

Programowanie współbieżne i rozproszone

WYKŁAD 8

dr inż. Krzysztof Pancierz

JSON

- JSON (JavaScript Object Notation):
 - samoopisujący się standard wymiany danych w aplikacjach internetowych,
 - sposób przekształcania obiektów JavaScript w łańcuch tekstowy i odwrotnie.
- Format JSON jest formatem tekstowym niezależnym od języka programowania.
- JSON oparty jest o:
 - zbiorów par "atrybut-wartość",
 - uporządkowane listy wartości.

JSON

- Zalety JSON w stosunku do XML:
 - JSON jest bardziej efektywny (używa mniejszej liczby znaków),
 - transformacje danych w formacie JSON na postać tekstową i odwrotnie jest szybsze niż przetwarzanie plików XML,
 - składnia JSON jest zgodna ze składnią języka JavaScript,
 - w JSON mogą być używane tablice.

JSON a XML

- Używanie XML jest wskazane w przypadkach złożonych obiektów oraz w sytuacjach gdy istnieją zdefiniowane transformacje XSLT.

JSON

- Wspólne cechy formatów JSON i XML:
 - są samoopisujące się,
 - są hierarchiczne,
 - mogą być parowane w programach napisanych w różnych językach,
 - mogą być przesyłane za pomocą XMLHttpRequest.

Składnia JSON

- Dane są parami "atrybut-wartość".
- Dane są odseparowane od siebie za pomocą przecinka.
- Nawiasy klamrowe { } obejmują obiekt.
- Nawiasy prostokątne [] obejmują tablice.

Składnia JSON

- Wartość JSON może być:
 - liczbą (całkowitą lub zmiennoprzecinkową),
 - łańcuchem znaków,
 - wartością logiczną (*true* albo *false*),
 - tablicą,
 - obiektem,
 - wartością pustą (*null*).

Dane w formacie JSON

- Przykład danych w formacie JSON:

```
{ "miasta": [  
  {"nazwa": "Rzeszów", "status": "wojewódzkie"},  
  {"nazwa": "Jasło", "status": "powiatowe"},  
  {"nazwa": "Krosno", "status": "powiatowe"}  
]}
```


Dane w formacie JSON

- Obiekty i tablice:

```
{ "miasta":
```

```
[  
  {"nazwa": "Rzeszów", "status": "wojewódzkie"},  
  {"nazwa": "Jasło", "status": "powiatowe"},  
  {"nazwa": "Krosno", "status": "powiatowe"}  
]  
}
```

Tablica



Obiekt



Dane w formacie XML

- Przykład danych w formacie XML:

```
<miasta>  
  <miasto><nazwa>Rzeszów</nazwa>  
  <status>wojewódzkie</status></miasto>  
  <miasto><nazwa>Jasło</nazwa>  
  <status>powiatowe</status></miasto>  
  <miasto><nazwa>Krosno</nazwa>  
  <status>powiatowe</status></miasto>  
</miasta>
```

Pliki JSON

- Dla plików JSON używane jest rozszerzenie ".json".
- Content-Type dla JSON to "application/json".

API w języku Java dla JSON

Pakiet javax.json:

<https://docs.oracle.com/javase/7/api/javax/json/package-summary.html>

API w języku Java dla JSON

- Wybrane klasy i interfejsy:
 - **Json** - wykorzystanie: tworzenie obiektów przetwarzania JSON

```
StringReader reader = new StringReader(lancuch);  
JsonParser parser = Json.createParser(reader);
```

API w języku Java dla JSON

- Metoda `next()` pozwala na parsowanie zdarzeń w wyspecyfikowanych lokalizacjach, np.:

```
{START_OBJECT
  "imie"KEY_NAME: "Jan"VALUE_STRING, "nazwisko"KEY_NAME: "Kowalski"VALUE_STRING,
  "numer_indeksu"KEY_NAME: 99999VALUE_NUMBER,
  "indeks"KEY_NAME : [START_ARRAY
    {START_OBJECT "przedmiot"KEY_NAME: "matematyka"VALUE_STRING, "ocena"KEY_NAME:
    "3.5"VALUE_STRING }END_OBJECT,
    {START_OBJECT "przedmiot"KEY_NAME: "fizyka"VALUE_STRING, "ocena"KEY_NAME:
    "4.0"VALUE_STRING }END_OBJECT
  ]END_ARRAY
}END_OBJECT
```

```
Event zdarzenie = parser.next(); // START_OBJECT
zdarzenie = parser.next(); // KEY_NAME
zdarzenie = parser.next(); // VALUE_STRING
zdarzenie.getString(); // "Jan"
```

API w języku Java dla JSON

- Wybrane klasy i interfejsy:
 - **JsonReader** - wykorzystanie: odczyt JSON
 - **JsonArrayBuilder** - wykorzystanie: tworzenie tablicy JSON
 - **JsonWriter** – wykorzystanie: zapis JSON

API w języku Java dla JSON

- Wybrane klasy i interfejsy (c.d.):
 - **JsonValue** - wykorzystanie: reprezentacja wartości w JSON
 - **JsonObject** - wykorzystanie: reprezentacja obiektu w JSON
 - **JsonArray** - wykorzystanie: reprezentacja tablicy w JSON
 - **JsonString** - wykorzystanie: reprezentacja łańcucha znaków w JSON
 - **JsonNumber** - wykorzystanie: reprezentacja liczby w JSON