

Zadanie 2

Z g-drive si importujte db s názvom **ov.sql** do Vášho PostgreSQL servera a rozšírte svoj aplikačný server o nasledujúce endpoint-y. Pri vypracovaní daného zadania je potrebné dávať pozor na **SQL injection**. Text zadania dopĺňa súbor dbs2021-0.1.0.yaml s popisom API.

1. Endpoint – metóda GET `/v1/ov/submissions`

Obsahuje nasledujúce možnosti, ktoré je možné ľubovoľne kombinovať:

1. Pre tabuľku `ov/or_podania_issues` vytvorte endpoint

GET `/v1/ov/submissions`

ktorý bude stránkovaný a bude vracať pole objektov, ktoré priamo mapujú databázové stĺpce:

`id`, `"br_court_name"`, `"kind_name"`, `"cin"`, `"registration_date"`,
`"corporate_body_name"`, `"br_section"`, `"br_insertion"`, `"text"`,
`"street"`, `"postal_code"`, `"city"`

Stránkovanie bude fungovať na základe GET parametrov `"page"` a `"per_page"`, kde štandardný počet záznamov na stránku je 10.

Príklad:

GET `/v1/ov/submissions/?page=1&per_page=10` je ekvivalentné s GET `/v1/ov/submissions`

Parameter `"page"` definuje stránku v poradí, ktorá sa aktuálne zobrazuje, parameter `"per_page"` hovorí o tom, koľko záznamov sa nachádza na jednej stránke.

Príklad:

GET `/v1/ov/submissions/?page=3&per_page=10` zobrazí záznamy v poradí 31 – 40 vrátane.

Keďže sú to parametre, tak si dávajte pozor, aby to fungovalo pre všetky možné kombinácie.

2. Umožnite vyhľadávanie nad nasledovnými stĺpcami: `"corporate_body_name"`, `"cin"`, `"city"` fulltextovo (OR medzi nimi) a nad poľom `"registration_date"` väčšie rovné aj menšie rovné hodnoty nasledovne:

Príklad:

GET
`/v1/ov/submissions/?query=Orange%20Slovensko®istration_date_gte=2016-06-`

17%2011%3A14%3A42.579414®istration_date_lte=2017-06-17%2011%3A14%3A42.579414

3. Umožnite zoradovať výsledky podľa jedného vybraného stĺpca spomedzi všetkých ktoré sa štandardne vracajú nasledovne:

Príklad:

GET

/v1/ov/submissions/?order_by=registration_date&order_type=desc

2. Endpoint - metóda POST

Vytvorte endpoint s metódou

POST /v1/ov/submissions

ktorá vytvorí nový záznam do tabuľky ov/or_podania_issues. Metóda musí validovať nasledovné hodnoty:

"br_court_name" – povinné pole

"kind_name" – povinné pole

"cin" – povinné pole, len číselná hodnota

"registration_date" – povinné pole, hodnota môže byť len z aktuálneho roka

"corporate_body_name" – povinné pole

"br_section" – povinné pole

"br_insertion" – povinné pole

"street" – povinné pole

"postal_code" – povinné pole

"city" – povinné pole

Systém automaticky doplní správne "created_at", "updated_at", "address_line" a zároveň vytvorí záznamy v tabuľkách "raw_issues" a "bulletin_issues". Do ostatných stĺpcov vložte znak -. V prípade úspešného zapísania, vráťte HTTP status kód 201 Created a detailom záznamu v rovnakom formáte ako je v liste.

V prípade validačnej chyby vráťte chybu v tvare s HTTP status kódom [422](#):

```
{
  "errors": [
    {
      "field": "registration_date",
      "reasons": ["required", "invalid_range"]
    },
    {
      "field": "cin",
      "reasons": ["required", "not_number"]
    },
    {
```

```
        "field": "street",
        "reasons": ["required"]
    }
]
```

3. Endpoint - DELETE

Vytvorte endpoint s metódou

```
DELETE /v1/ov/submissions/<id>
```

kde <id> je id záznamu na zmazanie. Daný záznam sa zmaže z DB. Nezabudnite zmazať aj príslušné záznamy v tabuľkách "raw_issues" a "bulletin_issues". V prípade, že záznam neexistuje vráťte HTTP status kód 404, ak sa podarilo záznam zmazať tak 204 a žiadny response body.