МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.16

Работа с данными формата JSON в языке Python

по дисциплине «Технологии программирования и алгоритмизация»

Выполнил
студент группы ИВТ-б-о-20-1
Дыбов Д.В. « »_____20__г.
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2021

Цель работы: приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x

Ход работы

1. Создал новый репозиторий для лабораторной работы №2.16;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. Проработал первый пример;
5. Проверил пример на работоспособность:

```
>>> add
Фамилия и инициалы? Овсиенко И.М.
Должность? Психолог
Год поступления? 2004
>>> add
Фамилия и инициалы? Сапогов Г.Е.
Должность? Секретарь
Год поступления? 2006
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Овсиенко И.М.                    | Психолог             |     2004 |
|    2 | Сапогов Г.Е.                     | Секретарь            |     2006 |
+------+----------------------------------+----------------------+----------+
>>> save job.json
```

Рисунок 1 – Результат выполнения первого примера

6. Сохранил данные в json файл

Рисунок 2 – Результат выгрузки данных в json файл

7. Загрузил данные из json файла

```
>>> list
Список работников пуст.
>>> load job.json
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Овсиенко И.М.                    | Психолог             |     2004 |
|    2 | Сапогов Г.Е.                     | Секретарь            |     2006 |
+------+----------------------------------+----------------------+----------+
>>>
```

Рисунок 3 – Загрузка данных json

8. Проработал индивидуальное задание;
9. Проверил его на работоспособность:

```
>>> add
Пункт назначения? Москва
Номер рейса? 12
Тип самолёта? 7524
>>> add
Пункт назначения? Санкт-Петербург
Номер рейса? 4
Тип самолёта? 8734
>>> list
+------+-------------------------------+----------------------+----------------------+
|  №   |        Пункт назначения       |     Номер рейса      |     Тип самолёта     |
+------+-------------------------------+----------------------+----------------------+
|    1 | Москва                        | 12                   |                 7524 |
|    2 | Санкт-Петербург               | 4                    |                 8734 |
+------+-------------------------------+----------------------+----------------------+
>>> save races1.json
>>>
```

Рисунок 4 – Результат выполнения индивидуального задания

10. Сохранил данные в json файл

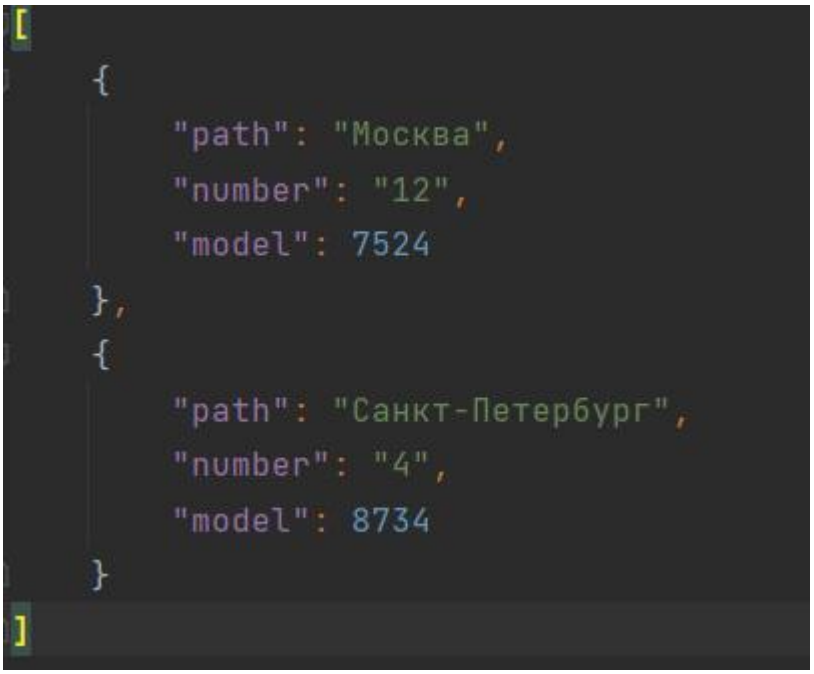

Рисунок 5 – Результат выгрузки данных в json файл

11. Загрузил данные из json файла

```
>>> list
Список работников пуст.
>>> load job.json
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Овсиенко И.М.                    | Психолог             |     2004 |
|    2 | Сапогов Г.Е.                     | Секретарь            |     2006 |
+------+----------------------------------+----------------------+----------+
>>>
```

Рисунок 6 – Загрузка данных json

12. Выполнил задание повышенной сложности;
13. Проверил задание на работоспособность:

```
>>> add
Пункт назначения Омск
Номер рейса 3
Тип самолёта 5479
>>> add
Пункт назначения Воронеж
Номер рейса 1
Тип самолёта 5768
>>> list
+------+--------------------------------+----------------------+----------------------+
|  №   |        Пункт назначения        |     Номер рейса      |     Тип самолёта     |
+------+--------------------------------+----------------------+----------------------+
|    1 | Воронеж                        | 1                    |                 5768 |
|    2 | Омск                           | 3                    |                 5479 |
+------+--------------------------------+----------------------+----------------------+
>>> save races2.json
```

Рисунок 4 – Результат выполнения индивидуального задания

12. С помощью сайта проверил пример на наличие ошибок;
13. Результат не выдал ошибок;

Рисунок 5 – Проверка кода на наличие ошибок

14. С помощью сайта проверил индивидуальное задание на наличие ошибок;

15. Результат не выдал ошибок;

Рисунок 6 – Проверка индивидуального задания на наличие ошибок

16. С помощью сайта проверил задание повышенной сложности на наличие ошибок;

17. Результат не выдал ошибок;

Python code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import json
import jsonschema


def get_airplane():
    path = input("Пункт назначения ")
    number = input("Номер рейса ")
    model = input("Тип самолёта ")

```

No syntax errors detected :)

Рисунок 7 – Проверка задание на наличие ошибок

18. Отправил все изменения на репозиторий.

Контрольные вопросы

1. Для чего используется JSON?

– JSON используется для обмена данными, которые являются структурированными и хранятся в файле или в строке кода.

2. Какие типы значений используются в JSON?

– string;

– number;

– object;

– array;

– boolean;

– null.

3. Как организована работа со сложными данными в JSON?

Данные могут быть вложены в формате JSON, используя JavaScript массивы, которые передаются как значения. При помощи вложенных массивов и объектов можно создать сложную иерархию данных.

4. Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?

Формат обмена данными JSON5 (JSON5) — это надмножество JSON, которое направлено на смягчение некоторых ограничений JSON путем расширения его синтаксиса для включения некоторых продуктов из ECMAScript 5.1.

JSON5 получил следующие новшества:

– строки могут охватывать несколько строк, экранируя новые символы строк;

– числа могут быть шестнадцатеричными;

– допускаются однострочные и многострочные комментарии;

– ключи объектов могут быть без кавычек, если они являются законными идентификаторами ECMAScript;

– объекты и массивы могут заканчиваться запятыми в конце.

Существует одно заметное отличие от JSON: методы load() и loads() поддерживают выборочную проверку (и отклонение) дубликатов ключей объектов.

5. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?

– json5.load();

– json5.loads();

– json5.tool();

– json5.dump();

– json5.dumps().

6. Какие средства предоставляет язык Python для сериализации данных в формате JSON?

– Процесс кодирования данных в необходимый формат называется сериализацией. Для того чтобы записать эти данные в файл с форматом JSON в Python, используются функция dump() и dumps().

7. В чем отличие функций json.dump() и json.dumps()?

Dump отличается от dumps тем, что dump записывает объект Python в файл JSON, а dumps сериализует объект Python и хранит его в виде строки.

8. Какие средства предоставляет язык Python для десериализации данных из формата JSON?

Когда есть файл JSON, который необходимо преобразовать в объект Python, тогда проводится десериализация. Для десериализации по аналогии используются две функции: load() и loads().

9. Какие средства необходимо использовать для работы с данными формата JSON, содержащими кирилицу?

При записи достаточно передать ensure_ascii=False, чтобы не экранировать не-ascii символы.

10. Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных? Приведите схему данных для примера 1.

Схема JSON – это словарь, который позволяет аннотировать и проверять документы JSON.

Преимущества:

– описывает ваш существующий формат(ы) данных;

– обеспечивает четкую читаемую документацию для человека и машины;

– проверяет данные, которые полезны для автоматизированного тестирования и обеспечения качества предоставляемых клиентом данных.

Пример схемы.

Schema = {

“type”: “object”,

```
“employees”: {
“name”: {“type”: “string”},
“post”: {“type”: “string”},
“year”: {“type”: “string”,
“format”: “date”}}
}
```

Вывод: в ходе занятия были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.