

## Aplikacje backendowe - wykład 3

---



dr Artur Bartoszewski  
Uniwersytet Radomski

# {JSON}

JSON (JavaScript Object Notation) to format danych, który jest popularny w informatyce do reprezentowania struktury danych w sposób lekki, czytelny dla człowieka i łatwy do przetwarzania przez komputery.

JSON jest językiem niezależnym od platformy i stosowany jest w wielu dziedzinach, głównie do wymiany danych między różnymi aplikacjami i systemami.



## Zalety języka JSON

- ✓ JSON jest zapisem danych w formie tekstowej i jest czytelny dla człowieka. Składa się z par klucz-wartość i struktur danych, które są łatwe do zrozumienia.
- ✓ JSON jest lekki, co oznacza, że zajmuje mało miejsca w pamięci i jest efektywny w przesyłaniu danych przez sieć.
- ✓ JSON jest niezależny od platformy i może być używany w różnych językach programowania. Obsługiwany jest przez wiele języków programowania, w tym JavaScript, Python, PHP, Java, C#, i wiele innych.
- ✓ JSON jest wszechstronny i może reprezentować różne rodzaje danych, takie jak obiekty, tablice, liczby, ciągi znaków, wartości logiczne i null.
- ✓ JSON jest łatwy do przetwarzania przez programy komputerowe. Parsowanie JSON do struktury danych w języku programowania jest proste i często dostępne w standardowych bibliotekach.

- ✓ Obiekt JSON to zbiór par klucz-wartość, gdzie klucze są ciągami znaków, a wartości mogą być innymi obiektami, tablicami, liczbami, ciągami znaków, wartościami logicznymi lub null.
- ✓ Obiekt JSON jest ograniczony nawiasami klamrowymi { }
- ✓ Elementy obiektu lub tablicy są oddzielane przecinkami.

```
{  
  "klucz1": "wartość1",  
  "klucz2": 42,  
  "klucz3": true,  
  "klucz4": null  
}
```

Ciągi znaków są umieszczone w cudzysłowach (jedno- lub podwójnych) i zawierają tekst.

- ✓ Tablica JSON to uporządkowany zestaw wartości, które są indeksowane liczbami całkowitymi (indeksy zaczynają się od zera). Tablice mogą zawierać różne typy danych.
- ✓ Tablica JSON jest ograniczona nawiasami kwadratowymi []

```
[  
    "jabłko",  
    "banan",  
    "śliwka"  
]
```

Przykład danych JSON.

```
{
  "imie": "Jan",
  "nazwisko": "Kowalski",
  "wiek": 30,
  "adres": {
    "ulica": "Kwiatowa",
    "numer": "123",
    "miasto": "Warszawa"
  },
  "zainteresowania": ["sport", "muzyka", "podróże"]
}
```

Obiekt JSON, który reprezentuje dane osobowe. Jest to obiekt zawierający pięć par klucz-wartość:

- "imie" z wartością "Jan".
- "nazwisko" z wartością "Kowalski".
- "wiek" z wartością 30.
- "adres" zagnieżdżony obiekt JSON, który zawiera trzy pary klucz-wartość reprezentujące adres.
- "zainteresowania" z tablicą zawierającą trzy ciągi znaków.

# Eksport tablicy do formatu JSON



Funkcja `json_encode()`: Funkcja `json_encode()` pozwala na zamianę tablicy na format JSON, który jest przydatny do przesyłania danych między aplikacjami i serwerami. Następnie możesz użyć `echo` do wyświetlenia danych JSON.

```
<?php
    $indeksowanaTablica = array("jabłko", "banan", "śliwka");
    echo json_encode($indeksowanaTablica);
    // Wynik: ["jabłko","banan","śliwka"]
?>
```

## Import z formatu JSON do tablicy



Aby zaimportować dane z formatu JSON do tablicy w języku PHP, można użyć funkcji `json_decode()`. Funkcja `json_decode()` przekształca dane w formacie JSON na strukturę danych w PHP, taką jak tablica lub obiekt.

```
<?php
$jsonDane = '{"jabłko": 3, "banan": 5, "śliwka": 2}';
$tablica = json_decode($jsonDane, true);
// Parametr true oznacza, że dane będą przetwarzane jako tablica asocjacyjna

// Teraz możesz używać zmiennej $tablica jak zwykłej tablicy
echo $tablica["jabłko"]; // Wyświetli 3
echo $tablica["banan"];  // Wyświetli 5
echo $tablica["śliwka"]; // Wyświetli 2

?>
```



# Import z formatu JSON do tablicy




Jeśli nie podasz drugiego argumentu true, wynikiem będzie obiekt zamiast tablicy.

```
<?php
$jsonDane = '[
    {"nazwa": "jabłko", "ilosc": 3},
    {"nazwa": "banan", "ilosc": 5},
    {"nazwa": "śliwka", "ilosc": 2}
]';

$tablica = json_decode($jsonDane);

// Teraz $tablica zawiera tablicę obiektów
echo $tablica[0]->nazwa; // Wyświetli "jabłko"
echo $tablica[1]->ilosc; // Wyświetli 5
?>
```

Plik JSON  
zawierający tablicę  
obiektów



## Przykład:



Przykład – zakodowanie tablicy dwuwymiarowej do formatu JSON

```
<?php
$tablica = array(
    "owoce" => array("jabłko", "banan", "śliwka"),
    "warzywa" => array("marchew", "ziemniak", "pomidor")
);

json_encode($tablica);

header('Content-Type: application/json');
echo json_encode($tablica);
?>
```

Efekt:

```
1  {
2      "owoce": [
3          "jabłko",
4          "banan",
5          "śliwka"
6      ],
7      "warzywa": [
8          "marchew",
9          "ziemniak",
10         "pomidor"
11     ]
12 }
```

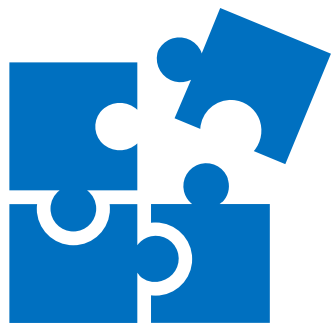
***header('Content-Type: application/json');***

Ustawia nagłówek HTTP odpowiedzi, informujący przeglądarkę, że treść odpowiedzi jest w formacie JSON. To umożliwia poprawne zinterpretowanie odpowiedzi przez klienta, który odbiera dane.



## Przykład do wykonania

---



Napisać który wyświetli aktualny kurs wybranych walut:



## Przykład do wykonania

W skrypcie skorzystamy z danych udostępnianych w formacie JSON przez Narodowy Bank Polski



NARODOWY  
BANK POLSKI

Przykładowo, dane o średnim kursie  
EURO znajdują się pod adresem:

[api.nbp.pl/api/exchangerates/rates/a/eur?format=json](https://api.nbp.pl/api/exchangerates/rates/a/eur?format=json)

```
1 {  
2   "table": "A",  
3   "currency": "euro",  
4   "code": "EUR",  
5   "rates": [  
6     {  
7       "no": "211/A/NBP/2023",  
8       "effectiveDate": "2023-10-31",  
9       "mid": 4.4475  
10    }  
11  ]  
12 }
```



## Przykład do wykonania

Skrypt pobiera plik spod wskazanego adresu i dekoduje na obiekt

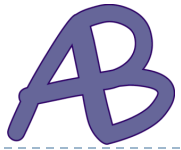
```
<?php
$url = 'http://api.nbp.pl/api/exchangerates/rates/a/eur?format=json';
$plikJSON = file_get_contents($url);
$tablicaDanych = json_decode($plikJSON); // dane będą przetwarzane jako tablica obiekt
print_r($tablicaDanych);
?>
```

```
1 {
2   "table": "A",
3   "currency": "euro",
4   "code": "EUR",
5   "rates": [
6     {
7       "no": "211/A/NBP/2023",
8       "effectiveDate": "2023-10-31",
9       "mid": 4.4475
10    }
11  ]
12 }
```

Wypisanie obiektu na ekran jest proste jednak aby odczytać wartość kursu euro należy przeanalizować strukturę otrzymanej odpowiedzi.

```
stdClass Object ( [table] => A [currency] => euro [code] => EUR [rates] => Array ( [0] => stdClass Object ( [no] => 211/A/NBP/2023 [effectiveDate] => 2023-10-31 [mid] => 4.4475 ) ) )
```

Tablica „rates” składa się z jednego pola „rates[0]” które zawiera obiekt (bez nazwy) zawierający trzy pola o nazwach „no”, „effectiveDate” i „mid”  
Pole „mid” przechowuje interesującą nas wartość - `rates[0]->mid`



## Przykład do wykonania

Wynikowy skrypt:

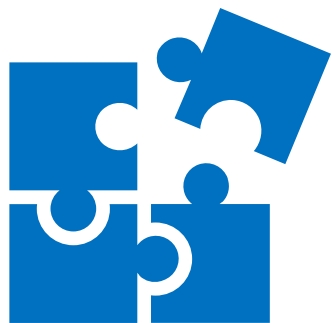
```
<?php
$url = 'http://api.nbp.pl/api/exchangerates/rates/a/eur?format=json';
$plikJSON = file_get_contents($url);
$tablicaDanych = json_decode($plikJSON); // dane będą przetwarzane jako tablica obiekt
$kurs = $tablicaDanych->rates[0]->mid;
$data = $tablicaDanych->rates[0]->effectiveDate;
echo "<p>Kurs euro z dnia: $data wynosi: $kurs PLN</p>";
?>
```

Kurs euro z dnia: 2023-10-31 wynosi: 4.4475 PLN



## Przykład do wykonania

---



Napisać skrypt który wyeksportuje tablicę dwuwymiarową do pliku JSON:

```
1 {  
2     "owoce": [  
3         "jabłko",  
4         "banan",  
5         "śliwka"  
6     ],  
7     "warzywa": [  
8         "marchew",  
9         "ziemniak",  
10        "pomidor"  
11    ]  
12 }
```

Część: 1 z 2



## Przykład do wykonania

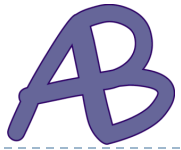
```
<?php
// Tablica wielowymiarowa
$wielowymiarowaTablica = array(
    "owoce" => array("jabłko", "banan", "śliwka"),
    "warzywa" => array("marchew", "ziemniak", "pomidor")
);

// Ścieżka do pliku, do którego chcemy zapisać dane
$filePath = 'data.json';

// Zapisz tablicę wielowymiarową do pliku w formie JSON
file_put_contents($filePath, json_encode($wielowymiarowaTablica));

// Odpowiedź z potwierdzeniem zapisu
$response = array('status' => 'Dane zapisane pomyślnie');
header('Content-Type: application/json');
echo json_encode($response);
?>
```





## Przykład do wykonania

---

W tym przypadku tablica wielowymiarowa \$wielowymiarowaTablica jest zapisywana do pliku data.json w formie JSON za pomocą funkcji json\_encode.

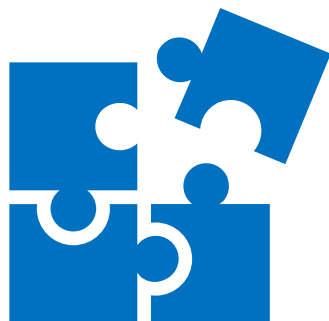
Skrypt zwraca również odpowiedź w formie JSON, potwierdzającą zapisanie danych.

Zawartość pliku wynikowego:

```
{"owoce":["jab\u0142ko","banan","\u015bliwka"],"warzywa":["marchew","ziemniak","pomidor']}
```



## Przykład do wykonania



Napisać skrypt który wczyta plik JSON i wyświetli jego zawartość w postaci tabeli

```
1 {  
2   "owoce": [  
3     "jabłko",  
4     "banan",  
5     "śliwka"  
6   ],  
7   "warzywa": [  
8     "marchew",  
9     "ziemniak",  
10    "pomidor"  
11  ]  
12 }
```



Kategoria	Elementy
owoce	jabłko, banan, śliwka
warzywa	marchew, ziemniak, pomidor

Część: 2 z 2



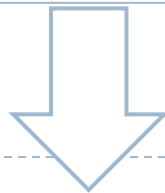
## Przykład do wykonania

```
<?php
// Ścieżka do pliku JSON
$filePath = 'data.json';

// Odczytaj zawartość pliku JSON
$jsonData = file_get_contents($filePath);

// Zdekoduj dane JSON do postaci tablicy asocjacyjnej
$dataArray = json_decode($jsonData, true);

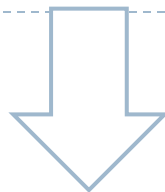
// Sprawdź, czy udało się zdekodować dane
if ($dataArray !== null) {
    // Wyświetl dane w postaci tabeli HTML
    echo '<table border="1">';
    echo '<tr><th>Kategoria</th><th>Elementy</th></tr>';
}
```





## Przykład do wykonania

---



```
foreach ($dataArray as $category => $items) {  
    echo '<tr>';  
    echo '<td>' . htmlspecialchars($category) . '</td>';  
    echo '<td>' . implode(', ', array_map('htmlspecialchars', $items)) . '</td>';  
    echo '</tr>';  
}  
  
    echo '</table>';  
} else {  
    echo 'Błąd dekodowania danych JSON';  
}  
?>
```



## Przykład do wykonania

---

Skrypt wykonuje następujące czynności:

1. Otwiera plik JSON i odczytuje jego zawartość za pomocą ***file\_get\_contents***.
2. Dekoduje zawartość JSON do postaci tablicy asocjacyjnej za pomocą ***json\_decode***.
3. Sprawdza, czy udało się zdekodować dane (czy wynik nie jest null).
4. Jeśli dane zostały zdekodowane poprawnie, skrypt generuje tabelę HTML, która wyświetla kategorie i elementy.
5. Jeśli wystąpił błąd podczas dekodowania, skrypt wyświetla komunikat o błędzie.



Funkcja **htmlspecialchars( )** służy do zabezpieczania treści przed atakami typu **Cross-Site Scripting (XSS)**, które mogą wystąpić, gdy niewłaściwie sformatowane dane są umieszczane w treści strony HTML. Funkcja ta zamienia znaki specjalne na ich odpowiedniki HTML, co uniemożliwia wykonanie kodu JavaScript wstrzykiwanego przez użytkownika.

W kontekście powyższego skryptu PHP, funkcja htmlspecialchars jest używana do zabezpieczenia danych przed umieszczeniem ich w treści tabeli HTML. Działa to w sposób następujący:

Znak **&** zamieniany jest na **&amp;**;

Znak **<** zamieniany jest na **&lt;**;

Znak **>** zamieniany jest na **&gt;**;

Znak **"** zamieniany jest na **&quot;**;

Znak **'** zamieniany jest na **&#039;**;

Dzięki tym zamianom, nawet jeśli dane zawierają znaki specjalne, zostaną one wyświetlone jako tekst, a nie jako fragmenty HTML, co pomaga zabezpieczyć stronę przed potencjalnymi atakami XSS.



## Literatura

---

W prezentacji użyto przykładów z książki:

- Żygłowicz Jerzy - PHP - Kompendium wiedzy, Helion

- <https://www.php.net>