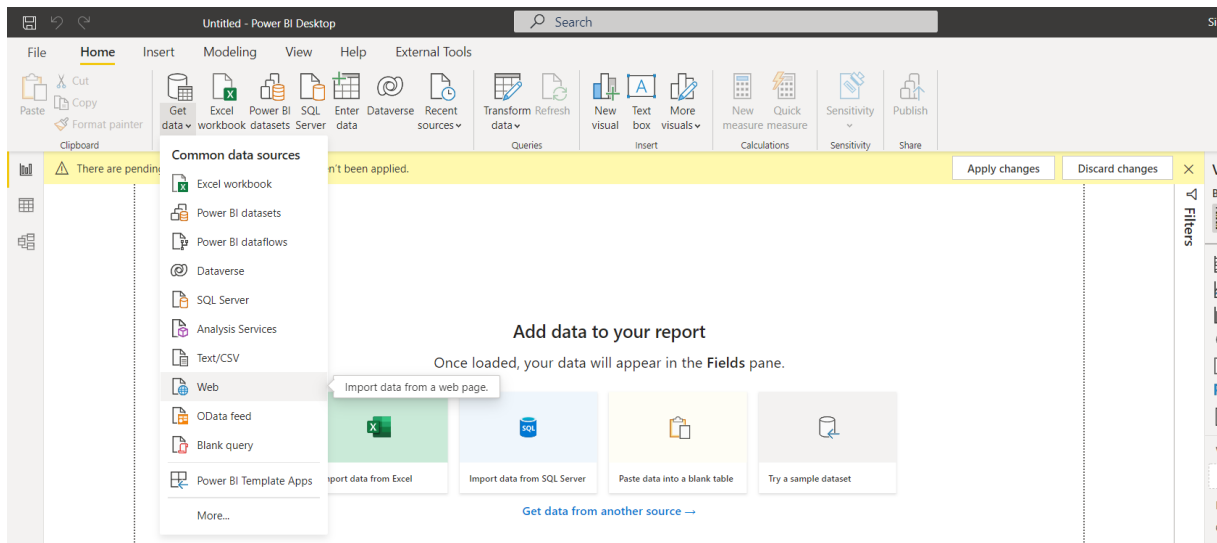


Jeśli tak i te dane wyglądają na to, czego poszukujemy, to kopiujemy URL z zakładki *Headers* i możemy od razu przejść do Power BI.

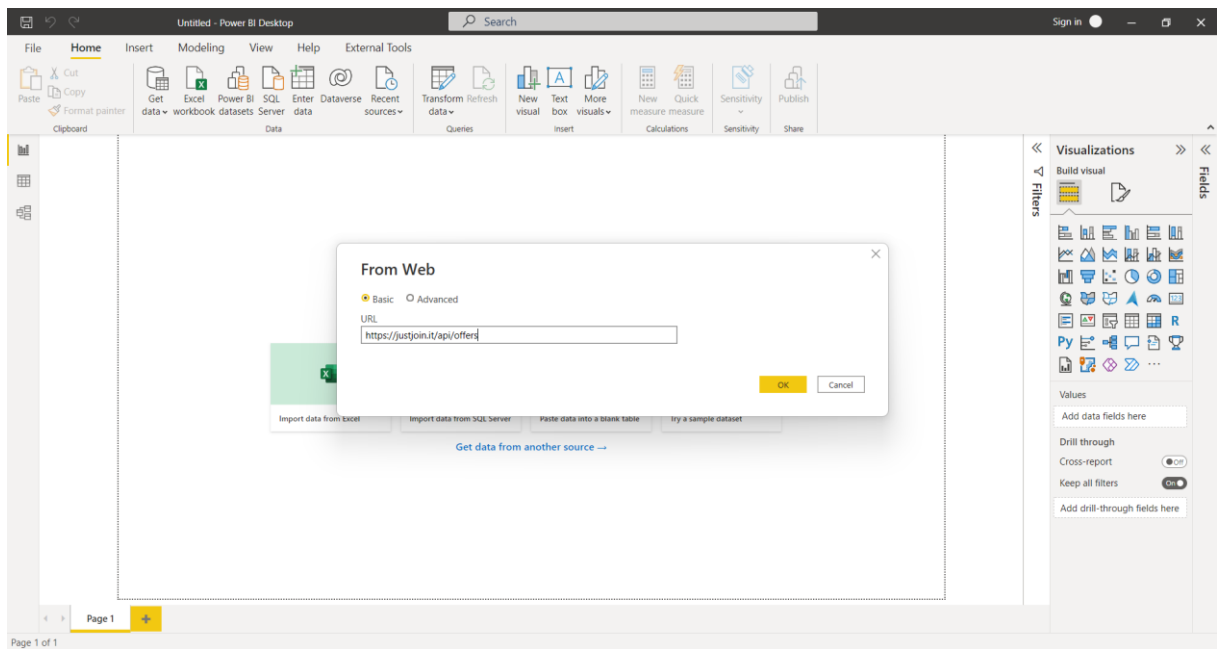
Zazwyczaj trzeba trochę pogrzebać i poprzyglądać się, który request będzie tym, którego potrzebujemy. Jeśli nie widzimy niczego ciekawego, albo nie ma requestów, to czasem warto odświeżyć lub wejść w pojedynczy rekord i na nim się przyglądać jak wygląda URL (możemy potrzebować tylko części albo też dany URL zwraca nam wynik przefiltrowany, a my możemy odpytać o całość).

## Ściąganie danych do Power BI

W Power BI korzystamy z konektora Web:



I po prostu wklejamy nasz URL:



Sam Power BI jest na tyle sprytny, że poradzi sobie z przetwarzaniem danych ustrukturyzowanych z pobranego URLa, więc naszym zadaniem jest głównie sprawdzenie, czy transformacje są takie, jakich chcieliśmy i czy potrzebujemy czegoś więcej (lub mniej).

Table: TransformColumnTypes(#"Expanded Column1",{"title", type text}, {"street", type text}, {"city", type text}, {"country\_code", type text}, {"address\_text", type text})

title	street	city	country_code	address_text
.NET Developer	ul. Zygmunta Vogla 8	Warszawa	PL	ul. Zygmunta Vogla 8, Warszawa
Senior Frontend Developer (React)	Ofiar Oświęcimskich 17	Wrocław	PL	Ofiar Oświęcimskich 17, Wrocław
QA Engineer	Ofiar Oświęcimskich 17	Wrocław	PL	Ofiar Oświęcimskich 17, Wrocław
FullStack Software Developer	Centrum	Kraków	PL	Centrum, Kraków
Mid Embedded Software Engineer	Zamoyskiego 24	Kraków	PL	Zamoyskiego 24, Kraków
Product Designer	Krucza 50	Warszawa	PL	Krucza 50, Warszawa
Product Designer	Królowej Bony 13	Gliwice	PL	Królowej Bony 13, Gliwice
Principal Software Engineer	Karmelicka 27	Kraków	PL	Karmelicka 27, Kraków
UI Software Developer	Lubicz 3	Kraków	PL	Lubicz 3, Kraków
Test Automation Engineer	Kasprzaka 2	Warszawa	PL	Kasprzaka 2, Warszawa
Frontend Developer Angular	Centrum	Warszawa	PL	Centrum, Warszawa
Ruby Developer	Al. Pokoju 18	Kraków	PL	Al. Pokoju 18, Kraków
Tester automatyzujący /Team lead	plac Aleja Roździeńskiego 188H	Katowice	PL	plac Aleja Roździeńskiego 188H, Katowice
Java Developer remote	Krzycka 43/1	Wrocław	PL	Krzycka 43/1, Wrocław
Senior Azure Specialist	Puławska	Warszawa	PL	Puławska, Warszawa
Software Developer	Lubicz 3	Kraków	PL	Lubicz 3, Kraków
Embedded Software Developer	Lubicz 3	Kraków	PL	Lubicz 3, Kraków
Frontend Angular Architect	Puławska 543	Warszawa	PL	Puławska 543, Warszawa
Java Software Architect	Puławska 543	Warszawa	PL	Puławska 543, Warszawa
Inżynier Systemowy Linux	Puławska 543	Warszawa	PL	Puławska 543, Warszawa
Head of IT	Łęborska 38	Gdańsk	PL	Łęborska 38, Gdańsk
Android Software Developer	Al. Zwycięstwa 96/98	Gdynia	PL	Al. Zwycięstwa 96/98, Gdynia
Java Developer	Plac Konstytucji 3 Maja 3	Wrocław	PL	Plac Konstytucji 3 Maja 3, Wrocław
Junior Front-end Developer (Vue)	Plac Konstytucji 3 maja	Wrocław	PL	Plac Konstytucji 3 maja, Wrocław
QA Engineer	Dolina Panny Marii 5	Lublin	PL	Dolina Panny Marii 5, Lublin
Node.js Developer	Świętego Antoniego 2/4	Wrocław	PL	Świętego Antoniego 2/4, Wrocław
Open Source Senior Python (Django) Developer	Bobrzyńskiego 25	Kraków	PL	Bobrzyńskiego 25, Kraków

Polecam przejrzanie po kolei tego co się dzieje w poszczególnych krokach i zwrócenie uwagi na to, że bardzo często pobieramy listy rekordów, a niejednokrotnie interesujące nas kolumny to znowu rekordy lub listy, więc je rozwijając będziemy duplikować wiersze:

Table: TransformColumnTypes(#"Expanded Column1",{"company\_logo\_url", type text}, {"skills", type text}, {"multilocation", type text}, {"way\_of\_apply", type text})

company_logo_url	skills	multilocation	way_of_apply
https://bucket.justjoin.it/offers/company_logos/thumb/7c20ab4d...	List	TRUE	form
https://bucket.justjoin.it/offers/company_logos/thumb/52c8b5933b...	List	TRUE	redirect
https://bucket.justjoin.it/offers/company_logos/thumb/a5f183f6b3c...	List	TRUE	redirect
https://bucket.justjoin.it/offers/company_logos/thumb/b500cad3a0c...	List	TRUE	form
https://bucket.justjoin.it/offers/company_logos/thumb/781db4dcfc1...	List	FALSE	form
https://bucket.justjoin.it/offers/company_logos/thumb/7436de39bb6...	List	TRUE	redirect
https://bucket.justjoin.it/offers/company_logos/thumb/a39ae8e3689...	List	TRUE	redirect
https://bucket.justjoin.it/offers/company_logos/thumb/aas6fedfd051f...	List	TRUE	redirect
https://bucket.justjoin.it/offers/company_logos/thumb/c2e6788791b...	List	TRUE	form

Pamiętaj też, że Power BI lubi czasem zrobić za nas trochę za dużo 😊 więc uważaj na kroki, które dorzuca automatycznie. Zawsze możesz je powyrzucać (i warto się tych nadmiarowych pozbywać), np. tu:

Table: TransformColumnTypes(#"Expanded postings.salary",{"postings.id", type text}, {"postings.name", type text}, {"postings.location.places", type text}, {"postings.location.fullyremote", type text}, {"postings.location.covidTimezone", type text})

postings.id	postings.name	postings.location.places	postings.location.fullyremote	postings.location.covidTimezone
BBKKTIV	Consult Red	List	FALSE	
FKI4FHDT	Magnify Logistics	List	TRUE	
1ABSHIA3	No Fluff Jobs	List	TRUE	
1ABSHIA3	No Fluff Jobs	List	TRUE	
1ABSHIA3	No Fluff Jobs	List	TRUE	
IVCFB8F	Magnify Logistics	List	TRUE	
Y1DWOSKQ	Lerta	List	TRUE	
Y1DWOSKQ	Lerta	List	TRUE	
ATPOGUGZ	Avanade Poland	List	FALSE	
ATPOGUGZ	Avanade Poland	List	FALSE	
2G9KQBVH	Relayr	List	FALSE	
2G9KQBVH	Relayr	List	FALSE	
2G9KQBVH	Relayr	List	FALSE	
2G9KQBVH	Relayr	List	FALSE	
2G9KQBVH	Relayr	List	FALSE	

No to teraz przed Tobą mnóstwo świetnej zabawy 😊 Przeróbki w Power Query, łączenie danych, dorzucanie funkcji, tworzenie prawidłowego modelu itp. itd. (czyli to co tygryski lubią najbardziej).

Przy tworzeniu raportu ograniczają nas już tylko nasza wyobraźnia (no i trochę też ograniczenia techniczne).

Pamiętajmy o tym, że tak pozyskane dane pokazują nam stan „na dzisiaj”, więc gdybyśmy chcieli porównywać je w czasie, to należałoby je też w jakiś sposób odkładać. Do pełni szczęścia brakuje nam więc jakiegoś ekstraktu tych danych (najlepiej w sposób oczywiście automatyczny).

## Ekstrakt danych z Power BI do pliku płaskiego

Kiedy skończysz już przerabianie swoich danych i masz gotowe query wynikowe, możesz pokusić się o eksport tych danych. Do zastanowienia się jest oczywiście kwestia, które dane będą wymagały takiego ekstraktu, czy np. tylko tabela faktów, czy także część słowników. W moim przypadku najprostszym sposobem wydał mi się skrypt R.

Z wymagań, które muszą być spełnione przed tym krokiem, to musisz mieć zainstalowany R (tego kroku nie opisuję, odsyłam do dokumentacji).

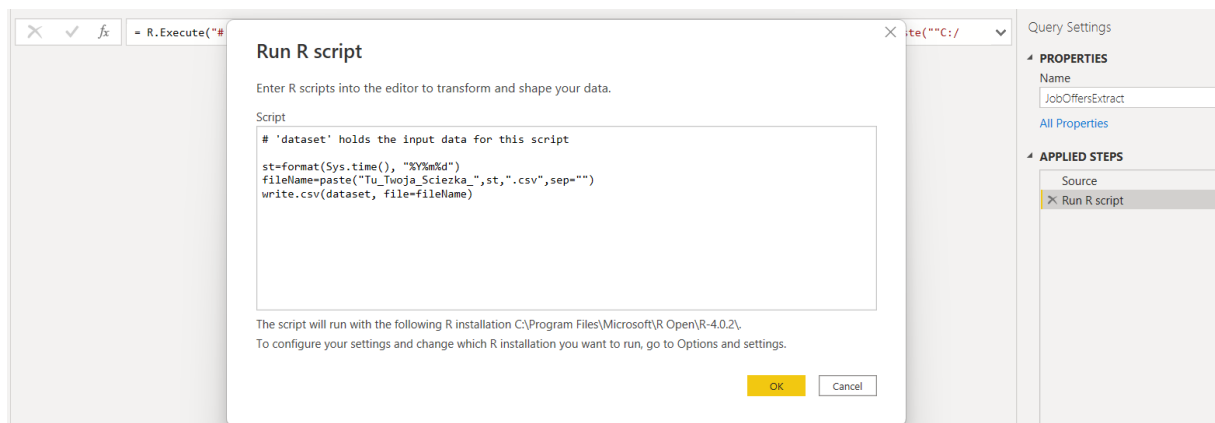
Na gotowym query wynikowym (tym lub tymi, które chcesz eksportować), a dokładniej kopii tego query (używam zarówno widoku bieżącego jak i tych danych do eksportu, więc potrzebuję dwóch query) dorzucamy krok *Run R Script*:



Najprostszy ekstrakt, który zadziała to:

```
write.csv(dataset, file = "Twoja_Sciezka.csv")
```

Ale ponieważ nam zależy na schedulowanym ekstrakcie (czyli danych zrzuconych np. każdego dnia), to zrobimy odrobinę bardziej skomplikowany skrypt:



```
st=format(Sys.time(), "%Y%m%d")
```

```
fileName=paste("Twoja_Sciezka ",st,".csv",sep="")
```

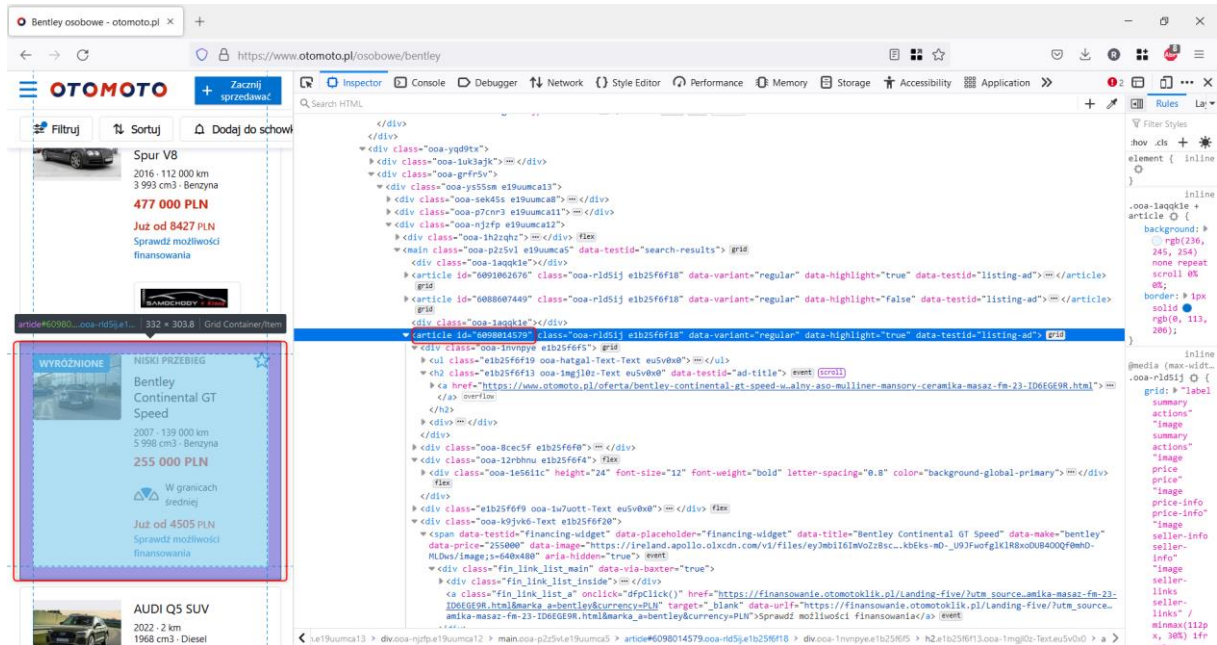
```
write.csv(dataset, file=fileName)
```

I tyle już wystarczy 😊

Kolejnym krokiem, aby działało się to automatycznie jest ustawienie harmonogramu. Tu posługujemy się Data Gatewayem w personal mode (niestety na ten moment działa to z personal mode). Instalacja i harmonogram są już standardowe, więc nie będę się nad tym rozwodzić.

## Wyłuskiwanie danych z tagów

Nie zawsze jednak uda nam się znaleźć adres endpointa. W tym przypadku pozostają nam stare dobre tagi i znajdowanie wzorców pośród nich, a następnie dzielenie poszczególnych sekcji w Power Query. W tym przypadku najprościej skorzystać znowu z Developer Tools (F12) i tym razem z zakładki Inspector. W poniższym przykładzie możemy zauważyć, że np. każda nowa oferta rozpoczyna się od tagu `article`.



Dalsze kroki będą już analogiczne, czyli dużo Power Query, dużo transformacji aż do momentu, gdy jesteśmy zadowoleni z uzyskanego efektu.

## O czym pamiętać

Na koniec jeszcze jedna informacja, na którą należy zwracać uwagę, czyli czy w ogóle możemy scrapować dane. Pamięamy o tym, żeby sprawdzić strony, z których chcemy pobierać dane, nie naruszać praw autorskich i ogólnie przestrzegać regulaminów. Polecam pogooglować bo to kolejny bardzo interesujący temat.