

# AZON API - opis realizacji

6 maja 2020

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Biblioteka</b>	<b>2</b>
2.1	Wyszukiwanie zasobów po ich autorze . . . . .	2
2.2	Wyszukiwanie zasobów po stronach . . . . .	3
2.3	Generowanie metadanych wyszukanych zasobów . . . . .	3
2.4	Inne metody oraz wersjonowanie biblioteki . . . . .	6
2.4.1	Lista języków programowania . . . . .	6
2.4.2	Lista centrów badawczych . . . . .	6
2.4.3	Lista laboratoriów . . . . .	7
2.4.4	Historia wersjonowania . . . . .	7
<b>3</b>	<b>Aplikacja</b>	<b>8</b>
3.1	Menu . . . . .	8
3.2	Przetestowanie wersjonowania (upgrade i downgrade) . . . . .	10

# 1 Wstęp

Biblioteka została utworzona na platformie GitHub ([link](#), gałąź *biblioteka*) w wyniku problemów z e-science'owym GitLab'em. Aplikacja korzystająca z biblioteki znajduje się [tutaj](#), gałąź master. Na GitHub'ie znajduje się kilka wersji biblioteki - dostępne do wglądu [tutaj](#). Opisy jak zrobić upgrade oraz downgrade biblioteki znajdują się w pliku *README.md* na [GitLab'ie](#).

Ogólnie został utworzony jeden pakiet (*package1*) chociaż jest też drugi (*package2*), lecz w którym nic nie ma - nie miałem pomysłu co by tam wrzucić. Wszystko co było wymagane jest w pakiecie *package1*, w tym drugim jest jedynie jakaś prosta funkcja, tylko w celu pokazania, że można korzystać z obu. W module *api\_handler.py* są napisane funkcje pozwalające wyszukiwać zasoby po autorze oraz po stronach, w module *bibtex\_classes.py* klasy odpowiadające poszczególnym typom wpisów w formacie BibTeX, a w module *api\_others.py* są inne funkcje zwracające np. listę języków programowania, centrów naukowych lub laboratoriów.

Przetestowanie wersjonowania znajduje się na samym końcu.

## 2 Biblioteka

### 2.1 Wyszukiwanie zasobów po ich autorze

W celu wyszukiwania zasobów po autorze została napisana klasa Entry, która zawiera wszystkie podstawowe atrybuty-klucze, które są pod kluczem *result* w odpowiedzi pod na przykład endpoint'em *authors/entries/2871/*. Została utworzona w celu uproszczenia przechowywania w pamięci uzyskanych wyników. Żeby uzyskać przykładowe obiekty klasy Entry została napisana funkcja *get\_author\_entries*, która zwraca wszystkie zasoby autora oraz je konwertuje na obiekty klasy Entry. Jeśli autor posiada więcej niż 100 zasobów to należy z Menu wybrać opcję wyszukiwania po stronie - funkcję *get\_author\_entries\_by\_page*:

```
1 def get_author_entries(author_pk, api_key):
2     try:
3         response = requests.get("https://api.e-science.pl/api/azon/authors/entries/"+str(
4             author_pk)+"/", headers={'X-API-Key': api_key})
5         response.raise_for_status()
6         json_data = response.json()
7         results = json_data['results']
8         entries_list=[]
9         for item in results:
10             entries_list.append(Entry(**item))
11         return entries_list
12     except HTTPError as http_err:
13         print(f'HTTP error occurred: {http_err}')
14
15 def get_author_entries_by_page(author_pk, api_key, page):
16     try:
17         response = requests.get("https://api.e-science.pl/api/azon/authors/entries/"+str(
18             author_pk)+"/?limit=100&offset=" + str(page) + "00", headers={'X-API-Key': api_key})
19         response.raise_for_status()
20         json_data = response.json()
21         results = json_data['results']
22         entries_list=[]
23         for item in results:
24             entries_list.append(Entry(**item))
25         return entries_list
26     except HTTPError as http_err:
27         print(f'HTTP error occurred: {http_err}')
```

## 2.2 Wyszukiwanie zasobów po stronach

W przypadku wyszukiwania zasobów po stronach również została wykorzystana klasa `Entry`, lecz tym razem wykorzystany endpoint to `entry/filter/`. Została napisana metoda `get_entries_by_page`, która jako parametr przyjmuje klucz API oraz numer strony i zwraca listę obiektów klasy `Entry` z danej strony. Ciało funkcji wygląda następująco:

```
1 def get_entries_by_page(api_key, page):
2     try:
3         response = requests.get("https://api.e-science.pl/api/azon/entry/filter/?limit=100&
4         offset=" + str(page) + "00", headers={'X-API-Key': api_key})
5         response.raise_for_status()
6         json_data = response.json()
7         results = json_data['results']
8         entries_list=[]
9         for item in results:
10            entries_list.append(Entry(**item))
11        return entries_list
12
13    except HTTPError as http_err:
14        print(f'HTTP error occurred: {http_err}')
```

## 2.3 Generowanie metadanych wyszukanych zasobów

Każdy wyszukany zasób ma atrybut `entry_type_id`, który określa jaki typ reprezentuje dany zasób. Patrząc na odpowiedź serwera, po zapytaniu na endpoint `entry/types/index/`, wybrane zostały typy, których było najwięcej tak żeby pokryć niemalże całą bazę wszystkich zasobów:

- książka - 1890, `entry_type_id = 1`,
- artykuł, rozdział - 2682, `entry_type_id = 2`,
- czasopismo - 7251, `entry_type_id = 18`,
- zbiór, baza danych - 13174, `entry_type_id = 15`,
- zdjęcie - 1956, `entry_type_id = 6`,
- prezentacja - 107, `entry_type_id = 7`,
- inny dokument - 964, `entry_type_id = 17`,
- kod źródłowy - 62, `entry_type_id = 24`,
- wideo - 3647, `entry_type_id = 14`,
- praca dyplomowa - 373, `entry_type_id = 4`,
- analiza chemiczna - 763, `entry_type_id = 10`,

W ten sposób uzyskano pewność przekonwertowania na format BibTeX 32869 zasobów z ogólnej liczby 41447 zasobów, czyli prawie 80% całej bazy. Jeśli zasobu nie da się przekonwertować to jest po prostu pomijany, a w logach konsoli wyskakuje informacja o `KeyError`. Została skonstruowana prosta funkcja `get_entries_details`, która jako parametr przyjmuje listę obiektów klasy `Entry` (`entries_list`) oraz klucz API (`api_key`) i za pomocą słownika `switcher` tworzy odpowiedni obiekt w zależności od `entry_type_id`. Został dodatkowo wprowadzone uśpienie wątku na 1 sekundę, gdyż przy mniejszych czasach (szczególnie jego braku) serwer syłał błędami typu 429 (*Too Many Requests*).

Jako ciekawostkę mogę podać, że ponad 25% wszystkich zasobów (prawie wszystkie typu zbiór, baza danych) jest autorstwa autora o imieniu Adrian Zakrzewski.

```

1  entries_BibTeXs = []
2  for entry in entries_list:
3      time.sleep(1)
4      pk = entry.pk
5      entry_type = entry.entry_type_id
6      response = requests.get("https://api.e-science.pl/api/azon/entry/"+str(pk)+"/", headers
7      ={'X-API-Key': api_key})
8      response.raise_for_status()
9      json_data = response.json()
10     switcher = {
11         '1': get_book,
12         '2': get_article,
13         '4': get_phdthesis,
14         '6': get_photo,
15         '7': get_misc,
16         '10': get_techreport,
17         '14': get_video,
18         '15': get_misc,
19         '17': get_misc,
20         '18': get_magazine,
21         '24': get_misc
22     }
23     item = switcher[entry_type](json_data)
24     entries_BibTeXs.append(item.get_BibTeX())

```

Każda z funkcji (*get\_book*, *get\_article* etc.) szuka innych kluczy w odpowiedzi (*json\_data*) - takich które są potrzebne lub opcjonalne do powstania BibTeXa. Przykładowo jeśli zasób jest książką funkcja szuka takich kluczy jak poniżej i tworzy odpowiedni obiekt, który po utworzeniu jest zwracany:

```

1  def get_book(json_data):
2      info={}
3      item = json_data['item']
4      info['authors']=json_data['authors']
5      info['title']=json_data['title']
6      info['year']=item['publish_time']
7      info['publisher']=item['publisher']
8      info['isbn']=item['isbn']
9      info['note']=json_data['comments']
10     info['address']=item['publish_place']
11     info['edition']=item['numeration']
12     info['series']=item['series_name']
13     info['pk']=json_data['pk']
14     book = BibTeX_classes.Book(**info)
15     return book

```

Funkcja *get\_entries\_details* przyjmuje jako parametr listę obiektów klasy *Entry*. Taką listę zwracają i funkcja wyszukiwania zasobów po autorze (*get\_author\_entries*) i funkcja wyszukiwania zasobów na poszczególnych stronach (*get\_entries\_by\_page*). Dzięki temu aplikacja jest w stanie wygenerować BibTeX-y dla każdego z wyszukiwanych zasobów aktualnie przechowywanych w pamięci.

Przykładowe wygenerowane BibTeX-y wyglądają następująco (dostępne też w pliku *BibTeX.bib* na gitlabie):

```

1  @inbook{47151,
2    author = "El bieta Litwi ska, Jadwiga Brzuchowska, El bieta Ch dzy ska",
3    title = "Computer models in town planning teaching",
4    publisher = "Katedra Planowania Przestrzennego Wydzia  Architektury Politechniki
5              Wroc awskiej",
6    pages = 2-13,
7    year = 1993,
8    address = "Wroc aw",

```

```

8  note = "Wyst pienie kongresowe. Ze zbioru "Planowanie i rodowisko w zmieniaj cej si
9  }
10
11 @misc{48241,
12  authors = "['Jadwiga Brzuchowska', 'El bieta Litwi ska', 'Tomasz Ossowicz', 'Justyna Ma ko -
13  Osiadacz']",
14  title = "Mo liwo ci powi za wybranych technik wspomagania planowania przestrzennego baz
15  danych typu GIS. Etap II",
16  type = "inny dokument"
17 }
18
19 @book{49843,
20  author = "Tomasz Ossowicz, El bieta Litwi ska, Jadwiga Brzuchowska",
21  title = "Raporty Wydzia Architektury Politechniki Wroc awskiej: Modelowanie rozwoju
22  przestrzennego Bydgoszczy przy pomocy modelu symulacyjno-decyzyjnego ORION",
23  publisher = "Instytut Architektury i Urbanistyki Politechniki Wroc awskiej",
24  year = 1997,
25  series = "Raporty Instytutu Architektury i Urbanistyki PWr",
26  address = "Wroc aw"
27 }
28
29 @book{26215,
30  author = "Tadeusz Zipser, Jerzy S awski, El bieta Litwi ska, Tomasz Ossowicz, Jadwiga
31  Brzuchowska",
32  title = "Model symulacyjno-decyzyjny ORION",
33  publisher = "Wydawnictwo Politechniki Wroc awskiej",
34  year = 1994,
35  address = "Wroc aw",
36  isbn = "83-7085-115-0"
37 }
38
39 @phdthesis{38884,
40  author = "Jerzy S awski",
41  title = "Model dynamiki system w osadniczych oparty na koncepcji potencja w. Zastosowanie
42  wsp czesnych koncepcji dynamiki system w",
43  school = "Politechnika Wroc awska",
44  year = 1990,
45  address = "Wroc aw",
46  note = "Praca doktorska."
47 }

```

W Menu jest opcja zapisania do pliku aktualnie przechowywanych w pamięci BibTeX-ów.

## Bibliografia dla przykładowych BibTeX-ów

- [1] Możliwości powiązań wybranych technik wspomagania planowania przestrzennego bazą danych typu gis. etap ii.
- [2] Elżbieta Chądzyńska Elżbieta Litwińska, Jadwiga Brzuchowska. *Computer models in town planning teaching*, page 2. Katedra Planowania Przestrzennego Wydział Architektury Politechniki Wrocławskiej.
- [3] Jerzy Sławski. *Model dynamiki systemów osadniczych oparty na koncepcji potencjałów. Zastosowanie współczesnych koncepcji dynamiki systemów*. PhD thesis, Politechnika Wrocławska, Wrocław, 1990. Praca doktorska.
- [4] Elżbieta Litwińska Tomasz Ossowicz Jadwiga Brzuchowska Tadeusz Zipser, Jerzy Sławski. *Model symulacyjno-decyzyjny ORION*. Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1994.
- [5] Jadwiga Brzuchowska Tomasz Ossowicz, Elżbieta Litwińska. *Raporty Wydział Architektury Politechniki Wrocławskiej: Modelowanie rozwoju przestrzennego Bydgoszczy przy pomocy modelu symulacyjno-decyzyjnego ORION*. Raporty Instytutu Architektury i Urbanistyki PWr. Instytut Architektury i Urbanistyki Politechniki Wrocławskiej, Wrocław, 1997.

## 2.4 Inne metody oraz wersjonowanie biblioteki

W celu przetestowania, żeby zobaczyć jak to jest z wersjonowaniem zostało napisanych parę metod, a dodanie każdej powodowało utworzenie nowej wersji biblioteki. Dotychczasową wersją biblioteki była wersja v3.0.

### 2.4.1 Lista języków programowania

Po utworzeniu tej metody został nadany nowy tag v4.0.

```
1 def get_programming_languages(api_key):
2     try:
3         response = requests.get("https://api.e-science.pl/api/azon//programminglanguages/",
4         headers={'X-API-Key': api_key})
5         response.raise_for_status()
6         json_data = response.json()
7         results = json_data['results']
8         lang_list=[]
9         for item in results:
10             lang_list.append(item['name'])
11         return lang_list
12
13     except HTTPError as http_err:
14         print(f'HTTP error occurred: {http_err}')
```

### 2.4.2 Lista centrów badawczych

Po utworzeniu tej metody został nadany nowy tag v5.0.

```
1 class ResearchCenters:
2     def __init__(self, _id, name, number, director, tel, email, webpage):
3         self.id=_id
4         self.name=name
5         self.number=number
6         self.director=director
7         self.tel=tel
8         self.email=email
9         self.webpage=webpage
10
11     def __repr__(self):
12         return f'Id - {self.id}\nNazwa - {self.name}\nNumer - {self.number}\nDyrektor - {self.
13         director}\nTelefon - {self.tel}\nE-mail - {self.email}\nStrona - {self.webpage}\n'
14
15 def get_pwr_reseach_centres(api_key):
16     try:
17         response = requests.get("https://api.e-science.pl/api/azon/databases/
18         pwr_research_centers/", headers={'X-API-Key': api_key})
19         response.raise_for_status()
20         json_data = response.json()
21         results = json_data['results']
22         rc_list=[]
23         for item in results:
24             info={}
25             info['_id']=item['id']
26             info['name']=item['name']
27             info['number']=item['number']
28             info['director']=item['director']
29             info['tel']=item['tel']
30             info['email']=item['email']
31             info['webpage']=item['webpage']
32             rc_list.append(ResearchCenters(**info))
33         return rc_list
```

```

33 except HTTPError as http_err:
34     print(f'HTTP error occurred: {http_err}')

```

### 2.4.3 Lista laboratoriów

Po utworzeniu tej metody został nadany nowy tag *v6.0*.

```

1 class Laboratory:
2     def __init__(self, _id, web_address, acronym, full_name, partner):
3         self.id=_id
4         self.web_address=web_address
5         self.acronym=acronym
6         self.full_name=full_name
7         self.partner=partner
8
9     def __repr__(self):
10        return f'Id - {self.id}\nStrona - {self.web_address}\nAkronim - {self.acronym}\nPe na
        nazwa - {self.full_name}\nPartner - {self.partner}\n'
11
12 def get_laboratories(api_key):
13     try:
14         response = requests.get("https://api.e-science.pl/api/azon/databases/elaboratory/",
        headers={'X-API-Key': api_key})
15         response.raise_for_status()
16         json_data = response.json()
17         results = json_data['results']
18         lab_list=[]
19         for item in results:
20             info={}
21             info['_id']=item['id']
22             info['web_address']=item['web_address']
23             info['full_name']=item['full_name']
24             info['acronym']=item['acronym']
25             info['partner']=item['partner']
26             lab_list.append(Laboratory(**info))
27     return lab_list

```

### 2.4.4 Historia wersjonowania

```

C:\Users\Radek\Desktop\6semestr\DPP\lab_9_pip\github>git log --oneline --graph
* 4771908 (HEAD -> biblioteka, tag: v6.0, origin/biblioteka) Add get_laboratories method
* e31c952 (tag: v5.0) Add get_pwr_research_centres method
* 73eaf38 (tag: v4.0) Add get_programming_languages_method
* eeea553 (tag: v3.0) Distribute version v3.0
* 80b1345 Finish method saving BibTex to the file
* 6387b6a (tag: v2.0) Finish converting all types of entries to bibtex
* dc020eb Change package1 structure
* 5302799 Add method for returning author entries
* 6a0a583 Distribute second version
* 8eb9456 Add second package
* 3568060 Update README.md
* 79c871a Distribute package with something else
* 375fc7a Change word in return statement
* 71765fe (tag: v1.0) Add return to say_hello method
* 142864d Distribute first version of library
* 5ccdf23 (origin/master, origin/HEAD, master) Initial commit

```

Rysunek 1: Historia logów

Na podstawie powyższej historii można spodziewać się, że wersja v3.0 nie będzie posiadała żadnej z trzech dodatkowych metod i idąc tak po kolei dopiero wersja ostateczna - czyli v6.0 będzie udostępniać wszystkie metody. Będzie się dało o tym przekonać podczas testowania aplikacji.

## 3 Aplikacja

W celu skorzystania z biblioteki został napisany prosty moduł aplikacji, który miał korzystać metod udostępnianych przez bibliotekę.

### 3.1 Menu

Utworzone menu pozwala na przetestowanie wszystkich wcześniej opisanych metod:

```
PS C:\Users\Radek\Desktop\6semestr\DPP\lab_9_pip\e-science> ${env:DEBUGPY_LAUNCHER_PORT}='57602'; & 'C:\Users\Radek\AppData\Local\Programs\debugpy\wheels\debugpy\launcher' 'c:\Users\Radek\Desktop\6semestr\DPP\lab_9_pip\e-science\main.py'
1: Wyszukaj zasoby po ID autora.
2: Wyszukaj zasoby po ID autora i stronie
3: Wyszukaj zasoby po stronie
4: Wyszukaj języki programowania
5: Wyszukaj centra badawcze PWR
6: Wyszukaj laboratoria
7: Wygeneruj BibTeX-y dla obecnie wyszukiwanych zasobów
8: Zapisz do pliku obecne BibTeX-y
9: Przetestuj pakiet package2
10: Wyjście z programu

Co chcesz zrobić?:
```

Rysunek 2: Utworzone menu



```
Tytuł - Próba zastosowania modelu podsystemu decyzyjnego "ORION" w planowaniu zagospodarowania turystycznego
Id - 46792
Autor - Barbara Jaskólska
Typ - książka
Partner - Politechnika Wrocławska
```

```
Tytuł - Computer models in town planning teaching
Id - 47151
Autor - Elżbieta Litwińska
Typ - artykuł, rozdział
Partner - Politechnika Wrocławska
```

Razem 6 zasobów

- 1: Wyszukaj zasoby po ID autora.
- 2: Wyszukaj zasoby po ID autora i stronie
- 3: Wyszukaj zasoby po stronie
- 4: Wyszukaj języki programowania
- 5: Wyszukaj centra badawcze PWR
- 6: Wyszukaj laboratoria
- 7: Wygeneruj BibTeX-y dla obecnie wyszukiwanych zasobów
- 8: Zapisz do pliku obecne BibTex-y
- 9: Przetestuj pakiet package2
- 10: Wyjście z programu

Co chcesz zrobić?: 7

Wygenerowano razem 6 BibTeX-ów

- 1: Wyszukaj zasoby po ID autora.
- 2: Wyszukaj zasoby po ID autora i stronie
- 3: Wyszukaj zasoby po stronie
- 4: Wyszukaj języki programowania
- 5: Wyszukaj centra badawcze PWR
- 6: Wyszukaj laboratoria
- 7: Wygeneruj BibTeX-y dla obecnie wyszukiwanych zasobów
- 8: Zapisz do pliku obecne BibTex-y
- 9: Przetestuj pakiet package2
- 10: Wyjście z programu

Co chcesz zrobić?: 8

Zapisano razem 6 BibTeX-ów

- 1: Wyszukaj zasoby po ID autora.
- 2: Wyszukaj zasoby po ID autora i stronie
- 3: Wyszukaj zasoby po stronie
- 4: Wyszukaj języki programowania
- 5: Wyszukaj centra badawcze PWR
- 6: Wyszukaj laboratoria
- 7: Wygeneruj BibTeX-y dla obecnie wyszukiwanych zasobów
- 8: Zapisz do pliku obecne BibTex-y
- 9: Przetestuj pakiet package2
- 10: Wyjście z programu

Co chcesz zrobić?:

Rysunek 3: Przykładowy output konsoli

## 3.2 Przetestowanie wersjonowania (upgrade i downgrade)

W celu przetestowania wersjonowania został napisany prosty *bash*'owy skrypt instalujący odpowiednią wersję i sprawdzający czy metody są znajdowane poprzez złapanie odpowiedniego wyjątku *AttributeError*:

```
1 def search_laboratories():
2     try:
3         labs = api_others.get_laboratories(api_key)
4     except AttributeError:
5         return "Nie ma takiej metody jak get_laboratories"
6     return f'Znaleziono {len(labs)} laboratori w'
```

Skrypt wygląda następująco:

```
1 #!/bin/bash
2
3 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v3.0#egg=azon-api-
4     library
5 echo -----Wersja v3.0----- >> results.txt
6 python3 main.py >> results.txt
7
8 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v4.0#egg=azon-api-
9     library
10 echo -----UPGRADE-Wersja v4.0----- >> results.txt
11 python3 main.py >> results.txt
12
13 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v5.0#egg=azon-api-
14     library
15 echo -----UPGRADE-Wersja v5.0----- >> results.txt
16 python3 main.py >> results.txt
17
18 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v6.0#egg=azon-api-
19     library
20 echo -----UPGRADE-Wersja v6.0----- >> results.txt
21 python3 main.py >> results.txt
22
23 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v5.0#egg=azon-api-
24     library
25 echo -----DOWNGRADE-Wersja v5.0----- >> results.txt
26 python3 main.py >> results.txt
27
28 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v4.0#egg=azon-api-
29     library
30 echo -----DOWNGRADE-Wersja v4.0----- >> results.txt
31 python3 main.py >> results.txt
32
33 pip3 install --upgrade git+git://github.com/radosz99/azon-api-library.git@v3.0#egg=azon-api-
34     library
35 echo -----DOWNGRADE-Wersja v3.0----- >> results.txt
36 python3 main.py >> results.txt
```

Output - w celu zaprezentowania rezultatów - został przekierowany do pliku *results.txt*, którego treść po uruchomieniu skryptu wygląda następująco:

```
1 -----Wersja v3.0-----
2 Nie ma takiej metody jak get_programming_languages
3 Nie ma takiej metody jak get_pwr_reseach_centres
4 Nie ma takiej metody jak get_laboratories
5
6 -----UPGRADE-Wersja v4.0-----
7 Znaleziono 30 jezykow programowania
8 Nie ma takiej metody jak get_pwr_reseach_centres
9 Nie ma takiej metody jak get_laboratories
10
11 -----UPGRADE-Wersja v5.0-----
12 Znaleziono 30 jezykow programowania
13 Znaleziono 15 centrow badawczych
14 Nie ma takiej metody jak get_laboratories
15
16 -----UPGRADE-Wersja v6.0-----
17 Znaleziono 30 jezykow programowania
18 Znaleziono 15 centrow badawczych
19 Znaleziono 38 laboratoriiow
20
21 -----DOWNGRADE-Wersja v5.0-----
22 Znaleziono 30 jezykow programowania
23 Znaleziono 15 centrow badawczych
24 Nie ma takiej metody jak get_laboratories
25
26 -----DOWNGRADE-Wersja v4.0-----
27 Znaleziono 30 jezykow programowania
28 Nie ma takiej metody jak get_pwr_reseach_centres
29 Nie ma takiej metody jak get_laboratories
30
31 -----DOWNGRADE-Wersja v3.0-----
32 Nie ma takiej metody jak get_programming_languages
33 Nie ma takiej metody jak get_pwr_reseach_centres
34 Nie ma takiej metody jak get_laboratories
```

Treść zgadza się z tym co powinno być - w wersji *v3.0* nie było żadnej z dodatkowych metod, w wersji *v4.0* doszło wyszukiwanie języków programowania, w wersji *v5.0* wyszukiwanie centrów badawczych, zaś w ostatniej - *v6.0* - wyszukiwanie laboratoriów.