Открытые данные

Краткий обзор технологий для понимания сбора и обработки данных

[Что такое Open data и кто их применяет](#_b0ldjtrqoek8)

[Принципы и методы построения открытых данных](#_tfs73tc4mmu6)

[Работа с Open data в Python](#_fzf7kd6ixhbn)

[Домашнее задание](#_cqknngnd5zt)

[Используемая литература](#_uvp6qax5r1ok)

# 

# Что такое Open data и кто их применяет

Концепция открытых данных основана на идее, что некоторые данные должны быть открытыми, машиночитаемыми и доступными для последующего использования без каких-либо ограничений. Сейчас основной поставщик таких данных – государство, и отражается это в особом документе – Методических рекомендациях по публикации открытых данных. В общем смысле открытые данные – это часть информации определенного рода, раскрываемой органами государственной власти и местного самоуправления. Двигатель внедрения открытых данных в России – проект «Открытые данные». Предполагается, что на этом фундаменте должна будет вырасти система открытого правительства Открытые данные имеют хороший потенциал как индикатор отображения различных показателей на государственном уровне.

Примеры открытых данных:

* Государственный реестр лекарственных средств;
* результаты голосований на уровне поселка/города/страны;
* данные вызовов экстренных служб по месяцам;
* список католических храмов в России;
* список зарегистрированных WiFi-точек в городе.

На основе таких данных можно построить множество отчетов на разных уровнях, подсчитать «индекс Биг Мака» в разных странах, отразить «пульс» города или страны, выявить проблемы и понять, как можно сделать жизнь в них лучше.

Вот несколько известных источников открытых данных:

* <https://proverki.gov.ru/opendata/>
* <http://data.gov.ru>
* <http://open.gov.ru>
* <http://data.mos.ru>
* <https://catalog.data.gov/dataset?res_format=CSV>

Типичные проблемы открытых данных – невалидность и разрозненность данных. Причины этого – в том, что данных огромное количество, они очень разнородны, а методические рекомендации по подготовке и выдаче открытых данных соблюдаются редко.

Пример плохого CSV-файла открытых данных:

|  |
| --- |
| regnumber,regdate,enddate,cancellationdate,nameregcertificate,country,tradename,internationalname,formrelease,stages,barcodes,normativedocumentation,pharmacotherapeuticgroup  ,,,,"ООО ""Валеант""",Россия,,,,"Производство готовой лекарственной формы,Херкель Б.В., Nobelweg 6, 3899 BN Zeewolde, the Netherlands, Нидерланды  ",,"П N009886-280411,2011,Бронхинол;  ",отхаркивающее средство растительного происхождения  ,,21.10.2010,,Дарница фармацевтическая фирма ЗАО,Украина,,,"таблетки 200 мг, упаковки ячейковые контурные - 2  ","Производитель (Все стадии производства),Дарница фармацевтическая фирма ЗАО, ~, Украина  ","4823006400096,Дарница фармацевтическая фирма ЗАО,пред-ль в Москве ОАО ""Бофарм"": 141191, Московская обл., г. Фрязино, ул. Станционная, д. 2, корп. 1,Украина  ","НД 42-7644-04,2004,ПЕНТОКСИФИЛЛИН-ДАРНИЦА;  ",вазодилатирующее средство  ,,21.10.2010,,Дарница фармацевтическая фирма ЗАО,Украина,,,"концентрат для приготовления раствора для внутривенного и внутриартериального введения 20 мг/мл, ампулы - 5  ","Производитель (Все стадии производства),Дарница фармацевтическая фирма ЗАО, ~, Украина  ","4823006400256,Дарница фармацевтическая фирма ЗАО,пред-ль в Москве ОАО ""Бофарм"": 141191, Московская обл., г. Фрязино, ул. Станционная, д. 2, корп. 1,Украина  ","НД 42-7644-04,2004,ПЕНТОКСИФИЛЛИН-ДАРНИЦА;  ",вазодилатирующее средство  ,,16.09.2010,01.10.2010,Галичфарм АО,Украина,,,"раствор для внутривенного введения 0.25 мг/мл, ампулы - 10  ","Производитель (Все стадии производства),Галичфарм АО, г. Львов, ул. Опрышковская, 6/8, Украина  ","4823000800250,Галичфарм АО,г. Львов, ул. Опрышковская, 6/8,Украина  ","НД 42-9049-05,2005,СТРОФАНТИН К;  ",кардиотоническое средство - сердечный гликозид  ,,,,"АО ""Галичфарм""",Украина,,,"раствор для внутривенного и внутримышечного введения 0.2500 мг/мл, ампулы из прозрачного бесцветного стекла - 10  ","Все стадии,АО ""Галичфарм"", , Украина  ","4823000800250,АО ""Галичфарм"",,Украина  ","010675-200911,2011,;  ",  ,,,12.03.2008,Московская фармацевтическая фабрика ОАО,Россия,,,"субстанция-настойка гомеопатическая матричная ~, флаконы темного стекла -  ","Производитель (Все стадии производства),Московская фармацевтическая фабрика ОАО, ~, Россия  ",,"ВФС 42-3628-00,2000,НАСТОЙКА ЗВЕРОБОЯ ГОМЕОПАТИЧЕСКАЯ (""ГИПЕРИКУМ 1 Д"");  ",гомеопатическое монокомпонентное средство  ,,,,Опытный завод АН Республики Башкортостан,Россия,,,"субстанция-порошок ~, пакеты полиэтиленовые двухслойные -  ",,"~,Опытный завод АН Республики Башкортостан,~,Россия  ",, |

# Принципы и методы построения открытых данных

По статистике, более 70% открытых данных размещены в формате CSV (Comma-Separated Values – значения, разделенные запятыми), при том что машиночитаемыми могут быть данные в следующих форматах:

* CSV;
* XML;
* JSON;
* HTML+RDFa;
* HTML+Microdata.

CSV – это текстовый формат, предназначенный в основном для табличных данных, так как в нем каждая строка текстового файла представляет собой строку в таблице. Разделителем по умолчанию является запятая, но на практике очень часто можно встретить и другие символы в качестве разделителей (точку сзапятой, двоеточие, табуляцию). Пример файла формата CSV:

|  |
| --- |
| id, level,phrase1,phrase2,numcode  90,11,“hello”,“world”,1  91,235,“si\”3\”sta”,“kingdom”,300  92,0,“couch”,,84 |

Первая строка – это обозначение колонок. Далее на каждой строчке следуют значения через разделитель (в данном случае это запятая). Все строки заключены в двойные кавычки (если такие же кавычки встречаются в тексте, их необходимо экранировать, как это сделано в записи с id=91 в колонке phrase1). Если значение отсутствует, то в строке будет пустое место (не пробел), за которым обязательно должен следовать разделитель (иначе нарушится структура), как это видно в строке id=92: в колонке phrase2 нет значения.

К открытым данным предъявляются следующие требования:

* портал открытых данных должен быть доступен по следующему адресу: <http://domain.com/opendata>;
* набор открытых данных включает в себя сами открытые данные (файл) и метаинформацию, состоящую из:
  + паспорта набора;
  + структуры набора (csv, xsd, json-schema);
  + иной информации, описывающей набор открытых данных, куда входят:
    - статическая информация;
    - лицензия;
    - человеческое отображение и машиночитаемое представление данных;
* реестр (т. е. список) наборов открытых данных должен иметь машиночитаемое представление и быть доступным по адресу <http://domain.com/opendata/list.xml> (в случае, если для формата данных выбран XML).

Паспорт открытых данных имеет четкую структуру:

1. Идентификационный номер.
2. Наименование набора открытых данных.
3. Описание набора открытых данных.
4. Владелец набора открытых данных.
5. Ответственное лицо.
6. Телефон ответственного лица.
7. Адрес электронной почты ответственного лица.
8. Гиперссылка (URL) на открытые данные.
9. Формат набора открытых данных.
10. Описание структуры набора открытых данных.
11. Дата первой публикации набора открытых данных.
12. Дата последнего внесения изменений.
13. Содержание последнего изменения.
14. Дата актуальности набора данных.
15. Ключевые слова, соответствующие содержанию набора данных.
16. Гиперссылки (URL) на версии открытых данных.
17. Гиперссылки (URL) на версии структуры набора данных.
18. Версия методических рекомендаций.

Идентификационный номер формируется из следующих параметров:

* формат идентификационного номера: <код организации>-<наименование набора>;
* код организации представляет собой идентификационный номер налогоплательщика (ИНН), соответствующий государственному органу, органу местного самоуправления или организации, опубликовавшей набор открытых данных;
* наименование набора открытых данных – сокращенное англоязычное название набора открытых данных, указывается в одно слово (уникальное в пределах организации).

Пример: 7712345678-inspections.

Наборы открытых данных могут быть оперативными (обновляться чаще, чем 1 раз в неделю) и долговременными (реже, чем 1 раз в неделю).

# Работа с Open data в Python

Так как большая часть Open data представлена в формате CSV, мы будем работать в Python с этим форматом. Для работы с CSV в стандартной поставке Python есть библиотека csv.

Файл формата CSV с которым будем работать:

|  |
| --- |
| "Grade","Ounce","Gram","Inch","mm","PPO"  "#TriBall",0.7199,20.41,0.60,15.24,1  "#0000",0.1943,5.51,0.38,9.40,5  "#000",0.1601,4.54,0.36,9.14,6  "#00",0.1231,3.49,0.33,8.38,8  "#0",0.1122,3.18,0.32,8.13,9  "#1",0.0924,2.62,0.30,7.62,11  "#2",0.0674,1.91,0.27,6.86,15  "#3",0.0536,1.52,0.25,6.35,19  "#4",0.0473,1.34,0.24,6.09,21  "#FF",0.0416,1.18,0.23,5.84,24  "#F",0.0370,1.05,0.22,5.59,27  "#TT",0.0346,0.98,0.21,5.33,29  "#T",0.0314,0.89,0.20,5.08,32  "#BBB",0.0233,0.66,0.19,4.82,43  "#BB",0.0201,0.57,0.18,4.57,50  "#B",0.0169,0.48,0.17,4.32,59  "2",0.0109,0.31,0.148,3.76,92  "4",0.0071,0.20,0.129,3.28,142  "5",0.0060,0.17,0.120,3.05,167  "6",0.0042,0.12,0.109,2.77,236  "7.5",0.0028,0.078,0.094,2.39,364  "8",0.0023,0.066,0.089,2.26,430  "8.5",0.0020,0.058,0.085,2.16,489  "9",0.0017,0.047,0.079,2.01,603  "12",0.0005,0.014,0.050,1.30,2025 |

Построчно читаем CSV файл:

|  |
| --- |
| import csv  csv\_path = /path/to/lesson\_7.csv'  with open(csv\_path, newline='') as csvfile:  reader = csv.DictReader(csvfile)  field\_names = reader.fieldnames  print(f"Field Names: {field\_names}")  for row in reader:  print(f"Row: {row}")  print(f"Access to row element: {row['Ounce']}") |

Вывод:

|  |
| --- |
| Field Names: ['Grade', 'Ounce', 'Gram', 'Inch', 'mm', 'PPO']  Row: OrderedDict([('Grade', '#TriBall'), ('Ounce', '0.7199'), ('Gram', '20.41'), ('Inch', '0.60'), ('mm', '15.24'), ('PPO', '1')])  Access to row element: 0.7199  Row: OrderedDict([('Grade', '#0000'), ('Ounce', '0.1943'), ('Gram', '5.51'), ('Inch', '0.38'), ('mm', '9.40'), ('PPO', '5')])  Access to row element: 0.1943  Row: OrderedDict([('Grade', '#000'), ('Ounce', '0.1601'), ('Gram', '4.54'), ('Inch', '0.36'), ('mm', '9.14'), ('PPO', '6')])  Access to row element: 0.1601  Row: OrderedDict([('Grade', '#00'), ('Ounce', '0.1231'), ('Gram', '3.49'), ('Inch', '0.33'), ('mm', '8.38'), ('PPO', '8')])  Access to row element: 0.1231  Row: OrderedDict([('Grade', '#0'), ('Ounce', '0.1122'), ('Gram', '3.18'), ('Inch', '0.32'), ('mm', '8.13'), ('PPO', '9')])  .. |

Добавим строку в существующий файл:

|  |
| --- |
| import csv  csv\_path = /path/to/lesson\_7.csv'  with open(csv\_path, 'a', newline='') as csvfile:  file\_writer = csv.writer(csvfile, delimiter=',', quotechar='"', quoting=csv.QUOTE\_MINIMAL)  writeable\_row = ["12345test", 1.2345, 6.78, 9.012, 3.45, 67]  file\_writer.writerow(writeable\_row) |

Создадим новый CSV-файл и прочитаем его:

|  |
| --- |
| new\_csv\_path = /path/to/lesson\_7\_new\_file.csv'  with open(new\_csv\_path, 'w', newline='') as csvfile:  fieldnames = ['id1', 'id2', 'id3', 'id4', 'id5', 'id6', 'id7', 'id8', 'id9', 'string']  file\_writer = csv.DictWriter(csvfile, fieldnames=fieldnames, delimiter=",", quoting=csv.QUOTE\_MINIMAL)  file\_writer.writeheader()  for i in range(10):  writeable\_row = {  'id1': 1,  'id2': 2,  'id3': 3,  'id4': 4,  'id5': 5,  'id6': 6,  'id7': 7,  'id8': 8,  'id9': 9,  'string': f"ne\"w te\\:st string {i}"  }  file\_writer.writerow(writeable\_row)  with open(new\_csv\_path, newline='') as csvfile:  reader = csv.DictReader(csvfile)  field\_names = reader.fieldnames  print(f"Field Names: {field\_names}")  for row in reader:  print(f"Row: {row}")  print(f"Access to row element: {row['id5']}") |

Вывод:

|  |
| --- |
| Field Names: ['id1', 'id2', 'id3', 'id4', 'id5', 'id6', 'id7', 'id8', 'id9', 'string']  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 0')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 1')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 2')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 3')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 4')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 5')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 6')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 7')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 8')])  Access to row element: 5  Row: OrderedDict([('id1', '1'), ('id2', '2'), ('id3', '3'), ('id4', '4'), ('id5', '5'), ('id6', '6'), ('id7', '7'), ('id8', '8'), ('id9', '9'), ('string', 'ne"w te\\:st string 9')])  Access to row element: 5 |

Получившийся файл:

|  |
| --- |
| id1,id2,id3,id4,id5,id6,id7,id8,id9,string  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 0"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 1"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 2"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 3"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 4"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 5"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 6"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 7"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 8"  1,2,3,4,5,6,7,8,9,"ne""w te\:st string 9" |

# Домашнее задание

1. Создать приложение, которое будет из готового файла с данными «Сбербанка» (https://www.sberbank.com/ru/analytics/opendata) выводить результат по параметрам:

* Тип данных
* Интервал дат
* Область

1. Визуализировать выводимые данные с помощью графика

# Используемая литература

1. <https://docs.python.org/3/library/csv.html>
2. <https://pythonprogramming.net/reading-csv-files-python-3/>
3. <https://www.rosminzdrav.ru/opendata>
4. <https://habr.com/post/331036/>
5. <https://data.gov.ru/opendata>
6. <https://data.mos.ru/opendata?categoryId=321&IsActual=true>
7. [https://ru.wikipedia.org/wiki/Открытые\_данные](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5)
8. <https://data.europa.eu/euodp/data/dataset?res_format=CSV>
9. <https://catalog.data.gov/dataset?res_format=CSV>