

Projektowanie aplikacji biznesowych (PAB) – laboratorium nr 2

JSON, YAML

Początek laboratorium:

- na pulpicie utworzyć folder lab2,
- otworzyć go w *Visual Studio Code* lub dowolnym IDE od *JetBrains* (np. *PHPStorm*, *IntelliJ*, *PyCharm* itp.).

Zadania (JSON):

- specyfikacja: <https://datatracker.ietf.org/doc/html/rfc8259>
- zastosowania JSON (*JSON use cases*):
<https://www.oracle.com/pl/database/what-is-json/#json-use-cases>
- poprawność składniowa dokumentu,
<https://www.techiedelight.com/json-introduction/> → *JSON Syntax Rules*
<https://jsonlint.com/> → *Proper JSON Format*
- poprawność strukturalna dokumentu, *JSON Schema*:
<https://szkoladevnet.pl/po-co-nam-json-schema/>
<https://json-schema.org/>
<https://json-schema.org/understanding-json-schema/>
<https://json-schema.org/learn/getting-started-step-by-step.html>
- przykłady *JSON Schema*:
<https://json-schema.org/learn/miscellaneous-examples>

Zadanie 2.1:

Zapoznać się z następującymi zagadnieniami dotyczącymi formatu JSON:

- składnia JSON,
- JSON vs XML,
- typy danych zawartości JSON'a,
- nazewnictwo kluczy: *camelCase* vs *snake_case*,
- kwestia unikalności nazw kluczy.

<https://www.json.org/json-en.html>

<https://www.geeksforgeeks.org/difference-between-json-and-xml/>

<https://www.techiedelight.com/json-introduction/>

https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf#page=11

Zadanie 2.2:

Pobrać plik *JSON* o nazwie *zad2.json* i otworzyć go w przeglądarce *Firefox*.
Następnie dokonać edycji pliku: poprawić wszystkie błędy składniowe.

–

Zadanie 2.3:

Utworzyć przykładowy plik *JSON* o nazwie *rozkład.json* odnośnie *rozkładu jazdy pociągów*, według wymagań:

Dla rozkładu jazdy pociągów powinna być dostępna informacja o okresie obowiązywania (data początkowa i data końcowa).

Dla każdego pociągu w rozkładzie jazdy powinna być dostępna informacja o:

- numerze pociągu (liczba całkowita),
- nazwie przewoźnika (maks. 15 znaków),
- rodzaju pociągu (osobowy, pociąg ekspres),
- opcjonalnej nazwie pociągu,
- czy pociąg jest regionalny (tak/nie/brak informacji),
- stacjach (min. 2), na których zatrzymuje się pociąg wraz z ich nazwami, godzinami przyjazdu i odjazdu tego pociągu,
- zestawieniu 1-5 wagonów (wagony klasy 2, wagony klasy 1, wagony sypialne, wagony z miejscami do leżenia, wagon restauracyjny).

–

Zadanie 2.4:

Otworzyć plik *rozkład.json* w przeglądarce internetowej w celu dodatkowego sprawdzenia czy plik się prawidłowo *parsuje*.

–

Zadanie 2.5:

Zapoznać się z podstawowymi zagadnieniami dotyczącymi *JSON Schema*:

- rozpoczęcie tworzenia schematu,
- definicja właściwości (*properties*), typy danych (*type*),
- wymaganie właściwości (*required*),
- formaty ciągów znaków (*format*),
- minimalna liczba wystąpień (*minItems*),
- dane zagnieżdżone (*nested data...*),
- tablice wartości, obiektów (*array of things...*),
- liczba elementów tablicy (*minItems*, *maxItems*),
- lista wartości (*enumerated values*),
- możliwość wartości *null*.

<https://json-schema.org/learn/getting-started-step-by-step#create-a-schema-definition>

<https://json-schema.org/learn/getting-started-step-by-step#define>

<https://json-schema.org/learn/getting-started-step-by-step#required>

<https://json-schema.org/understanding-json-schema/reference/string#format>

<https://json-schema.org/learn/getting-started-step-by-step#define-optional-properties>

<https://json-schema.org/learn/getting-started-step-by-step#nest-data>

<https://json-schema.org/learn/miscellaneous-examples#arrays-of-things>

<https://json-schema.org/understanding-json-schema/reference/array#length>

<https://json-schema.org/learn/miscellaneous-examples#enumerated-values>

<https://www.serverless360.com/blog/specifying-json-schema-elements-null-in-logic-apps>

Zadanie 2.6:

Wypróbować narzędzie *online* do generowania *JSON Schema* dla wcześniej utworzonego *rozklad.json*.

Utworzyć plik *schemat.json* i wkleić do niego wstępnie wygenerowaną *schema*.

<https://transform.tools/json-to-json-schema>

Zadanie 2.7:

Powrócić do pliku *rozklad.json* i dodać w 2 linijce odwołanie do *JSON Schema*.

```
"$schema": "./schemat.json",
```

```
rozklad.json > ...
1  {
2    "$schema": "./schemat.json",
3    "od": "2024-10-01",
4    "do": "2024-12-31",
5    "pociagi": [
6      {
```

Zadanie 2.8:

W pliku *schemat.json* uzupełnić *definicję struktury* dla plików *JSON* mających przechowywać informacje o rozkładzie jazdy pociągów tak, żeby obowiązywały wszystkie zasady ustalone w zadaniu 2.3.

–

Zadanie 2.9:

Powrócić do pliku *rozklad.json* i wypróbować sprawdzanie poprawności *strukturalnej*.

Przykład:

```
rozklad.json 1 ●
rozklad.json > [abc]
1  {
2    "$sche
3    "od":
4    "do": "2024-13-31",
5    "pociagi": [
6      {
7        "nr": 1.
```

String is not a RFC3339 date.
Data końcowa obowiązywania rozkładu
View Problem No quick fixes

Zadanie 2.10:

Wypróbować narzędzie *online* do sprawdzania poprawności *strukturalnej*.

<https://www.jsonschemavalidator.net/>

Zadanie 2.11: *

Zapoznać się z przykładami zastosowania formatu *JSON*.

<https://getcomposer.org/doc/04-schema.md>

<https://getcomposer.org/schema.json>

<https://docs.npmjs.com/cli/v10/configuring-npm/package-json>

<https://github.com/SchemaStore/schemastore/blob/master/src/schemas/json/package.json>

<https://mariadb.com/resources/blog/using-json-in-mariadb/>

Zadania (YAML):

- specyfikacja: <https://yaml.org/>
- https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html
- YAML syntax example, YAML vs JSON, zastosowania YAML:
<https://www.redhat.com/en/topics/automation/what-is-yaml>
- typy danych YAML: <https://learn.getgrav.org/17/advanced/yaml#basic-data-types>
- JSON to YAML converter: <https://www.bairesdev.com/tools/json2yaml>
- YAML Schema: https://asdf-standard.readthedocs.io/en/1.1.1/schemas/yaml_schema.html

Zadanie 2.12: *

Zapoznać się z następującymi zagadnieniami dotyczącymi formatu YAML:

- składnia YAML, komentarze,
- listy, elementy list, słowniki,
- typy danych zawartości YAML'a, możliwe wartości *Boolean*,
- wieloliniowe wartości, podział | i >.

Napisany wcześniej JSON z rozkładem jazdy pociągów przekonwertować na YAML za pomocą poniższego narzędzia online.

Wskazać wady i zalety obu formatów.

<https://www.bairesdev.com/tools/json2yaml>

Zadanie 2.13: *

Zapoznać się z przykładami zastosowania formatu YAML.

<https://docs.docker.com/compose/intro/compose-application-model/#illustrative-example>

<https://docs.docker.com/engine/storage/volumes/#use-a-volume-with-docker-compose>

<https://docs.spring.io/spring-boot/docs/current/reference/html/application-properties.html>

<https://dev.to/fullstackhacker/spring-boot-yaml-example-1kb2>

* – zadania/podpunkty do samodzielnego dokończenia/wykonania,

* – zadania/podpunkty dla zainteresowanych.