

Implementacja i testowanie endpointów

Framework Flask
ARIUS Zadanie 4.

Julia Chilczuk 325261

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

23 kwietnia 2025

Spis treści

1. Wstęp	2
2. Endpoint /teacher-list	2
3. Endpoint /teacher-details	4
4. Endpoint /book-lesson	7
5. Endpoint /add-teacher	11
6. Endpoint /get-lessons	14
7. Wireshark	18

1. Wstęp

W ramach realizacji zadania zaimplementowano oraz przetestowano endpointy aplikacji przygotowanej w ramach laboratorium 3. Wykorzystano framework Flask.

2. Endpoint /teacher-list

Endpoint /teacher-list zwraca listę dostępnych nauczycieli w formacie JSON, nie przyjmuje argumentów. Dane na wyjściu to:

- id_nauczyciela
- imię
- nazwisko
- prowadzone przedmioty

```
# Lista nauczycieli
@app.route('/teacher-list', methods=['GET'])
def teacher_list():
    teachers = Teacher.query.all()
    teacher_data = [
        {
            "id_nauczyciela": teacher.id,
            "imie": teacher.first_name,
            "nazwisko": teacher.last_name,
            "przedmioty": [subject.name for subject in teacher.subjects]
        }
        for teacher in teachers
    ]
    return jsonify({"nauczyciele": teacher_data}), 200
```

Rys. 1: Implementacja endpointu /teacher-list

```
PS C:\Users\j-chi\Documents\6_semestr\ARIUS\zad3\flask_env\Scripts> python test.py
Test /teacher-list:
{
  "nauczyciele": [
    {
      "id_nauczyciela": 1,
      "imie": "Jan",
      "nazwisko": "Kowalski",
      "przedmioty": [
        "matematyka",
        "fizyka"
      ]
    },
    {
      "id_nauczyciela": 2,
      "imie": "Anna",
      "nazwisko": "Nowak",
      "przedmioty": [
        "chemia",
        "biologia"
      ]
    }
  ]
}
```

Rys. 2: Wynik wywołania endpointu /teacher-list

2388...	5395.461339	127.0.0.1	127.0.0.1	HTTP	232 GET /teacher-list HTTP/1.1
2388...	5395.461465	127.0.0.1	127.0.0.1	TCP	44 5000 → 57768 [ACK] Seq=1 Ack=189 Win=65280 Len=0
2388...	5395.555207	127.0.0.1	127.0.0.1	TCP	210 5000 → 57768 [PSH, ACK] Seq=1 Ack=189 Win=65280 Len=0
2388...	5395.555275	127.0.0.1	127.0.0.1	TCP	44 57768 → 5000 [ACK] Seq=189 Ack=167 Win=65280 Len=0
2388...	5395.555465	127.0.0.1	127.0.0.1	HTTP/J...	886 HTTP/1.1 200 OK , JSON (application/json)
2388...	5395.555503	127.0.0.1	127.0.0.1	TCP	44 57768 → 5000 [ACK] Seq=189 Ack=1009 Win=64512 Len=0

▶ Frame 238801: 232 bytes on wire (1856 bits), 232 bytes captured (1856 bits) on interface \Device\NPF_{Loopback}, id 0
 ▶ Null/Loopback
 ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 ▶ Transmission Control Protocol, Src Port: 57768, Dst Port: 5000, Seq: 1, Ack: 1, Len: 188
 ▶ Hypertext Transfer Protocol
 GET /teacher-list HTTP/1.1\r\n
 Request Method: GET
 Request URI: /teacher-list
 Request Version: HTTP/1.1
 Host: localhost:5000\r\n
 User-Agent: python-requests/2.32.3\r\n
 Accept-Encoding: gzip, deflate\r\n
 Accept: */*\r\n
 Connection: keep-alive\r\n
 Authorization: Julia Chilczuk\r\n
 \r\n
 [Response in frame: 238805]
 [Full request URI: http://localhost:5000/teacher-list]

Rys. 3: Zapytanie HTTP

2388...	5395.555275	127.0.0.1	127.0.0.1	TCP	44 57768 → 5000 [ACK] Seq=189 Ack=167 Win=65280 Len=0
2388...	5395.555465	127.0.0.1	127.0.0.1	HTTP/J...	886 HTTP/1.1 200 OK , JSON (application/json)
2388...	5395.555503	127.0.0.1	127.0.0.1	TCP	44 57768 → 5000 [ACK] Seq=189 Ack=1009 Win=64512 Len=0
2388...	5395.555934	127.0.0.1	127.0.0.1	TCP	44 57768 → 5000 [FIN, ACK] Seq=189 Ack=1009 Win=64512 Len=0

▶ Frame 238805: 886 bytes on wire (7088 bits), 886 bytes captured (7088 bits) on interface \Device\NPF_{Loopback}, id 0
 ▶ Null/Loopback
 ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 ▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 57768, Seq: 167, Ack: 189, Len: 842
 ▶ [2 Reassembled TCP Segments (1008 bytes): #238803(166), #238805(842)]
 ▶ Hypertext Transfer Protocol
 HTTP/1.1 200 OK\r\n
 Response Version: HTTP/1.1
 Status Code: 200
 [Status Code Description: OK]
 Response Phrase: OK
 Server: Werkzeug/3.1.3 Python/3.13.0\r\n
 Date: Wed, 23 Apr 2025 10:44:21 GMT\r\n
 Content-Type: application/json\r\n
 Content-Length: 842\r\n
 Connection: close\r\n
 \r\n
 [Request in frame: 238801]
 [Time since request: 0.094126000 seconds]
 [Request URI: /teacher-list]
 [Full request URI: http://localhost:5000/teacher-list]
 File Data: 842 bytes
 ▶ JavaScript Object Notation: application/json
 Object
 Member: nauczyciele
 Array
 Object
 Member: id_nauczyciela
 Member: imie
 Member: nazwisko
 Member: przedmioty
 Object
 Member: id_nauczyciela
 [Path with value: /nauczyciele/[]/id_nauczyciela:2]
 [Member with value: id_nauczyciela:2]
 Number value: 2
 Key: id_nauczyciela
 [Path: /nauczyciele/[]/id_nauczyciela]
 Member: imie

Rys. 4: Odpowiedź HTTP

3. Endpoint /teacher-details

Endpoint `/teacher-details` przyjmuje argument `id_nauczyciela` jako dane wejściowe i zwraca wszystkie dane tego nauczyciela (włącznie z prowadzonymi przedmiotami i kalendarzem) w formacie JSON lub informację, że takiego nauczyciela nie ma.

```
@app.route('/teacher-details', methods=['GET'])
def teacher_details():
    teacher_id = request.args.get('id_nauczyciela', type=int)
    teacher = Teacher.query.get(teacher_id)
    if not teacher:
        return jsonify({"error": f"Nauczyciel o ID {teacher_id} nie istnieje."}), 404

    teacher_data = {
        "id_nauczyciela": teacher.id,
        "imie": teacher.first_name,
        "nazwisko": teacher.last_name,
        "opis": teacher.description,
        "ocena": teacher.rating,
        "numer_telefonu": teacher.phone_number,
        "stawka": teacher.rate,
        "waluta": teacher.currency,
        "email": teacher.email,
        "przedmioty": [subject.name for subject in teacher.subjects],
        "grafik": [
            {"od": schedule.available_from, "do": schedule.available_to}
            for schedule in teacher.schedule
        ]
    }
    return jsonify(teacher_data), 200
```

Rys. 5: Implementacja endpointu `/teacher-list`.

```

PS C:\Users\j-chi\Documents\6_semestr\ARIUS\zad3\flask_env\Scripts> python test.py

Test /teacher-details (pozytywny):
{
  "email": "jan.kowalski@example.com",
  "grafik": [
    {
      "do": "17:00",
      "od": "09:00"
    }
  ],
  "id_nauczyciela": 1,
  "imie": "Jan",
  "nazwisko": "Kowalski",
  "numer_telefonu": "123456789",
  "ocena": 4.5,
  "opis": null,
  "przedmioty": [
    "matematyka",
    "fizyka"
  ],
  "stawka": 50,
  "waluta": "PLN"
}

Status code: 200

Test /teacher-details (negatywny):
{
  "error": "Nauczyciel o ID 999 nie istnieje."
}

Status code: 404

```

Rys. 6: Wynik wywołania endpointu /teacher-list.

```

2809... 6303.573816 127.0.0.1 127.0.0.1 HTTP 252 GET /teacher-details?id_nauczyciela=1 HTTP/1.1
2809... 6303.573940 127.0.0.1 127.0.0.1 TCP 44 5000 → 58293 [ACK] Seq=1 Ack=209 Win=65280 Len=0
2809... 6303.605840 127.0.0.1 127.0.0.1 TCP 210 5000 → 58293 [PSH, ACK] Seq=1 Ack=209 Win=65280 Len=0
2809... 6303.605886 127.0.0.1 127.0.0.1 TCP 44 58293 → 5000 [ACK] Seq=209 Ack=167 Win=65280 Len=0
2809... 6303.605931 127.0.0.1 127.0.0.1 HTTP/1.1 378 HTTP/1.1 200 OK , JSON (application/json)
2809... 6303.605958 127.0.0.1 127.0.0.1 TCP 44 58293 → 5000 [ACK] Seq=209 Ack=501 Win=65024 Len=0

> Frame 280959: 252 bytes on wire (2016 bits), 252 bytes captured (2016 bits) on interface \Device\NPF_{...}, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 58293, Dst Port: 5000, Seq: 1, Ack: 1, Len: 208
> Hypertext Transfer Protocol
  GET /teacher-details?id_nauczyciela=1 HTTP/1.1\r\n
    Request Method: GET
    Request URI: /teacher-details?id_nauczyciela=1
      Request URI Path: /teacher-details
      Request URI Query: id_nauczyciela=1
    Request Version: HTTP/1.1
  Host: localhost:5000\r\n
  User-Agent: python-requests/2.32.3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept: */*\r\n
  Connection: keep-alive\r\n
  Authorization: Julia Chilczuk\r\n
  \r\n
  [Response in frame: 280963]
  [Full request URI: http://localhost:5000/teacher-details?id_nauczyciela=1]

```

Rys. 7: Zapytanie HTTP - poprawne dane wejściowe.

```

2809... 6303.605931 127.0.0.1 127.0.0.1 TCP 44 58293 → 5000 [ACK] Seq=209 Ack=167 Win=65280 Len=0
2809... 6303.605931 127.0.0.1 127.0.0.1 HTTP/J... 378 HTTP/1.1 200 OK , JSON (application/json)
2809... 6303.605958 127.0.0.1 127.0.0.1 TCP 44 58293 → 5000 [ACK] Seq=209 Ack=501 Win=65024 Len=0
2809... 6303.606361 127.0.0.1 127.0.0.1 TCP 44 58293 → 5000 [FIN, ACK] Seq=209 Ack=501 Win=65024 Len=0
▶ Frame 280963: 378 bytes on wire (3024 bits), 378 bytes captured (3024 bits) on interface \Device\NPF_{Loopback}, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 58293, Seq: 167, Ack: 209, Len: 334
▶ [2 Reassembled TCP Segments (500 bytes): #280961(166), #280963(334)]
▶ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\r\n
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 10:59:29 GMT\r\n
    Content-Type: application/json\r\n
  ▶ Content-Length: 334\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 280959]
  [Time since request: 0.032115000 seconds]
  [Request URI: /teacher-details?id_nauczyciela=1]
  [Full request URI: http://localhost:5000/teacher-details?id_nauczyciela=1]
  File Data: 334 bytes
  ▶ JavaScript Object Notation: application/json
    ▶ Object
      ▶ Member: email
      ▶ Member: grafik
      ▶ Member: id_nauczyciela
      ▶ Member: imie
      ▶ Member: nazwisko
      ▶ Member: numer_telefonu
      ▶ Member: ocena
        [Path with value: /ocena:4.5]
        [Member with value: ocena:4.5]
        Number value: 4,5
        Key: ocena
        [Path: /ocena]
      ▶ Member: opis
      ▶ Member: przedmioty
      ▶ Member: stawka
      ▶ Member: waluta

```

Rys. 8: Poprawna odpowiedź HTTP.

```

2809... 6305.654039 127.0.0.1 127.0.0.1 HTTP 254 GET /teacher-details?id_nauczyciela=999 HTTP/1.1
2809... 6305.654169 127.0.0.1 127.0.0.1 TCP 44 5000 → 58296 [ACK] Seq=1 Ack=211 Win=65280 Len=0
2809... 6305.660842 127.0.0.1 127.0.0.1 TCP 216 5000 → 58296 [PSH, ACK] Seq=1 Ack=211 Win=65280 Len=0
2809... 6305.660926 127.0.0.1 127.0.0.1 TCP 44 58296 → 5000 [ACK] Seq=211 Ack=173 Win=65280 Len=0
2809... 6305.661027 127.0.0.1 127.0.0.1 HTTP/J... 95 HTTP/1.1 404 NOT FOUND , JSON (application/json)
2809... 6305.661096 127.0.0.1 127.0.0.1 TCP 44 58296 → 5000 [ACK] Seq=211 Ack=224 Win=65280 Len=0
2809... 6305.661766 127.0.0.1 127.0.0.1 TCP 44 58296 → 5000 [FIN, ACK] Seq=211 Ack=224 Win=65280 Len=0
2809... 6305.661841 127.0.0.1 127.0.0.1 TCP 44 5000 → 58296 [ACK] Seq=224 Ack=212 Win=65280 Len=0
▶ Frame 280992: 254 bytes on wire (2032 bits), 254 bytes captured (2032 bits) on interface \Device\NPF_{Loopback}, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 58296, Dst Port: 5000, Seq: 1, Ack: 1, Len: 210
▶ Hypertext Transfer Protocol
  ▶ GET /teacher-details?id_nauczyciela=999 HTTP/1.1\r\n
    Request Method: GET
    ▶ Request URI: /teacher-details?id_nauczyciela=999
      Request URI Path: /teacher-details
      ▶ Request URI Query: id_nauczyciela=999
    Request Version: HTTP/1.1
    Host: localhost:5000\r\n
    User-Agent: python-requests/2.32.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: */*\r\n
    Connection: keep-alive\r\n
    Authorization: Julia Chilczuk\r\n
    \r\n
    [Response in frame: 280996]
    [Full request URI: http://localhost:5000/teacher-details?id_nauczyciela=999]

```

Rys. 9: Odpowiedź HTTP - błędne dane wejściowe.

```

2809... 6305.661027 127.0.0.1 127.0.0.1 TCP 44 58296 → 5000 [ACK] Seq=211 Ack=173 Win=65280 Len=
+ 2809... 6305.661027 127.0.0.1 127.0.0.1 HTTP/J... 95 HTTP/1.1 404 NOT FOUND , JSON (application/json)
2809... 6305.661096 127.0.0.1 127.0.0.1 TCP 44 58296 → 5000 [ACK] Seq=211 Ack=224 Win=65280 Len=
▶ Frame 280996: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface \Device\NPF_{Loopback}, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 58296, Seq: 173, Ack: 211, Len: 51
▶ [2 Reassembled TCP Segments (223 bytes): #280994(172), #280996(51)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 404 NOT FOUND\r\n
    Response Version: HTTP/1.1
    Status Code: 404
    [Status Code Description: Not Found]
    Response Phrase: NOT FOUND
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 10:59:31 GMT\r\n
    Content-Type: application/json\r\n
  ▶ Content-Length: 51\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 280992]
  [Time since request: 0.006988000 seconds]
  [Request URI: /teacher-details?id_nauczyciela=999]
  [Full request URI: http://localhost:5000/teacher-details?id_nauczyciela=999]
  File Data: 51 bytes
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▼ Member: error
      [Path with value: /error:Nauczyciel o ID 999 nie istnieje.]
      [Member with value: error:Nauczyciel o ID 999 nie istnieje.]
      String value: Nauczyciel o ID 999 nie istnieje.
      Key: error
      [Path: /error]

```

Rys. 10: Niepoprawna odpowiedź HTTP.

4. Endpoint /book-lesson

Endpoint /book-lesson służy do rezerwacji lekcji z danym nauczycielem. Przyjmuje argumenty `id_studenta`, `id_nauczyciela`, `data`, `godzina` oraz `nazwa przedmiotu` jako dane wejściowe. Endpoint zwraca potwierdzenie wykonania rezerwacji lub informację o niemożliwości rezerwacji z powodu:

- wskazany termin jest poza godzinami pracy nauczyciela,
- wskazany termin jest zarezerwowany,
- niepoprawne dane wejściowe lub ich brak.

```

@app.route('/book-lesson', methods=['POST'])
def book_lesson():
    data = request.json
    student_id = data.get('id_studenta')
    teacher_id = data.get('id_nauczyciela')
    date = datetime.strptime(data.get('data'), '%Y-%m-%d').date()
    time = datetime.strptime(data.get('czas'), '%H:%M').time() # Pobranie godziny lekcji
    subject_name = data.get('nazwa_przedmiotu')

    # Pobranie danych z bazy
    teacher = Teacher.query.get(teacher_id)
    student = Student.query.get(student_id)
    subject = Subject.query.filter_by(name=subject_name).first()

    # Walidacja danych wejściowych
    if not teacher:
        return jsonify({"error": f"Nauczyciel o ID '{teacher_id}' nie istnieje."}), 404
    if not student:
        return jsonify({"error": "Student o podanym ID nie istnieje."}), 404
    if not subject:
        return jsonify({"error": f"Przedmiot '{subject_name}' nie istnieje."}), 400

    # Sprawdzenie grafiku nauczyciela
    schedule = Schedule.query.filter_by(teacher_id=teacher_id).first()
    if not schedule:
        return jsonify({"error": "Grafik nauczyciela nie został zdefiniowany."}), 400

    if not (schedule.available_from <= time.strftime('%H:%M') <= schedule.available_to):
        return jsonify({"error": "Termin poza godzinami pracy nauczyciela."}), 400

```

Rys. 11: Implementacja endpointu /book-lesson

```

# Sprawdzenie, czy termin jest już zajęty
existing_lesson = Lesson.query.filter_by(
    teacher_id=teacher_id,
    date=date,
    time=time
).first()
if existing_lesson:
    return jsonify({"error": "Termin jest już zarezerwowany."}), 400

# Rezerwacja lekcji
new_lesson = Lesson(
    teacher_id=teacher_id,
    student_id=student_id,
    subject_id=subject.id,
    date=date,
    time=time
)
db.session.add(new_lesson)
db.session.commit()
return jsonify({"message": "Lekcja została pomyślnie zarezerwowana."}), 201

```

Rys. 12: Implementacja endpointu /book-lesson c.d.


```

PS C:\Users\j-chi\Documents\6_semestr\ARIUS\zad3\flask_env\Scripts> python test.py

Test /book-lesson (pozytywny):
{
  "message": "Lekcja została pomyslnie zarezerwowana."
}

Status code: 201

Test /book-lesson (Termin poza godzinami pracy nauczyciela):
{
  "error": "Termin poza godzinami pracy nauczyciela."
}

Status code: 400

```

Rys. 13: Wynik wywołania endpointu /book-lesson.

```

3207... 7159.204877 127.0.0.1 127.0.0.1 HTTP/JSON 156 POST /book-lesson HTTP/1.1 , JSON (application/json)
3207... 7159.204901 127.0.0.1 127.0.0.1 TCP 44 5000 → 58811 [ACK] Seq=1 Ack=354 Win=65024 Len=0
▶ Frame 320756: 156 bytes on wire (1248 bits), 156 bytes captured (1248 bits) on interface \Device\NPF_{...} id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 58811, Dst Port: 5000, Seq: 242, Ack: 1, Len: 112
▶ [2 Reassembled TCP Segments (353 bytes): #320754(241), #320756(112)]
▼ Hypertext Transfer Protocol
  POST /book-lesson HTTP/1.1\r\n
    Request Method: POST
    Request URI: /book-lesson
    Request Version: HTTP/1.1
    Host: localhost:5000\r\n
    User-Agent: python-requests/2.32.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: */*\r\n
    Connection: keep-alive\r\n
    Authorization: Julia Chilczuk\r\n
    Content-Length: 112\r\n
    Content-Type: application/json\r\n
    \r\n
    [Response in frame: 320760]
    [Full request URI: http://localhost:5000/book-lesson]
    File Data: 112 bytes
▼ JavaScript Object Notation: application/json
  Object
    Member: id_studenta
      [Path with value: /id_studenta:1]
      [Member with value: id_studenta:1]
      Number value: 1
      Key: id_studenta
      [Path: /id_studenta]
    Member: id_nauczyciela
    Member: data
    Member: czas
    Member: nazwa_przedmiotu

```

Rys. 14: Zapytanie HTTP - poprawne dane wejściowe.

```

3207... 7159.214953 127.0.0.1 127.0.0.1 TCP 44 58811 → 5000 [ACK] Seq=354 Ack=230 Win=65280 Len=0
3207... 7159.214953 127.0.0.1 127.0.0.1 HTTP/JSON 103 HTTP/1.1 201 CREATED , JSON (application/json)
3207... 7159.214987 127.0.0.1 127.0.0.1 TCP 44 58811 → 5000 [ACK] Seq=354 Ack=230 Win=65280 Len=0
3207... 7159.215292 127.0.0.1 127.0.0.1 TCP 44 58811 → 5000 [FIN, ACK] Seq=354 Ack=230 Win=65280 Len=0
3207... 7159.215335 127.0.0.1 127.0.0.1 TCP 44 5000 → 58811 [ACK] Seq=230 Ack=355 Win=65024 Len=0

▶ Frame 320760: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface \Device\NPF_{...}, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 58811, Seq: 171, Ack: 354, Len: 59
▶ [2 Reassembled TCP Segments (229 bytes): #320758(170), #320760(59)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 201 CREATED\r\n
    Response Version: HTTP/1.1
    Status Code: 201
    [Status Code Description: Created]
    Response Phrase: CREATED
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 11:13:45 GMT\r\n
    Content-Type: application/json\r\n
  ▶ Content-Length: 59\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 320756]
  [Time since request: 0.010076000 seconds]
  [Request URI: /book-lesson]
  [Full request URI: http://localhost:5000/book-lesson]
  File Data: 59 bytes
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▼ Member: message
      [Path with value: /message:Lekcja zostala pomyslnie zarezerwowana.]
      [Member with value: message:Lekcja zostala pomyslnie zarezerwowana.]
      String value: Lekcja zostala pomyslnie zarezerwowana.
      Key: message
      [Path: /message]

```

Rys. 15: Poprawna odpowiedź HTTP.

```

3209... 7161.241010 127.0.0.1 127.0.0.1 HTTP/JSON 156 POST /book-lesson HTTP/1.1 , JSON (application/json)
3209... 7161.241055 127.0.0.1 127.0.0.1 TCP 44 5000 → 58814 [ACK] Seq=1 Ack=354 Win=65024 Len=0
3209... 7161.244366 127.0.0.1 127.0.0.1 TCP 218 5000 → 58814 [PSH, ACK] Seq=1 Ack=354 Win=65024 Len=170
3209... 7161.244409 127.0.0.1 127.0.0.1 TCP 44 58814 → 5000 [ACK] Seq=354 Ack=175 Win=65280 Len=0
3209... 7161.244654 127.0.0.1 127.0.0.1 HTTP/JSON 102 HTTP/1.1 400 BAD REQUEST , JSON (application/json)
3209... 7161.244691 127.0.0.1 127.0.0.1 TCP 44 58814 → 5000 [ACK] Seq=354 Ack=233 Win=65280 Len=0

▶ Frame 320951: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface \Device\NPF_{...}, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 58814, Seq: 175, Ack: 354, Len: 58
▶ [2 Reassembled TCP Segments (232 bytes): #320949(174), #320951(58)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 400 BAD REQUEST\r\n
    Response Version: HTTP/1.1
    Status Code: 400
    [Status Code Description: Bad Request]
    Response Phrase: BAD REQUEST
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 11:13:47 GMT\r\n
    Content-Type: application/json\r\n
  ▶ Content-Length: 58\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 320947]
  [Time since request: 0.003644000 seconds]
  [Request URI: /book-lesson]
  [Full request URI: http://localhost:5000/book-lesson]
  File Data: 58 bytes
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▼ Member: error
      [Path with value: /error:Termin poza godzinami pracy nauczyciela.]
      [Member with value: error:Termin poza godzinami pracy nauczyciela.]
      String value: Termin poza godzinami pracy nauczyciela.
      Key: error
      [Path: /error]

```

Rys. 16: Niepoprawna odpowiedź HTTP.

5. Endpoint /add-teacher

Endpoint /add-teacher służy do dodawania do bazy danych nowego nauczyciela. Przyjmuje argumenty imię, nazwisko, prowadzone przedmioty, opis, ocena nauczyciela, numer telefonu, stawka, waluta, email, grafik pracy. Atrybut stawka musi być liczbą całkowitą. Endpoint zwraca id_nauczyciela lub informację o niemożliwości dodania nauczyciela z powodu braku/złych danych.

```
@app.route('/add-teacher', methods=['POST'])
def add_teacher():
    data = request.json
    try:
        first_name = data['imie']
        last_name = data['nazwisko']
        description = data.get('opis', '')
        rating = float(data['ocena'])
        phone_number = data['numer_telefonu']
        rate = data['stawka']
        currency = data['waluta']
        email = data['email']
        subjects = data['przedmioty']
        schedule = data['grafik']

        # Walidacja stawki
        if not isinstance(rate, int):
            return jsonify({"error": "Stawka musi być liczba całkowita."}), 400

        # Walidacja przedmiotów
        subject_objects = Subject.query.filter(Subject.name.in_(subjects)).all()
        if len(subject_objects) != len(subjects):
            return jsonify({"error": "Niektóre przedmioty są niepoprawne."}), 400
```

Rys. 17: Implementacja endpointu /add-teacher

```
# Tworzenie nauczyciela
new_teacher = Teacher(
    first_name=first_name,
    last_name=last_name,
    description=description,
    rating=rating,
    phone_number=phone_number,
    rate=rate,
    currency=currency,
    email=email,
    subjects=subject_objects
)
db.session.add(new_teacher)
db.session.flush() # Pobranie ID nauczyciela przed zatwierdzeniem

# Dodanie grafiku
new_schedule = Schedule(
    teacher_id=new_teacher.id,
    available_from=schedule['od'],
    available_to=schedule['do']
)
db.session.add(new_schedule)
db.session.commit()

return jsonify({"id_nauczyciela": new_teacher.id}), 201
except KeyError as e:
    return jsonify({"error": f"Brak wymaganego pola: {str(e)}"}), 400
except ValueError as e:
    return jsonify({"error": str(e)}), 400
```

Rys. 18: Implementacja endpointu /add-teacher c.d.

```

PS C:\Users\j-chi\Documents\6_semestr\ARIUS\zad3\flask_env\Scripts> python test.py

Test /add-teacher (pozytywny):
{
  "id_nauczyciela": 6
}

Status code: 201

Test /add-teacher (negatywny):
{
  "error": "Brak wymaganego pola: 'ocena'"
}

Status code: 400

```

Rys. 19: Wynik wywołania endpointu /book-lesson.

```

3447... 7702.644110 127.0.0.1 127.0.0.1 HTTP/JSON 314 POST /add-teacher HTTP/1.1 , JSON (application/json)
3447... 7702.644110 127.0.0.1 127.0.0.1 768 44 5000 50000 51601 51601 1 1 513 116 65031 1 0
▶ Frame 344786: 314 bytes on wire (2512 bits), 314 bytes captured (2512 bits) on interface \Device\NPF_{...}, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 59102, Dst Port: 5000, Seq: 242, Ack: 1, Len: 270
▶ [2 Reassembled TCP Segments (511 bytes): #344784(241), #344786(270)]
▼ Hypertext Transfer Protocol
  ▼ POST /add-teacher HTTP/1.1\r\n
    Request Method: POST
    Request URI: /add-teacher
    Request Version: HTTP/1.1
    Host: localhost:5000\r\n
    User-Agent: python-requests/2.32.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: */*\r\n
    Connection: keep-alive\r\n
    Authorization: Julia Chilczuk\r\n
    ▶ Content-Length: 270\r\n
    Content-Type: application/json\r\n
    \r\n
    [Response in frame: 344790]
    [Full request URI: http://localhost:5000/add-teacher]
    File Data: 270 bytes
  ▼ JavaScript Object Notation: application/json
    ▼ Object
      ▼ Member: imie
        [Path with value: /imie:Adam]
        [Member with value: imie:Adam]
        String value: Adam
        Key: imie
        [Path: /imie]
      ▶ Member: nazwisko
      ▶ Member: opis
      ▶ Member: ocena
      ▶ Member: numer_telefonu
      ▶ Member: stawka
      ▶ Member: waluta
      ▶ Member: email
      ▶ Member: przedmioty
      ▶ Member: grafik

```

Rys. 20: Zapytanie HTTP - poprawne dane wejściowe.

```

3447... 7702.674184 127.0.0.1 127.0.0.1 TCP 44 59102 → 5000 [ACK] Seq=512 Ack=171 Win=65280 Len=0
3447... 7702.674230 127.0.0.1 127.0.0.1 HTTP/JSON 70 HTTP/1.1 201 CREATED , JSON (application/json)
3447... 7702.674257 127.0.0.1 127.0.0.1 TCP 44 59102 → 5000 [ACK] Seq=512 Ack=197 Win=65280 Len=0
3447... 7702.674633 127.0.0.1 127.0.0.1 TCP 44 59102 → 5000 [FIN, ACK] Seq=512 Ack=197 Win=65280 Len=0

> Frame 344790: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{...}, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 5000, Dst Port: 59102, Seq: 171, Ack: 512, Len: 26
> [2 Reassembled TCP Segments (196 bytes): #344788(170), #344790(26)]
Hypertext Transfer Protocol
  HTTP/1.1 201 CREATED\r\n
    Response Version: HTTP/1.1
    Status Code: 201
    [Status Code Description: Created]
    Response Phrase: CREATED
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 11:22:48 GMT\r\n
    Content-Type: application/json\r\n
  Content-Length: 26\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 344786]
  [Time since request: 0.030120000 seconds]
  [Request URI: /add-teacher]
  [Full request URI: http://localhost:5000/add-teacher]
  File Data: 26 bytes
JavaScript Object Notation: application/json
  Object
    Member: id_nauczyciela
      [Path with value: /id_nauczyciela:6]
      [Member with value: id_nauczyciela:6]
      Number value: 6
      Key: id_nauczyciela
      [Path: /id_nauczyciela]

```

Rys. 21: Poprawna odpowiedź HTTP.

```

3449... 7704.713531 127.0.0.1 127.0.0.1 TCP 44 5000 → 59105 [ACK] Seq=1 Ack=241 Win=65024 Len=0
3449... 7704.713673 127.0.0.1 127.0.0.1 HTTP/JSON 81 POST /add-teacher HTTP/1.1 , JSON (application/json)
3449... 7704.713742 127.0.0.1 127.0.0.1 TCP 44 5000 → 59105 [ACK] Seq=1 Ack=278 Win=65024 Len=0
3449... 7704.713753 127.0.0.1 127.0.0.1 TCP 44 5000 → 59105 [ACK] Seq=1 Ack=278 Win=65024 Len=0

> Frame 344905: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface \Device\NPF_{...}, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 59105, Dst Port: 5000, Seq: 241, Ack: 1, Len: 37
> [2 Reassembled TCP Segments (277 bytes): #344903(240), #344905(37)]
Hypertext Transfer Protocol
  POST /add-teacher HTTP/1.1\r\n
    Request Method: POST
    Request URI: /add-teacher
    Request Version: HTTP/1.1
    Host: localhost:5000\r\n
    User-Agent: python-requests/2.32.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: */*\r\n
    Connection: keep-alive\r\n
    Authorization: Julia Chilczuk\r\n
  Content-Length: 37\r\n
  Content-Type: application/json\r\n
  \r\n
  [Response in frame: 344909]
  [Full request URI: http://localhost:5000/add-teacher]
  File Data: 37 bytes
JavaScript Object Notation: application/json
  Object
    Member: imie
      [Path with value: /imie:Adam]
      [Member with value: imie:Adam]
      String value: Adam
      Key: imie
      [Path: /imie]
    Member: nazwisko
      [Path with value: /nazwisko:Nowak]
      [Member with value: nazwisko:Nowak]
      String value: Nowak
      Key: nazwisko
      [Path: /nazwisko]

```

Rys. 22: Zapytanie HTTP - błędne dane wejściowe.

```

3449... 7704.720362 127.0.0.1 127.0.0.1 TCP 44 59105 → 5000 [ACK] Seq=278 Ack=175 Win=65280 Len=0
3449... 7704.720362 127.0.0.1 127.0.0.1 HTTP/JSON 91 HTTP/1.1 400 BAD REQUEST , JSON (application/json)
3449... 7704.720454 127.0.0.1 127.0.0.1 TCP 44 59105 → 5000 [ACK] Seq=278 Ack=175 Win=65280 Len=0
▶ Frame 344909: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface \Device\NPF_{...}_Loopback, id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 59105, Seq: 175, Ack: 278, Len: 47
▶ [2 Reassembled TCP Segments (221 bytes): #344907(174), #344909(47)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 400 BAD REQUEST\r\n
    Response Version: HTTP/1.1
    Status Code: 400
    [Status Code Description: Bad Request]
    Response Phrase: BAD REQUEST
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 11:22:50 GMT\r\n
    Content-Type: application/json\r\n
  ▶ Content-Length: 47\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 344905]
  [Time since request: 0.006689000 seconds]
  [Request URI: /add-teacher]
  [Full request URI: http://localhost:5000/add-teacher]
  File Data: 47 bytes
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▼ Member: error
      [Path with value: /error:Brak wymaganego pola: 'ocena']
      [Member with value: error:Brak wymaganego pola: 'ocena']
      String value: Brak wymaganego pola: 'ocena'
      Key: error
      [Path: /error]

```

Rys. 23: Niepoprawna odpowiedź HTTP.

6. Endpoint /get-lessons

Endpoint `/get-lessons` służy do pozyskiwania informacji o zarezerwowanych lekcjach danego studenta we wskazanym przedziale czasu. Przyjmuje argumenty `id_studenta`, `data_początkowa`, `data_końcowa`, zwraca listę zarezerwowanych lekcji, z informacją o dacie i godzinie, `id_nauczyciela`, imię i nazwisko nauczyciela oraz przedmiot.

```

@app.route('/get-lessons', methods=['GET'])
def get_lessons():
    try:
        # Pobranie i walidacja parametrów wejściowych
        student_id = request.args.get('id_studenta', type=int)
        if not student_id:
            return jsonify({"error": "Brak wymaganego parametru 'id_studenta'."}), 400

        start_date_str = request.args.get('data_początkowa')
        end_date_str = request.args.get('data_końcowa')
        if not start_date_str or not end_date_str:
            return jsonify({"error": "Brak wymaganych parametrów 'data_początkowa' lub 'data_końcowa'."}), 400

        try:
            start_date = datetime.strptime(start_date_str, '%Y-%m-%d')
            end_date = datetime.strptime(end_date_str, '%Y-%m-%d')
        except ValueError:
            return jsonify({"error": "Niepoprawny format daty. Użyj formatu 'YYYY-MM-DD'."}), 400

        # Sprawdzenie, czy student istnieje
        student = Student.query.get(student_id)
        if not student:
            return jsonify({"error": f"Student o Id {student_id} nie istnieje."}), 404

```

Rys. 24: Implementacja endpointu /get-lessons

```

        # Pobranie lekcji w podanym zakresie dat
        lessons = Lesson.query.filter(
            Lesson.student_id == student_id,
            Lesson.date.between(start_date, end_date)
        ).all()

        # Przygotowanie danych do odpowiedzi
        lesson_data = [
            {
                "data": lesson.date.strftime('%Y-%m-%d'),
                "godzina": lesson.time.strftime('%H:%M'),
                "id_nauczyciela": lesson.teacher.id,
                "imie_nauczyciela": lesson.teacher.first_name,
                "nazwisko_nauczyciela": lesson.teacher.last_name,
                "przedmiot": lesson.subject.name
            }
            for lesson in lessons
        ]

        return jsonify({"lekcje": lesson_data}), 200

    except Exception as e:
        return jsonify({"error": f"Wystąpił nieoczekiwany błąd: {str(e)}"}), 500

```

Rys. 25: Implementacja endpointu /get-lessons c.d.

```

PS C:\Users\j-chi\Documents\6_semestr\ARIUS\zad3\flask_env\Scripts> python test.py

Test /get-lessons (pozytywny):
{
  "lekcje": [
    {
      "data": "2025-04-12",
      "godzina": "09:00",
      "id_nauczyciela": 3,
      "imie_nauczyciela": "Piotr",
      "nazwisko_nauczyciela": "Wi\u015bniewski",
      "przedmiot": "matematyka"
    },
    {
      "data": "2025-04-20",
      "godzina": "16:00",
      "id_nauczyciela": 3,
      "imie_nauczyciela": "Piotr",
      "nazwisko_nauczyciela": "Wi\u015bniewski",
      "przedmiot": "biologia"
    }
  ]
}

Status code: 200

Test /get-lessons (negatywny):
{
  "error": "Student o Id 999 nie istnieje."
}

Status code: 404

```

Rys. 26: Wynik wywołania endpointu /get-lesson.

```

+ 3745... 8302.600697 127.0.0.1 127.0.0.1 HTTP 306 GET /get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30 HTTP/1.1
3745... 8302.600817 127.0.0.1 127.0.0.1 TCP 44 5000 → 59449 [ACK] Seq=1 Ack=263 Win=65024 Len=0
3745... 8302.610272 127.0.0.1 127.0.0.1 TCP 210 5000 → 59449 [PSH, ACK] Seq=1 Ack=263 Win=65024 Len=166 [TCP PDU reassembled in 374597]
3745... 8302.610360 127.0.0.1 127.0.0.1 TCP 44 59449 → 5000 [ACK] Seq=263 Ack=167 Win=65280 Len=0
+ 3745... 8302.610789 127.0.0.1 127.0.0.1 HTTP/JSON 483 HTTP/1.1 200 OK , JSON (application/json)

Frame 374593: 306 bytes on wire (2448 bits), 306 bytes captured (2448 bits) on interface \Device\NPF_{...} id 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 59449, Dst Port: 5000, Seq: 1, Ack: 1, Len: 262
Hypertext Transfer Protocol
  GET /get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30 HTTP/1.1\r\n
    Request Method: GET
    Request URI: /get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30
    Request URI Path: /get-lessons
    Request URI Query: id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30
      Request URI Query Parameter: id_studenta=3
      Request URI Query Parameter: data_pocz%C4%85tkowa=2025-04-01
      Request URI Query Parameter: data_ko%C5%84cowa=2025-04-30
    Request Version: HTTP/1.1
  Host: localhost:5000\r\n
  User-Agent: python-requests/2.32.3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept: */*\r\n
  Connection: keep-alive\r\n
  Authorization: Julia Chilczuk\r\n
  \r\n
[Response in frame: 374597]
[Full request URI: http://localhost:5000/get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30]

```

Rys. 27: Zapytanie HTTP - poprawne dane wejściowe.


```
3745... 8302.610360 127.0.0.1 127.0.0.1 TCP 44 59449 → 5000 [ACK] Seq=263 Ack=16/ Win=65280 Len=0
3745... 8302.610789 127.0.0.1 127.0.0.1 HTTP/JSON 483 HTTP/1.1 200 OK , JSON (application/json)

▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 59449, Seq: 167, Ack: 263, Len: 439
▶ [2 Reassembled TCP Segments (605 bytes): #374595(166), #374597(439)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 200 OK\r\n
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
    Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    Date: Wed, 23 Apr 2025 11:32:48 GMT\r\n
    Content-Type: application/json\r\n
  ▶ Content-Length: 439\r\n
  Connection: close\r\n
  \r\n
  [Request in frame: 374593]
  [Time since request: 0.010092000 seconds]
  [Request URI: /get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30]
  [Full request URI: http://localhost:5000/get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30]
  File Data: 439 bytes
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▼ Member: lekcje
      ▼ Array
        ▼ Object
          ▼ Member: data
            [Path with value: /lekcje/[]/data:2025-04-12]
            [Member with value: data:2025-04-12]
            String value: 2025-04-12
            Key: data
            [Path: /lekcje/[]/data]
            ▶ Member: godzina
            ▶ Member: id_nauczyciela
            ▶ Member: imie_nauczyciela
            ▶ Member: nazwisko_nauczyciela
            ▶ Member: przedmiot
          ▶ Object
            Key: lekcje
            [Path: /lekcje]
```

Rys. 28: Poprawna odpowiedź HTTP

```
3746... 8304.664153 127.0.0.1 127.0.0.1 HTTP 308 GET /get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30 HTTP/1.1
3746... 8304.664266 127.0.0.1 127.0.0.1 TCP 44 5000 → 59452 [ACK] Seq=1 Ack=265 Win=65024 Len=0
3746... 8304.668609 127.0.0.1 127.0.0.1 TCP 216 5000 → 59452 [PSH, ACK] Seq=1 Ack=265 Win=65024 Len=172 [TCP PDU reassembled in 374629]
3746... 8304.668701 127.0.0.1 127.0.0.1 TCP 44 59452 → 5000 [ACK] Seq=265 Ack=173 Win=65280 Len=0
3746... 8304.668794 127.0.0.1 127.0.0.1 HTTP/JSON 92 HTTP/1.1 404 NOT FOUND , JSON (application/json)
3746... 8304.668851 127.0.0.1 127.0.0.1 TCP 44 59452 → 5000 [ACK] Seq=265 Ack=221 Win=65280 Len=0

▶ Frame 374625: 308 bytes on wire (2464 bits), 308 bytes captured (2464 bits) on interface \Device\NPF_{...} id 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 59452, Dst Port: 5000, Seq: 1, Ack: 1, Len: 264
▼ Hypertext Transfer Protocol
  GET /get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30 HTTP/1.1\r\n
  Request Method: GET
  ▼ Request URI: /get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30
    Request URI Path: /get-lessons
    ▼ Request URI Query: id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30
      Request URI Query Parameter: id_studenta=999
      Request URI Query Parameter: data_pocz%C4%85tkowa=2025-04-01
      Request URI Query Parameter: data_ko%C5%84cowa=2025-04-30
    Request Version: HTTP/1.1
    Host: localhost:5000\r\n
    User-Agent: python-requests/2.32.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: */*\r\n
    Connection: keep-alive\r\n
    Authorization: Julia Chilczuk\r\n
    \r\n
  [Response in frame: 374629]
  [Full request URI: http://localhost:5000/get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30]
```

Rys. 29: Zapytanie HTTP - błędne dane wejściowe

```

3746... 8304.668794 127.0.0.1 127.0.0.1 TCP 44 59452 → 5000 [ACK] Seq=265 Ack=173 Win=65280 Len=0
3746... 8304.668794 127.0.0.1 127.0.0.1 HTTP/JSON 92 HTTP/1.1 404 NOT FOUND , JSON (application/json)
3746... 8304.668851 127.0.0.1 127.0.0.1 TCP 44 59452 → 5000 [ACK] Seq=265 Ack=221 Win=65280 Len=0

> Frame 374629: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 5000, Dst Port: 59452, Seq: 173, Ack: 265, Len: 48
> [2 Reassembled TCP Segments (220 bytes): #374627(172), #374629(48)]
> Hypertext Transfer Protocol
  > HTTP/1.1 404 NOT FOUND\r\n
    > Response Version: HTTP/1.1
    > Status Code: 404
    > [Status Code Description: Not Found]
    > Response Phrase: NOT FOUND
    > Server: Werkzeug/3.1.3 Python/3.13.0\r\n
    > Date: Wed, 23 Apr 2025 11:32:50 GMT\r\n
    > Content-Type: application/json\r\n
    > Content-Length: 48\r\n
    > Connection: close\r\n
    > \r\n
    > [Request in frame: 374625]
    > [Time since request: 0.004641000 seconds]
    > [Request URI: /get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30]
    > [Full request URI: http://localhost:5000/get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30]
    > File Data: 48 bytes
  > JavaScript Object Notation: application/json
    > Object
      > Member: error
        > [Path with value: /error:Student o Id 999 nie istnieje.]
        > [Member with value: error:Student o Id 999 nie istnieje.]
        > String value: Student o Id 999 nie istnieje.
        > Key: error
        > [Path: /error]

```

Rys. 30: Niepoprawna odpowiedź HTTP.

7. Wireshark

Dane wysyłane i odbierane przechwycone zostały w programie Wireshark, gdzie można było dokładnie przeanalizować pakiety. Załączony do sprawozdania plik `przechwycone_pakiety.pcapng` zawiera ruch przechwycony podczas testowania endpointów.

No.	Time	Source	Destination	Protocol	Length	Info
816	14.513694	127.0.0.1	127.0.0.1	HTTP	232	GET /teacher-list HTTP/1.1
820	14.528385	127.0.0.1	127.0.0.1	HTTP/JSON	886	HTTP/1.1 200 OK , JSON (application/json)
847	16.557597	127.0.0.1	127.0.0.1	HTTP	252	GET /teacher-details?id_nauczyciela=1 HTTP/1.1
851	16.580837	127.0.0.1	127.0.0.1	HTTP/JSON	378	HTTP/1.1 200 OK , JSON (application/json)
964	18.612763	127.0.0.1	127.0.0.1	HTTP	254	GET /teacher-details?id_nauczyciela=999 HTTP/1.1
968	18.622818	127.0.0.1	127.0.0.1	HTTP/JSON	95	HTTP/1.1 404 NOT FOUND , JSON (application/json)
999	20.543362	127.0.0.1	127.0.0.1	HTTP/JSON	156	POST /book-lesson HTTP/1.1 , JSON (application/json)
1003	20.686699	127.0.0.1	127.0.0.1	HTTP/JSON	103	HTTP/1.1 201 CREATED , JSON (application/json)
1034	22.741245	127.0.0.1	127.0.0.1	HTTP/JSON	156	POST /book-lesson HTTP/1.1 , JSON (application/json)
1038	22.749278	127.0.0.1	127.0.0.1	HTTP/JSON	102	HTTP/1.1 400 BAD REQUEST , JSON (application/json)
1065	24.799507	127.0.0.1	127.0.0.1	HTTP/JSON	156	POST /book-lesson HTTP/1.1 , JSON (application/json)
1069	24.807748	127.0.0.1	127.0.0.1	HTTP/JSON	92	HTTP/1.1 400 BAD REQUEST , JSON (application/json)
1184	26.839119	127.0.0.1	127.0.0.1	HTTP/JSON	153	POST /book-lesson HTTP/1.1 , JSON (application/json)
1188	26.842975	127.0.0.1	127.0.0.1	HTTP/JSON	95	HTTP/1.1 400 BAD REQUEST , JSON (application/json)
1455	28.894558	127.0.0.1	127.0.0.1	HTTP/JSON	314	POST /add-teacher HTTP/1.1 , JSON (application/json)
1459	28.941567	127.0.0.1	127.0.0.1	HTTP/JSON	70	HTTP/1.1 201 CREATED , JSON (application/json)
1490	31.003204	127.0.0.1	127.0.0.1	HTTP/JSON	81	POST /add-teacher HTTP/1.1 , JSON (application/json)
1494	31.010622	127.0.0.1	127.0.0.1	HTTP/JSON	91	HTTP/1.1 400 BAD REQUEST , JSON (application/json)
1523	33.084154	127.0.0.1	127.0.0.1	HTTP/JSON	315	POST /add-teacher HTTP/1.1 , JSON (application/json)
1529	33.091429	127.0.0.1	127.0.0.1	HTTP/JSON	97	HTTP/1.1 400 BAD REQUEST , JSON (application/json)
1718	35.135026	127.0.0.1	127.0.0.1	HTTP	306	GET /get-lessons?id_studenta=3&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30 HTTP/1.1
1722	35.142617	127.0.0.1	127.0.0.1	HTTP/JSON	483	HTTP/1.1 200 OK , JSON (application/json)
1987	37.188974	127.0.0.1	127.0.0.1	HTTP	308	GET /get-lessons?id_studenta=999&data_pocz%C4%85tkowa=2025-04-01&data_ko%C5%84cowa=2025-04-30 HTTP/1.1
1991	37.193847	127.0.0.1	127.0.0.1	HTTP/JSON	92	HTTP/1.1 404 NOT FOUND , JSON (application/json)

Rys. 31: Przechwycony ruch obserwowany w Wireshark.