Додаток 4

### СТРУКТУРА даних файлу-відповіді на файл-запит (Додаток 3) з переліком об’єктів Реєстру, охоронні зони яких «перетинаються» з отриманим «полігоном»

| **Рівень вкладеності** | **Назва параметра** | **Обов’язковість** | **Тип даних** | **Опис** | **Правила** |
| --- | --- | --- | --- | --- | --- |
| 1 | sign | 1 | string | Підпис відповіді. Підписується JSON рядок з результатами. | base64 |
| 1 | code | 1 | number | Код виконання запита | Містить код результату прикладного рівня (див. Додаток 9) |
| 1 | error | 0 | string | Опис помилки | Додається коли параметр code містить значення відмінне від 0 |
| 2 | results | 1 | object | Масив з переліком ідентифікаторів обʼєктів | Ідентифікатори мають формат UUIDВсі ідентифікатори групуються згідно переліку вхідних параметрівОбʼєкт буде порожнім при помилках |

Приклад (помилка):

{

 “sign”: “...”,

 “code”: 1,

 “error”: “Відсутній параметр ‘params’”,

 “results”: {}

}

Приклад (успішний):

{

 “sign”: “...”,

 “code”: 0,

 “result”: {

 “0”: [

 “00000000-0000-0000-0000-000000000000”,

 …

 ],

 “1”: [

 “00000000-0000-0000-0000-000000000000”,

 …

 ],

 }

}

**Де:**

 **“0”, “1”** - ключі масиву, які відповідають масиву вхідних параметрів, тобто при виконанні запита передається масив координат відповідного типу (point, linestring, polygon, інше.), результати будуть згруповані по цим наборам координат, наприклад результату вище відповідає запит:

{

 “params”: {

 "areas": [

 “polygon (1 1, 1 2, 2 3, 3 5, 1 1)”,

 “point(0 0)”

 ]

 }

}