Projektowanie aplikacji biznesowych (PAB) – laboratorium nr 2

JSON (JavaScript Object Notation)

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

• *JSON Syntax Rules*:

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

• *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.w3schools.com/js/js_json_xml.asp 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*.
Poprawić wszystkie błędy składniowe.

–

Zadanie 2.3:
Utworzyć przykładowy plik *JSON* o nazwie **`rozklad.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, pospieszny, ekspres),
• opcjonalnej nazwie pociągu,
• czy pociąg jest regionalny (tak/nie/brak informacji),
• nazwach stacji/przystanków (min. 2), na których zatrzymuje się pociąg wraz z godzinami przyjazdu i odjazdu,
• zestawieniu 1-5 wagonów (wagony klasy 2, wagony klasy 1, wagony sypialne, wagony z miejscami do leżenia, wagon restauracyjny).

–

Zadanie 2.4:
Otworzyć plik **`rozklad.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*.

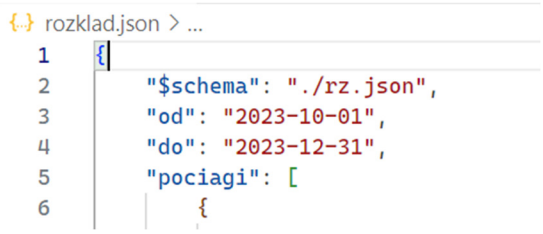
https://json-schema.org/learn/getting-started-step-by-step#create-a-schema-definition
https://json-schema.org/learn/getting-started-step-by-step#define
https://json-schema.org/learn/getting-started-step-by-step#required
https://json-schema.org/understanding-json-schema/reference/string#format
https://json-schema.org/learn/getting-started-step-by-step#define-optional-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:
Utworzyć plik **rz.json**. Wkleić do niego poniższy fragment.

```
{
    "title": "Rozkład",
    "description": "Rozkład jazdy pociągów ...",
    "type": "object"
}
```

Zadanie 2.7:
Powrócić do pliku **rozklad.json** i dodać w 2 linijce odwołanie do *JSON Schemy*.

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

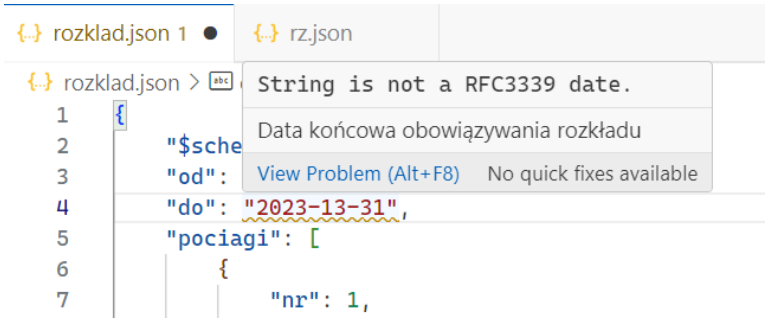


Zadanie 2.8:
W pliku **rz.json** napisać *definicję struktury* dla plików *JSON* mających przechowywać informacje o rozkładzie jazdy pociągów tak, żeby obowiązywały wszystkie zasady ustalone w zadaniu 2.3.

–

Zadanie 2.9:
Powrócić do pliku **rozklad.json** i wypróbować sprawdzanie poprawności *strukturalnej*.

Przykład:



Zadanie 2.10:
Wypróbować narzędzie online do generowania JSON Schema.
Wypróbować również sprawdzanie poprawności *strukturalnej*.

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

https://www.jsonschemavalidator.net/

* – zadania/podpunkty do samodzielnego dokończenia/wykonania,

* – zadania/podpunkty dla zainteresowanych.

Po zakończonym laboratorium należy skasować wszystkie pobrane oraz utworzone przez siebie pliki z komputera w sali laboratoryjnej.

Wersja pliku: v1.0

Inne:*

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/