```
[
{
    "telfax": false,
    "tel": [
        "35-61-28"
    ],
    "address": "21050, м. Вінниця, 1 провул. Ботанічний, 38",
    "name": "Державна установа \"Територіальне медичне об'єднання МВС України
    "tin": "08734457",
    "note": "моз",
    "code": "0432",
    "reg": "02"
},
{
    "telfax": false,
    "tel": [
        "35-11-54"
    ],
    "address": "21014, м. Вінниця, вул. Свердлова, 185",
    "name": "Військово-медичний клінічний центр Центрального регіону",
    "tin": "07816153",
    "note": "моу",
    "code": "0432",
    "reg": "02"
},
{
    "telfax": false,
    "tel": [
        "35-31-85"
    ],
    "address": "21018, м. Вінниця, вул. Пирогова, 46",
    "name": "Вінницька обласна клінічна лікарня ім. М.І. Пирогова",
    "tin": "02011031",
    "code": "0432",
    "reg": "02"
},
{
    "telfax": false,
    "tel": [
        "2-33-40"
    ],
    "address": "22000, Вінницька обл., м. Хмільник, вул. Кірова, 71",
    "name": "Комунальна установа \"Хмільницька центральна районна лікарня\"",
    "tin": "01982695",
    "code": "04338",
    "reg": "02"
},
{
    "telfax": false,
    "tel": [
        "2-43-92"
    ],
    "address": "24400, Вінницька обл., м. Бершадь, вул. Будкевича, 2",
    "name": "Бершадська окружна лікарня інтенсивного лікування",
    "tin": "01982494",
    "code": "04352",
    "reg": "02"
},
{
```


український центр суспільних даних

Конвертація з формату CSV у JSON

посібник

Київ 2017

Конвертація з формату CSV у JSON. Посібник.

версія 0.9

© Ренат Насрідінов, Український центр суспільних даних, 2017.

поширюється за ліцензією Creative Commons Attribution 4.0 International (CC BY 4.0). Ви можете вільно ділитися — копіювати і розповсюджувати матеріал у будь-якому вигляді чи форматі, змінювати — реміксувати, трансформувати, і брати матеріал за основу для будь-яких цілей, навіть комерційних.

Ліцензіар не може відкликати ці дозволи, поки ви дотримуєтесь умов ліцензії.

Зміст

Вступ 4
Попередні домовленості 4
Масив CSV....4
Термінологія....5
Формат JSON 5
Конвертація за п’ять кроків....6
Що можна зробити за допомогою цього інструменту? 6
Опції конвертації....6
Опції завантаження налаштувань 6
Крок перший – вхідний масив даних в CSV 7
Крок другий (опціональний) – налаштування опцій вхідного CSV 8
Крок третій (опціональний) – налаштування опцій виводу 9
Крок четвертий (опціональний) – налаштування виводу JSON за допомогою шаблону 11
Створення вкладених структур без використання шаблонів....11
Крок п’ятий – генерація JSON 12
CSV у JSON. Перетворення для знавців PowerShell....13
Приклади....14
Приклад 1. Звичайний JSON 14
Приклад 2. MongoDB JSON 14
Приклад 3. CSV у JSON з ключем (унікальний ключ) 14
Приклад 4. CSV у JSON з ключем (не унікальний ключ) 15
Приклад 5. CSV у масив JSON 15
Приклад 6. CSV у масив колонок JSON 15
Приклад 7. Лапки у CSV розцінюються як дані 16
Приклад 8. Апостроф виступає у якості лапок 16
Приклад 9. CSV містить обернену косу риску для екранування 16
Приклад 10. Фільтр записів CSV за умовами 17
Приклад 11. Сортування вихідного JSON 17
Приклад 12. Використання значення null для порожніх полів 18
Приклад 13. Округлення десяткових значень 18
Приклад 14. Порядковий номер поля, яке буде визначено ключовим 19
Приклад 15. Назви ключів для масивів полів та даних 19
Приклад 16. Обробка значення NULL у CSV 20
Приклад 17. Вивід JSON за допомогою шаблону 21
Приклад 18. Створення вкладених об’єктів шляхом модифікації заголовків CSV 21
Корисні посилання 22

Користування онлайн-сервісом конвертації формату CSV у JSON

Вступ

Серед безлічі способів конвертувати один формат в інший(у нашому випадку CSV у JSON) варто виділити онлайн-конвертор на сайті convertcsv.com.

І хоча в подальшому мова піде лише про конвертацію цих форматів у цьому напрямку, не можна не зазначити, що сайт пропонує інструменти конвертації з/у формати HTML, JSON, SQL, XML, Excel, TXT, KML, YAML, та багато інших.

Розробники конвертера запевняють, що обробка виконується на стороні клієнта — це означає, що дані не передаються в Інтернет, а обробляються на комп'ютері користувача. І це дійсно так.

Посібник підготовлено Українським центром суспільних даних в якості методичного матеріалу до тренінгів «Відкриті дані для державних службовців».

Даний посібник здебільшого є перекладом інструкцій зі сторінки конвертору, до якого додано деякі пояснення та опис методу перетворення даних у форматі CSV у формат JSON за допомогою PowerShell.

Призначення посібника — надати можливість повною мірою використовувати доступні інструменти для створення якісних наборів відкритих даних.

Попередні домовленості

Масив CSV

Далі у прикладах буде використовуватись тестовий масив CSV-даних

```
код,автор,назва,день,кількість
1,Стус,Зимові дерева,четвер,8
2,Тичина,Спокуса,середа,14
3,Загребельний,Диво,субота,3
```

який утворено з наступної таблиці:

код	автор	назва	день	кількість
1	Стус	Зимові дерева	четвер	8
2	Тичина	Спокуса	середа	14
3	Загребельний	Диво	субота	3

Якщо вважати, що перший рядок цієї таблиці містить заголовки стовпчиків, то таблиця фактично містить 3 рядки даних.

Іноді у масиві може бути скорочено кількість рядків, або кількість стовпчиків.

Термінологія

Термінологія для таблиць, окрім звичних нам рядків та стовпчиків, і включає також поняття *поля* (*field*) та *запису* (*record*).

Поле – це місце перетину стовпчика та рядка, тобто клітинка, а *запис* — це сукупність полів, тобто рядок.

Формат JSON

JSON (JavaScript Object Notation — об'єктний запис JavaScript) —назва формату обміну даними, що з'явився на початку 2000-х років і з того часу виступає універсальним стандартизованим (ECMA-404, RFC 7159) форматом для обміну даними.

У JSON використовується дві структури даних:

- Набір пар «ім'я / значення»
- Список значень

Пара ім'я/значення у JSON має вигляд `"ім'я": "значення"` і у відношенні до конвертації CSV у JSON, що описана тут, зазвичай як `"ім'я"` виступає позначення стовпчику, що містить поле (якщо є заголовок, то це відповідне ім'я, якщо заголовок відсутній, то поле за замовчуванням отримує позначку `FIELDN`, де `N` – порядковий номер стовпця, що містить поле, зліва направо, починаючи з 1), а у якості `"значення"` – значення цього поля.

Якщо представити тестову таблицю у форматі JSON, то комірка на перетині другого рядка та другого стовпця, яка місить значення `Тичина`, буде являтися таким об'єктом:

```
"автор": "Тичина"
```

комірка на перетині першого рядка та п'ятого стовпця

```
"кількість": "8"
```

Дві структури, описані вище, набувають таких форм:

Значення може бути рядком* в подвійних лапках (`"Зимові дерева"`), або числом (42), або логічними `true` чи `false`, або `null`, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

Об'єкт — сукупність пар ім'я/значення, розділених комами, що обмежена символами фігурних дужок { та }. У кожній парі ім'я відділяється від значення двокрапкою:

`{"код": 1, "автор": "Стус"}` – 1 об'єкт, що містить 2 пари «ім'я/значення»

Масив — це сукупність (0 або більше) значень будь-якого дозволеного типу, розділених комами, обмежена символами квадратних дужок].

```
["Зимові дерева", "четвер", 8]
```

Рядок — це сукупність (0 або більше) символів юнікода, що обмежена подвійними лапками (").

Конвертація за п’ять кроків

Після переходу за посиланням ми бачимо перелік можливостей та опцій:

What can this tool do – Що можна зробити за допомогою цього інструменту?

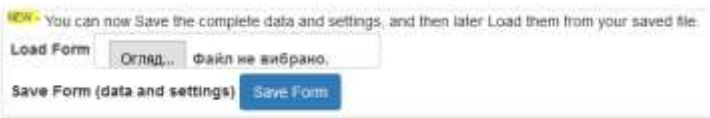
- **CSV to JSON** (CSV у JSON) — отримуємо масив структур JSON (приклад 1), що співпадає з вхідним CSV (і опціонально придатний для MongoDB), приклад 2
- **CSV to Keyed JSON** (CSV у JSON з ключем) — генерує JSON, у якому обране поле CSV буде виступати у якості ключового поля для решти полів (так звана хеш-таблиця або асоціативний масив. Якщо значення ключового поля унікальне, ми отримуємо `"keyvalue": {object}` (приклад 3), інакше `"keyvalue": [{object1}, {object2}, ...]` (приклад 4)
- **CSV to JSON Array** (CSV у масив JSON) — масив, який містить в свою чергу масив, утворений із рядків CSV-файлу (приклад 5)
- **CSV to JSON Column Array** (CSV у масив колонок JSON) — отримуємо файл JSON, де ключами виступають значення першого рядка, а значеннями — колонки файлу CSV (приклад 6)
- **Generate JSON via Template** (Генерація JSON по шаблону) — використання шаблону для формування виводу JSON
- Цифрові, логічні та `null` (порожні) значення можуть визначатися автоматично

What are my options – Опції конвертації

- Обирати регістр атрибутів
- Обирати ліміт кількості записів для обробки
- Обирати поля для виводу, а також вирівнювати їх
- Фільтрувати вивід JSON
- Створювати вкладені структури у JSON, використовуючи символ слешу (/) у заголовках стовпчиків CSV
- Сортувати дані CSV за зростанням або зменшенням перед конвертацією у JSON
- Конвертувати порожні (`NULL`) значення у CSV в аналогічні (`null`) у JSON
- Опціонально виводити `null` замість `""` для порожніх полів (значень)
- Опціонально не виводити об’єкт `field: value` (`ключ: значення`), якщо значення порожнє
- MongoDB Mode (Режим MongoDB) – один об’єкт на рядок без коми в якості роздільника

Також існує можливість зберегти налаштування конвертації у файл, а потім завантажувати їх (щоб не налаштовувати конвертацію кожного разу). Це робиться за допомогою ось цієї форми.

Опції завантаження налаштувань



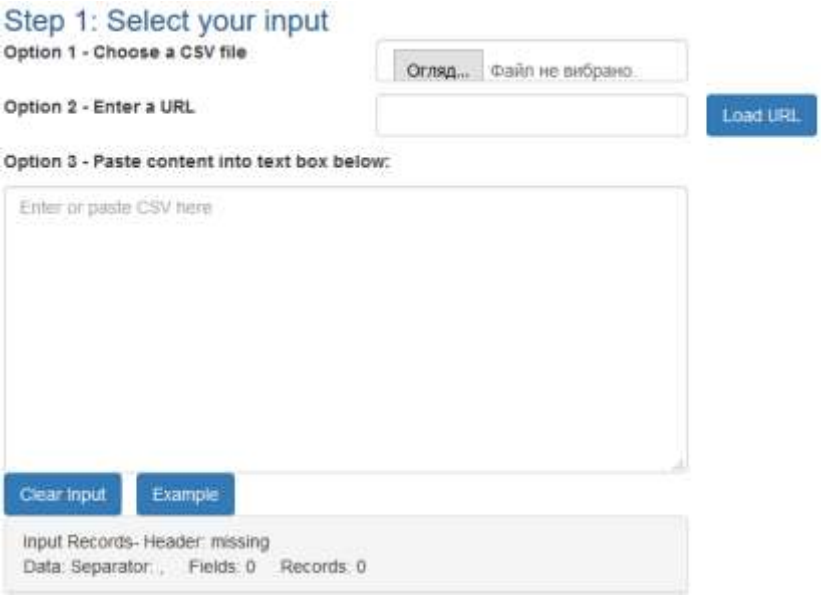
- *Load Form* — завантажити налаштування з файлу
- *Save Form* —зберегти налаштування у файл

Перш ніж перейти далі, зазначимо, що з 5 кроків обов'язковими є лише перший і останній. Зазвичай все працює нормально, але якщо ні — потрібно уважно придивитися до вашого CSV і, можливо, експериментувати з налаштуваннями із кроку 2.

Крок перший – вхідний масив даних в CSV

CSV-дані можна передати для конвертації трьома шляхами:

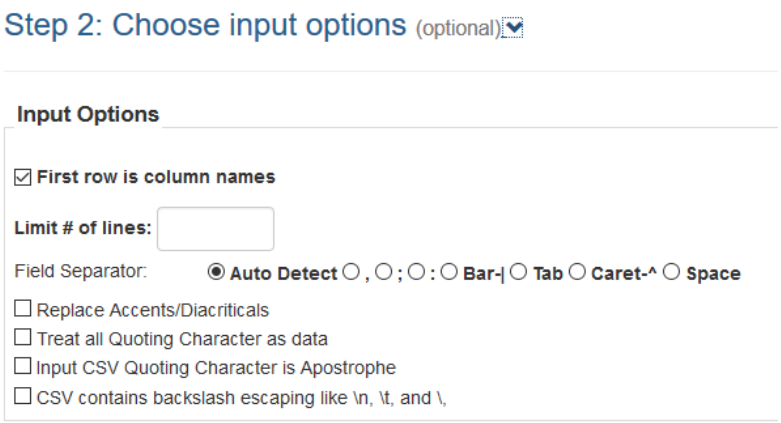
- **Option 1 - Choose a CSV file** – відкриття файлу з диска комп'ютера для конвертації. Після натискання «Огляд» з'являється діалогове вікно відкриття файлу
- **Option 2 - Enter a URL** – завантаження файлу із вебу по відомому URL. Потрібно вставити URL у поле справа та натиснути «Load URL» («Завантажити URL»)
- **Option 3 - Paste content into text box below** — просто вставити текст CSV у поле нижче. Кнопка «Clear input» («Очистити введене») очищує поле від введеного тексту, кнопка «Example» очищує поле і додає туди тестовий CSV-текст.



Нижче міститься панель, на якій після завантаження можна буде побачити інформацію щодо CSV:

- **Header** – Наявність заголовку. Набуває значень true якщо заголовок є, або missing, якщо його немає.
- **Fields** – Кількість полів (стовпчиків)
- **Records** – Кількість записів (рядків)
- **Separator** – Символ, що використовується як роздільник

Крок другий (опціональний) – налаштування опцій вхідного CSV

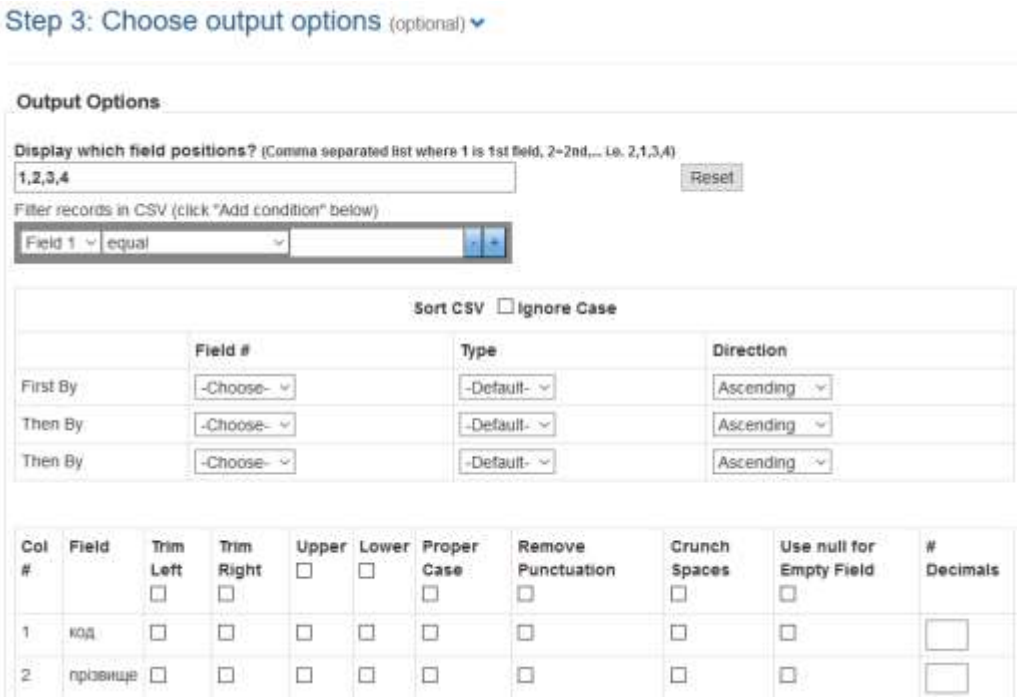


Опції CSV впливають на синтаксичний розбір Ваших даних.

- **First row is column names** – назви стовпчиків у першому рядку. Якщо відмітити чекбокс, значення першого рядка будуть вважатися назвами полів і будуть перетворені на значення ключів JSON (приклад 1).
- **Limit # of lines** – кількість рядків CSV, що будуть оброблені.
- **Field Separator** – символ-роздільник, що розділяє значення у CSV. Можливі варіанти, зліва направо: *Auto Detect* – автоматичне визначення, кома, крапка з комою, двокрапка, *Bar* (Вертикальна риска |), *Tab* – символ табуляції, *Caret* – символ каретки (^), *Space* – пробіл.
- **Replace Accents/Diacriticals** – Заміняти символи із верхніми та нижніми діакритичними знаками близькими за накресленням відповідниками з латинського алфавіту (опція більш актуальна для Західної Європи, де таких символів багато). Наприклад, чеське слово `háček` у CSV буде перетворено на `hacek`.
- **Treat all Quoting Character as data** – розцінювати лапки як дані. В цьому випадку якщо якесь значення в CSV-масиві буде подано у лапках, такі лапки будуть збережені (приклад 7).
- **Input CSV Quoting Character is Apostrophe** — Апостроф виступає у якості лапок. Значення, які містять роздільник, але які заключено у апострофи (‘) в якості лапок, будуть трактуватись як одне значення (приклад 8).
- **CSV contains backslash escaping like \n, \t, and ** – CSV містить обернену косу риску (\) для екранування керуючих послідовностей, наприклад `\n` або `\t`. У випадку обрання опції символ оберненої косої риски не буде в свою чергу екрановано і перетворено на `\\`, як того вимагає стандарт (приклад 9).

Крок третій (опціональний) – налаштування опцій виводу

На цьому кроці налаштовуються різноманітні фільтри, а також обробка значень.



Display which field positions? – Показувати поля з наступними позиціями. У це поле вводяться порядкові номери полів (стовпчиків), починаючи з 1, розділені комою. Так, якщо для нашого тестового CSV ввести у це поле 1, 2, 3, 5, то JSON буде сформовано без ключа `день`

Filter records in CSV – Фільтр записів CSV за умовами. Якщо натиснути на кнопку Add condition, з'являється можливість (показано на скріншоті вище) додавати прості арифметичні умови фільтрації записів за допомогою вказання умов для окремих полів, поля при цьому іменуються `FieldN`, де `N` – порядковий номер поля, починаючи з 1. Можливі варіанти умов:

- *equal* – дорівнює
- *not equal* – не дорівнює
- *less than* – менше ніж
- *less than or equal* – менше ніж або дорівнює
- *greater than*– більше ніж
- *greater than or equal* – більше ніж або дорівнює

Натискання на кнопки - + для кожної умови дозволяє комбінувати умови за допомогою логічних операторів `I` (AND) та `АБО` (OR) (приклад 10).

Опція Sort CSV (Сортувати CSV) дозволяє сортувати дані (максимум по трьох полях), потрібні поля вказуються зверху вниз – *Спочатку по* (*First By*), Потім по (*Then By*). Якщо стоїть позначка *Ignore Case*, регістр символів не буде враховуватись, в іншому випадку слово `Арифмометр` буде перед `арифмометр`.

Умови сортування задаються наступним чином. Зі списку *Field #* (*Поле №*) обирається номер поля (поля нумеруються з 1, починаючи з крайнього лівого). Список Type дозволяє вказувати тип даних

☐ Force Wrap numeric values in double quotes
If to Keyed JSON, which field # is key? 1 (Default 1)
☑ If to JSON Array, create array for column names with name = fields and Data name = data
☑ Always overwrite template when generating standard JSON
☑ Consider value of NULL in CSV to be null in JSON
☐ Do Not output field/value pair if value is empty
☑ Recreate nested objects and arrays (Headings use /)
☐ MongoDB Mode Each record on one line, no commas in between

для кожного поля – *String* (*рядок*[*]) або *Numeric* (*Цифри*), що впливатиме на сортування (приклад 11).

Поле *Direction* (Напрямок) задає напрямок сортування – *Ascending* (за зростанням) або *Descending* (за зменшенням).

Нижче розташовано таблицю налаштувань, що дозволяє керувати обробкою змінних полів як для всіх полів одразу (для цього потрібно поставити відмітку у заголовку відповідного стовпчику налаштувань), так і для окремого поля (поставити відмітку безпосередньо для такого поля).

В таблиці для кожного поля відображено його порядковий номер (стовпчик *Col #*), назву (стовпчик *Field*, при цьому якщо CSV має заголовок, то у якості значень беруться заголовки полів, якщо ні, то замість назви поля буде відображено `FIELD`*N*, де *N* – порядковий номер поля, починаючи з 1) та перелік опцій.

Col #	Field	Trim Left	Trim Right	Upper	Lower	Proper Case	Remove Punctuation	Crunch Spaces	Use null for Empty Field	# Decimals
		☐	☐	☐	☐	☐	☐	☐	☐	
1	код	☐	☐	☐	☐	☐	☐	☐	☐	

- **Trim Left** – обрізати пробіли на початку значень (“ Київ“ → “Київ“)
- **Trim Right** – обрізати пробіли в кінці значень (“Київ “ → “Київ“)
- **Upper** – конвертувати у великі літери (“Київ“ → “КИЇВ“)
- **Lower** – конвертувати у прописні літери (“КИЇВ“ → “Київ“)
- **Proper Case** – капіталізувати перші літери слів (“МІСТО КИЇВ“ → “Місто Київ“)
- **Remove Punctuation** – прибрати пунктуацію (“перший,другий“ → “ першийдругий“)
- **Crunch Spaces** – послідовні пробіли перетворюються на один (“Місто Київ“ → “Місто Київ“)
- **Use null for Empty Field** – використовувати значення `null` для порожніх полів. Якщо значення поля в CSV є порожнім, у JSON для таке значення буде представлено як `null`, у відповідності до RFC 7159 (приклад 12).
- **# Decimals** – кількість десяткових знаків (*працює лише якщо знаком роздільника цілої та десяткової частини є крапка*, наприклад `3.14`). У випадку, якщо значення є десятковим, зменшує кількість знаків після роздільника до зазначеного (**з округленням**). Так, якщо значення дорівнює `8.246` і у **# Decimals** вказано 2, то таке значення в JSON буде конвертовано як `8.25` (приклад 13)

Наступна група опцій застосовується до всіх значень CSV-масиву Ваших даних.

- **Force Wrap numeric values in double quotes** – вважати цифрові значення числовими, навіть якщо їх узято у подвійні лапки (рядок “42” буде вважатися числом 42, а рядок “3.14” – числом 3.14[*])
- **If to Keyed JSON, which field # is key? (Default 1)** – якщо конвертувати у JSON з ключем, яке поле буде цим ключем? Визначає порядковий номер поля, яке буде визначено ключовим, за замовчуванням це перше поле масиву CSV (приклад 14)

---

[*] Тут «рядок» позначає не «рядок у таблиці», а тип даних, який складається із символів алфавіту, в даному випадку “1” є не цифрою, а звичайним символом.

- **If to JSON Array, create array for column names with name= and Data name=** – якщо конвертувати у масив JSON, створювати окремі масиви для заголовків полів та даних, що будуть мати ключи з назвами name=<*назва_ключа_заголовків*> та *Data name*=<*назва_ключа_масиву_даних*>, за замовчуванням назвами цих масивів є fields для масиву заголовків полів та data для масиву даних (приклад 15).
- **Always overwrite template when generating standard JSON** – завжди перезаписувати шаблон при генерації стандартного JSON
- **Consider value of NULL in CSV to be null in JSON** – вважати значення NULL (незалежно від регістру літер – NULL або null) значенням null для JSON у відповідності до RFC 7159 (приклад 16).
- **Do Not output field/value pair if value is empty** – не виводити пару «ключ – значення», якщо значення відсутнє.
- **Recreate nested objects and arrays (Headings use /)** – якщо у заголовках використовується коса риска /, створювати вкладені об'єкти і масиви (див. Вкладені об'єкти)
- **MongoDB Mode** – Режим MongoDB – один запис на рядок, без коми в якості роздільника[†]. Якщо ви не знаєте, що це і для чого, не вмикайте цієї опції, оскільки дані в такому форматі не є стандартним JSON, що відповідає RFC 7159 і не обробляються стандартними засобами.

Крок четвертий (опціональний) – налаштування виводу JSON за допомогою шаблону

Це доволі езотерична практика, тому детальний розгляд питання застосування шаблонів при перетворенні виходить за рамки даного посібника.

Застосування шаблону дозволяє швидко сформувати JSON потрібної структури, наприклад створювати вкладені структури. Так, якщо тестова таблиця виглядатиме наступним чином:

код	книга		день	кількість
	автор	назва		
1	Стус	Зимові дерева	четвер	8

і у вихідному JSON необхідно відобразити інформацію щодо книги у вигляді у такому ж вигляді, достатньо створити відповідний шаблон і натиснути кнопку **Convert to JSON via Template** (приклад 17 – шаблон та отриманий JSON), після чого перейти безпосередньо до кроку 5.

Якщо у Вас є бажання дізнатися про шаблони більше, можете ознайомитись з цим на сторінці конвертеру, а також перейшовши за посиланням.

Створення вкладених структур без використання шаблонів за допомогою заголовків

Використання вкладених об'єктів є однією з властивостей формату, корисною для організації даних (нагадаємо, що відсутність такої властивості у формату CSV позбавляє його гнучкості в питаннях структурування даних).

Якщо масив CSV містить заголовки у першому рядку, то за умови незначної модифікації потрібних заголовків можна отримати вкладені об'єкти. Так, якщо звернутися до прикладу вхідних даних із

[†] Мається на увазі не кома в значеннях JSON, а кома між об'єктами JSON.

попереднього розділу, то для отримання об’єкта `книга`, що міститиме відповідні вкладені пари ключ/значення:

```
"книга": {
    "автор": "Стус",
    "назва": "Зимові дерева"
    }
```

потрібно у файлі CSV модифікувати заголовки відповідних стовпчиків, додавши до них спереду необхідне ім’я об’єкта та косу риску:

До (буде згенеровано звичайний JSON):

```
код,автор,назва,день,кількість
```

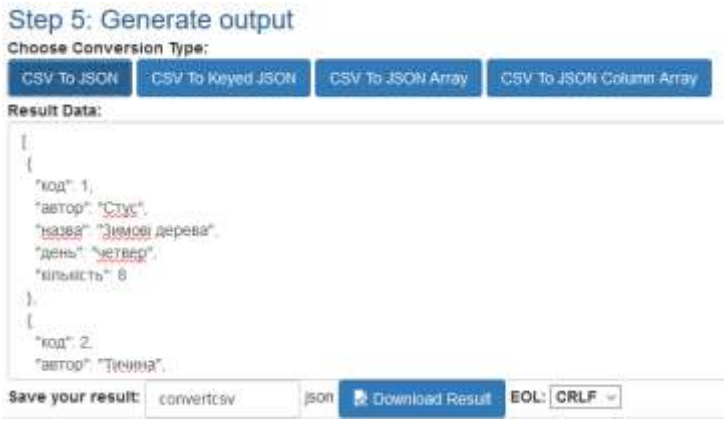
Після (буде згенеровано новий об’єкт):

```
код,книга/автор,книга/назва,день,кількість
```

Для наочності вхідні дані CSV та згенерований JSON можна переглянути у прикладі 18.

Слід відмітити, що при використанні цього методу створюється новий об’єкт, а не масив.

Крок п’ятий – генерація JSON



На цьому, завершальному кроці шляхом натискання на одну з чотирьох кнопок обирається тип конвертації:

- **CSV to JSON** – CSV у звичайний JSON
- **CSV to Keyed JSON** – CSV у JSON з ключем
- **CSV to JSON Array** – CSV у масив JSON
- **CSV to JSON Column Array** – CSV у масив колонок JSON

У полі **Result Data** міститься сковертований у JSON масив CSV-даних. Ви можете виділити і скопіювати його. Також можна просто зберегти JSON у файл на диску, натиснувши кнопку **Download Result** (*Завантажити результат*).

Поле **Save your result** (*Збереження результату*) містить ім’я файлу, який буде збережено. Зверніть увагу, що розширення `.json` писати не треба, воно буде додано автоматично.

Список **EOL** (**E**nd **O**f **L**ine, *Кінець строки*) призначений для вибору, якою послідовністю керуючих символів буде завершуватися кожний рядок – **CRLF** або **LF**. Зазвичай варто залишити варіант **CRLF**.

Для допитливих зазначимо, що **CRLF** використовується в операційних системах Microsoft Windows, а **LF** – в Unix-подібних операційних системах. Але оскільки кодуванням JSON за замовчуванням є Unicode, то не має значення, які саме символи кінця рядка використовуються, бо в Unicode вони сприймаються однаково.

CSV у JSON. Перетворення для знавців PowerShell

Перетворити CSV на JSON можливо не лише за допомогою онлайн-конвертору. Якщо Ви знаєте, що таке PowerShell, то цей розділ саме для Вас. Оскільки основною темою посібнику було використання онлайн-інструменту, то ця частина пропонується для самостійного вивчення тими, хто вже мав невеличкий досвід використання командлетів PowerShell.

Починаючи з версії Powershell 3.0 (входить в поставку Windows 8, доступна для Windows 7 та Windows Server 2008,але не для Windows Vista) для конвертації CSV у JSON можна використовувати вбудовані командлети Import-Csv та ConvertTo-Json.

Процедура конвертації зводиться до виконання наступного конвейеру командлетів з подальшим перенаправленням результату у файл:

```
Import-Csv [опції] файл.csv | ConvertTo-Json [опції] > файл.json
```

Кожен з двох командлетів має власні необов'язкові параметри, що дозволяють певною мірою керувати процесом конвертації.

Import-Csv має наступні опції:

-Delimiter <Char> — дозволяє вказати символ-роздільник значень. Наприклад, якщо роздільником є не кома (яка встановлена як параметр за замовчуванням, що більш звично для Європи та США), а крапка з комою, то достатньо вказати:

```
Import-Csv -Header "id","name" -Delimiter ";" файл.csv
```

(лапки навколо символу-роздільника потрібно вказувати лише для крапки з комою, інші символи, наприклад двокрапка, цього не потребують).

-Header <String[]> – дозволяє визначити заголовки стовпчиків для файлу, що імпортується. Заголовок вказується як розділений комами перелік значень, кожне з яких узято у одинарні чи подвійні лапки. Якщо значень менше, ніж стовпчиків, буде конвертовано стільки стовпчиків, стільки вказано значень, інші буде проігноровано, якщо більше – залишкові заголовки буде проігноровано.

-Encoding <String> – визначає кодування файлу CSV. Можливі значення цього параметру: Unicode, ASCII (за замовчуванням), UTF7, UTF8, UTF32, BigEndianUnicode, OEM.

Приклад:

```
Import-Csv -Encoding UTF8 -Delimiter ";" -Header "id","name" filename.csv
```

ConvertTo-Json має кілька опцій, але нас цікавить лише одна:

-Compress – дозволяє отримати стиснений JSON без відступів.

Приклади

Приклад 1. Звичайний JSON ↩

```
[
 {
   "код": 1,
   "автор": "Стус",
   "назва": "Зимові дерева",
   "день": "четвер",
   "кількість": 8
 },
 {
   "код": 2,
   "автор": "Тичина",
   "назва": "Спокуса",
   "день": "середа",
   "кількість": 14
 }
]
```

Приклад 2. MongoDB JSON ↩

```
{"код": 1, "автор": "Стус", "назва": "Зимові дерева", "день": "четвер", "кількість": 8}
{"код": 2, "автор": "Тичина", "назва": "Спокуса", "день": "середа", "кількість": 14}
```

Приклад 3. CSV у JSON з ключем (унікальний ключ) ↩

```
{
   "1": {
      "автор": "Стус",
      "назва": "Зимові дерева",
      "день": "четвер",
      "кількість": 8
   },
   "2": {
      "автор": "Тичина",
      "назва": "Спокуса",
      "день": "середа",
      "кількість": 14
   }
}
```

Приклад 4. CSV у JSON з ключем (не унікальний ключ) ↩

```
{
   "2": [
      {
         "автор": "Стус",
         "назва": "Зимові дерева",
         "день": "четвер",
         "кількість": 8
      },
      {
         "автор": "Тичина",
         "назва": "Спокуса",
         "день": "середа",
         "кількість": 14
      }
   ]
}
```

Приклад 5. CSV у масив JSON ↩

```
[
  [1,"Стус","Зимові дерева","четвер",8],
  [2,"Тичина","Спокуса","середа",14]
]
```

Приклад 6. CSV у масив колонок JSON ↩

```
{
   "код":[1,2],
   "автор":["Стус","Тичина"],
   "назва":["Зимові дерева","Спокуса"],
   "день":["четвер","середа"],
   "кількість":[8,14]
}
```

Приклад 7. Лапки у CSV розцінюються як дані ↩

CSV:

```
код,автор,назва,день,кількість
1,Стус,"Зимові дерева",четвер,8
```

JSON:

```
[
 {
   "код": 1,
   "автор": "Стус",
   "назва": "\"Зимові дерева\"",
   "день": "четвер",
   "кількість": 8
 }
]
```

Приклад 8. Апостроф виступає у якості лапок ↩

CSV:

```
код,автор,назва,день,кількість
1,Стус,'Зимові, дерева',четвер,8
```

JSON:

```
[
 {
   "код": 1,
   "автор": "Стус",
   "назва": "Зимові, дерева",
   "день": "четвер",
   "кількість": 8
 }
]
```

Приклад 9. CSV містить обернену косу риску для екранування ↩

CSV:

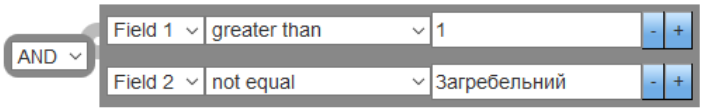
```
код,автор,назва,день,кількість
1,Стус,Зимові\nдерева,четвер,8
```

JSON:

```
[
 {
   "код": 1,
   "автор": "Стус",
   "назва": "Зимові\nдерева",
   "день": "четвер",
   "кількість": 8
 }
]
```

Приклад 10. Фільтр записів CSV за умовами ↩

Умови:



Значення стовпчика 1 повинно бути більшим, ніж 1 *ТА* значення стовпчика 2 не повинно дорівнювати Загребельний.

JSON:

```
[
 {
   "код": 2,
   "автор": "Тичина",
   "назва": "Спокуса",
   "день": "середа",
   "кількість": 14
 },
]
```

Приклад 11. Сортування вихідного JSON ↩

Умови:

Значення у стовпчику 5 вважати рядком, сортувати за зменшенням.

CSV:

```
код,автор,назва,день,кількість
2,Тичина,Спокуса,середа,14
3,Загребельний,Диво,субота,114
```

JSON:

```
[
 {"код": 2,
   "автор": "Тичина",
   "назва": "Спокуса",
   "день": "середа",
   "кількість": 14},
 {"код": 3,
   "автор": "Загребельний",
   "назва": "Диво",
   "день": "субота",
   "кількість": 114}
]
```

За умови трактування значень у 5 стовпчику як цифр, сортування за зменшенням призвело б до того, що об’єкти JSON буде виведено у зворотному порядку — спочатку 114, потім 14.

Приклад 12. Використання значення null для порожніх полів ↩

CSV (стовпчик 4 порожній)

```csv
код,автор,назва,день,кількість
2,Тичина,Спокуса,,14
3,Загребельний,Диво,"",114
```

JSON:

```json
[
 {
   "код": 2,
   "автор": "Тичина",
   "назва": "Спокуса",
   "день": null,
   "кількість": 14
 },
 {
   "код": 3,
   "автор": "Загребельний",
   "назва": "Диво",
   "день": null,
   "кількість": 114
 }
]
```

Приклад 13. Округлення десяткових значень ↩

CSV

```csv
код,прізвище,день,кількість
1,Стус,четвер,8.113
2,Тичина,середа,1.1469
```

JSON:

```json
[
 {
   "код": 1,
   "прізвище": "Стус",
   "день": "четвер",
   "кількість": 8.11
 },
 {
   "код": 2,
   "прізвище": "Тичина",
   "день": "середа",
   "кількість": 1.15
 }
]
```

Приклад 14. Порядковий номер поля, яке буде визначено ключовим ↩

If to Keyed JSON, which field # is key? 4 **(Default 1)**

CSV

```
код,автор,назва,день,кількість
1,Стус,Зимові дерева,четвер,8
2,Тичина,Спокуса,середа,14
```

JSON:

```
{
   "четвер": {
      "код": 1,
      "автор": "Стус",
      "назва": "Зимові дерева",
      "кількість": 8
   },
   "середа": {
      "код": 2,
      "автор": "Тичина",
      "назва": "Спокуса",
      "кількість": 14
   }
}
```

Приклад 15. Назви ключів для масивів полів та даних ↩

☑ **If to JSON Array, create array for column names with name =** header **and Data name =** array

CSV

```
код,автор,назва,день,кількість
1,Стус,Зимові дерева,четвер,8
2,Тичина,Спокуса,середа,14
```

JSON:

```
{
  "header": ["код", "автор", "назва", "день", "кількість"],
  "array": [
  [1,"Стус","Зимові дерева","четвер",8],
  [2,"Тичина","Спокуса","середа",14]
]
}
```

Приклад 16. Обробка значення NULL у CSV ↩

CSV

```csv
код,автор,назва,день,кількість
1,Стус,Зимові дерева,null,8
2,Тичина,Спокуса,NULL,14
```

JSON:

```json
[
 {
   "код": 1,
   "автор": "Стус",
   "назва": "Зимові дерева",
   "день": null,
   "кількість": 8
 },
 {
   "код": 2,
   "автор": "Тичина",
   "назва": "Спокуса",
   "день": null,
   "кількість": 14
 }
]
```

Приклад 17. Вивід JSON за допомогою шаблону ↩

Шаблон:

Top

```
[
```

Repeating Section for each Line of CSV

```
{lb}
    "{h1}":"{f1}"
   ,"книга": [
      {lb}"{h2}":"{f2}"{rb},
      {lb}"{h3}":"{f3}"{rb}
      ],
   "{h4}":"{f4}",
   "{h5}":{(f5)==""?"null":f5}
{rb}
```

Bottom

```
]
```

JSON

```
[{
    "код": "1",
    "книга": [
        {"автор": "Стус"},
        {"назва": "Зимові дерева"}
],
    "день": "четвер",
    "кількість": 8
}, {
    "код": "2",
    "книга": [
        {"автор": "Тичина"},
        {"назва": "Спокуса"}
        ],
    "день": "середа",
    "кількість": 14
}]
```

Приклад 18. Створення вкладених об’єктів шляхом модифікації заголовків CSV ↩

CSV:

```
код,книга/автор,книга/назва,день,кількість
1,Стус,Зимові дерева,четвер,8
```

JSON

```
[
 {
   "код": 1,
   "книга": {
      "автор": "Стус",
      "назва": "Зимові дерева"
   },
   "день": "четвер",
   "кількість": 8
}
]
```

Корисні посилання

- Відкритий посібник з відкритих даних – Український центр суспільних даних
- Посібник по роботі з даними – Тексти.org.ua
- Посібник «Відкриті дані українських міст. Рекомендації щодо впровадження політик» – громадянська мережа ОПОРА
- Методичні рекомендації щодо оприлюднення наборів даних у формі відкритих даних (проект) – Державне агентство з питань електронного урядування