Розробникам, програмістам, роботам...

**API** (*Application Programming Interface, або Інтерфейс прикладного програмування*) — корисний та сучасний інструмент для розробників при роботі з інформаційними системами. Це набір готових процедур, підпрограм, функцій, посилань чи параметрів, які дозволяють особливим чином використовувати інформаційні системи для отримання структурованих або неструктурованих наборів даних чи іншої взаємодії. Під час розвитку [відкритих даних](https://data.rada.gov.ua/open/main/opendata/page4) він набув особливої популярності через зручність та оперативність доступу до потрібної інформації шляхом запитів з відповідними параметрами.

Деякі державні та комерційні інформаційні системи мають або використовують API. У тому числі й у систем Верховної Ради України є режим "особливого" доступу (в автоматизованому режимі, напряму або за допомогою цього Порталу) — через API, документація до якого представлена нижче. Якщо ви розробник, то завжди зможете інтегрувати набори даних систем (розділів) Верховної Ради України у власні продукти та рішення. Доступ до API відбувається анонімно без обмежень, але якщо система захисту інфрастуктури раптом заблокує ваш IP, зверніться до [адміністратора Порталу](mailto:lawup@rada.gov.ua?subject=OpenData%20API) для реєстрації IP вашої системи у "білому списку".

Інформаційні системи (розділи) з API:

* [Портал відкритих даних ВРУ](https://data.rada.gov.ua/open/main/api/page2)
* [Законодавство України](https://data.rada.gov.ua/open/main/api/page3)

Портал відкритих даних ВРУ

Цей Портал крім звичайного відображення сторінок паспортів наборів та реєстрів (формат HTML) може формувати структуровані набори метаданих кожного паспорту чи набору у трьох відкритих форматах: CSV, JSON та XML. Структури цих наборів описані у [відповідному розділі](https://data.rada.gov.ua/open/data/etc). Для цього потрібно знати внутрішній id (або глобальний guid) ідентифікатор набора даних та додати до її звичайної URL адреси потрібний формат через кому. Наприклад:

| **Формат** | **URL** |
| --- | --- |
| HTML | http://data.rada.gov.ua/open/data/id |
| CSV | http://data.rada.gov.ua/open/data/id**.csv** |
| JSON | http://data.rada.gov.ua/open/data/id**.json** |
| XML | http://data.rada.gov.ua/open/data/id**.xml** |

**Зауваження:** API повертає дату та час оновлення файлу чи набору у заголовку HTTP (Last-Modified), тому враховуйте це при програмуванні роботів. Щоб зменшити трафік та навантаження за рахунок перевірки (If-Modified-Since), зберігайте та повертайте її серверу при наступному запиті.

На відміну від статичних метаданих (паспорту чи реєстру набору), які знаходяться у каталозі http://data.rada.gov.ua/ogd/ та оновлюються щогодини, оперативний набір, отриманий через API, додатково містить перелік всіх наявних файлів для перевірки чи завантаження.

Порядок роботи

Зазвичай ідентифікатори не змінюються, а файли з часом оновлюються і можуть, навіть, змінити шлях в результаті реорганізації даних. Тому не варто жорстко запам'ятовувати прямі шляхи до файлів! Перед завантаженням файлу набору спочатку потрібно звернутися за його паспортом й оновити свою копію, перевірити, чи змінилися файли, які завантажували перед тим, а вже потім завантажити сам набір, розмір якого в кілька разів може відрізнятися від метаданих.

## Законодавство України

ІПС ["Законодавство України"](http://zakon.rada.gov.ua/) Верховної Ради України в Інтернет/Інтранет, набори якої представлені в [окремому реєстрі](https://data.rada.gov.ua/open/data/zak) на Порталі відкритих даних ВРУ, також має API для завантаження й оновлення текстів (та їх редакцій) всіх нормативно-правових документів, які в ній знаходяться.

Для того, щоб скористатися цим API **потрібно**:

1. передавати в запитах UserAgent рядок "OpenData" або встановити куку OpenData будь-яке значення,
2. формувати посилання на документи та переліки наступного вигляду:

| **Документ або список** | **Формат** | **URL** |
| --- | --- | --- |
| *Документи* | | |
|  |  |  |
| Текст документа nreg | HTML | http://data.rada.gov.ua/laws/show/nreg |
| Картка документа nreg | HTML | http://data.rada.gov.ua/laws/card/nreg |
| Документ повністю nreg | JSON | http://data.rada.gov.ua/laws/show/nreg**.json** |
| Картка документа nreg | JSON | http://data.rada.gov.ua/laws/card/nreg**.json** |
| Чистий текст документа nreg | TXT | http://data.rada.gov.ua/laws/show/nreg**.txt** |
| *Списки документів* | | |
| Список поновлених документів | HTML | http://data.rada.gov.ua/laws/main/r |
| Список найновіших надходжень (за день) | HTML | http://data.rada.gov.ua/laws/main/nn |
| Список нових надходжень (30 днів) | HTML | http://data.rada.gov.ua/laws/main/n |
| Список всіх документів (по сторінках) | HTML | http://data.rada.gov.ua/laws/main/a[/page1] |
| Список nreg номерів поновлених документів | TXT | http://data.rada.gov.ua/laws/main/r**.txt** |
| Список dokid номерів поновлених документів | JSON | http://data.rada.gov.ua/laws/main/r**.json** |
| Список  RSS поновлених документів | XML | http://data.rada.gov.ua/laws/main/r**.xml** |
| Список карток та посилань на документи, розділених табуляціями | TSV | http://data.rada.gov.ua/laws/main/r**.tsv** |

**Зауваження:** API повертає дату та час оновлення документа чи списка у заголовку HTTP (Last-Modified), тому враховуйте це при програмуванні роботів. Щоб зменшити трафік та навантаження за рахунок перевірки (If-Modified-Since), зберігайте та повертайте її серверу при наступному запиті.

### Структури даних

Опис структур даних в картці документа знаходится у файлах по [карткам](https://data.rada.gov.ua/open/data/doc-stru) та [історії](https://data.rada.gov.ua/open/data/ist-stru) документів на Порталі. Але ми зібрали їх для зручності у таблиці нижче:

| **Група section** | **Елемент element** | **Формат format** | **Параметри attributes** | **Призначення annotation** |
| --- | --- | --- | --- | --- |
| **docs** | **doc** | object[\t] | @doc\_card=(dokid nreg nazva status types organs minjust npix) | Картка документа |
| **doc** | dokid | integer | index=true | Ідентифікатор документа |
| **doc** | nreg | string | pattern=^[0-9nprvz][0-9\/\\_\-a-zа-яїіёєґ]{3,11}$ | Системний номер документа |
| **doc** | nazva | string |  | Назва документа |
| **doc** | status | object[:], short | default=0, reference=#stan, @stat\_card=(status status\_from status\_to) | Стан документа |
| **doc** | types | array[|], integer | notnull, unique, reference=#typs | Види документа |
| **doc** | organs | object[:] | @org\_card=(orgid orgdat orgnum) | Видавники документа |
| **doc** | minjust | object[:] | @min\_card=(minid mindat minnum) | Реєстрація документа |
| **doc** | npix | byte | default=0 | Номер картинки для відображення у списку |
| **docs** | **ist** | object[\t] | @ist\_card=(dokid history links publics temy klasy tags komitet bookmark) | Історія документа, публікації, класифікація, зв'язки |
| **ist** | history | array[|], object[:] | @hist\_card=(poddat podid pidstava) | Історія документа |
| **ist** | links | object[#] | @link\_card=(pos\_links pr\_links zv\_links text\_links) | Відношення документа |
| **ist** | publics | array[|], object[:] | @pubs\_card=(pubdat vydid n\_pub) | Публікації документа |
| **ist** | temy | array[|], integer | reference=#temy | Теми документа |
| **ist** | klasy | array[|], integer | reference=#klasname | Класифікація документа |
| **ist** | tags | array[|], integer | reference=#tags | Ознаки документа |
| **ist** | komitet | array[|], integer | reference=#koms | Комітети документа |
| **ist** | bookmark | object[#] | reference=#book, @book\_card=(book\_pos book\_cnt) | Інформація про закладки в тексті |

**SON** (від англ. *JavaScript Object Notation*) – текстовий відкритий формат, оснований на Javascript представлені та призначений для обміну даними в Інтернет між сервером та клієнтом або сервером і сервером. Хоча він позиціонується, як незалежний від системи і мови програмування, частіше за все використовується за допомогою програм на Javascript, але як і інші текстові формати, легко читається людиною.

Найбільшу популярність JSON набув після створення інтерактивних веб-сторінок, дані до яких через API передавались під час взаємодії користувача з елементами інтерфейсу (т.з. технологія AJAX). За рахунок своєї лаконічності, на відміну від XML, простоті й швидкості використання саме в програмах на Javascript, широкими можливостями в серіалізації даних – рекурсивного перетворення в текстовий вигляд складних об’єктів, формат активно використовується для формування «на льоту» та передачі структур даних в Інтернет в різних інформаційних системах і сервісах.

[Синтаксис JSON](https://ru.wikipedia.org/wiki/JSON) представляє собою текст у вигляді програмного коду Javascript, який описує кілька варіантів структур:

1. масив в квадратних дужках (значення через кому),
2. іменований масив (чи об’єкт) у фігурних дужках (пари ключ-значення, розділені двома крапками, через кому),
3. просте значення (число, рядок, true, false, null),
4. функцію.

Для перетворення формату JSON в об’єкт в пам’яті (*десеріалізації*) в Javascript використовується функція eval(), яка виконує завантажений в текстовий рядок програмний код.

Якщо для обробки використовувати тільки мову Javascript, виникають як мінімум дві проблеми. По-перше, щоб обробити дані, потрібно мати приблизно в два рази більше вільної пам’яті, ніж файл (спочатку завантажується повний текст JSON, а після його виконання, дані об’єкту завантажуються повторно у вигляді відповідних структур). По-друге, можливість передачі програмного коду, який автоматично стартує, дозволяє зловмисникам отримати несанкціонований доступ до системи. Цю вразливість також можна уникнути, якщо спочатку перевірити і очистити текст JSON від заборонених об’єктів, а дані перетворити на коректні значення.

Через нераціональне використання пам’яті і ресурсів в JSON краще працювати зі об’єктами будь-якою складністю, але не дуже великих розмірів, в швидкому режимі «запит-відповідь».