

## Запрос объектов мониторинга в СПИК

Запрос объектов мониторинга позволяет получить список транспортных средств, доступных данному пользователю.

Алгоритм запроса объектов мониторинга в СПИК:

1. Пройти авторизацию в **СПИК** (см. [Авторизация](#)) и получить токен авторизации.
2. Получить количество доступных объектов мониторинга, вызвав у сервиса **SpicUnitsService** метод **GetAllUnitsCount**.
3. Составить **SpicIdObjectsChunkRequest** для постраничной загрузки объектов. В запросе необходимо указать смещение и количество объектов.
4. Получить объект мониторинга, вызвав у сервиса **SpicUnitsService** метод **GetAllUnits** с параметром **SpicIdObjectsChunkRequest**, составленном в пункте 3.

Ниже приведены примеры для JS и C#, иллюстрирующие данный алгоритм в контексте запроса **REST** и **SOAP** соответственно.

### Пример на JS.

```
var baseUrl = 'http://localhost:8081/spic/units/rest/';
var getAllUnitsCountUrl = baseUrl;
var getAllUnitsServerUrl = baseUrl + 'getAllUnitsPaged';
var getUnitServerUrl = baseUrl + 'getUnit';
var unitsPerPage = 20;

var provider = new HttpJsonRequestProvider();

function getAllUnitsCount() {

    return provider.GETAuthorized(getAllUnitsCountUrl,
        null,
        getAuthorizationToken()).response;

}

function getUnit(unitId) {

    return provider.POST(getUnitServerUrl,
        { @jectId: unitId },
        getAuthorizationToken()).response;

}

function getUnitsPage(offset) {

    return provider.POSTAuthorized(getAllUnitsServerUrl,
```

```
Offset: offset,
Count: unitsPerPage
```

```
}.getAuthorizationToken().response;
```

```
} var offset = 0;
```

```
var units = getUnitsPage(offset).Units;
```

## Пример получения списка объектов мониторинга на языке C#.

1. Создать заглушку к сервису **SpicUnitsService** .
2. Добавить **AuthorizationBehavior** к заглушке.
3. Выполнить авторизацию в СПИК.
4. Внести код в проект, расположенный ниже.

```
//????????? ?????????? ??????????.
var unitsClient = new SpicSoapUnitsServiceClient();

//????????????? ?????????? ??????????????.
unitsClient.Endpoint.Behaviors.Add(new AuthorizationBehavior());

//????????????? ?????????????? ?????????????? ?????????? ??????????????????.
var unitsCount = unitsClient.GetAllUnitsCount();

//????????????? ??????????.
var request = new SpicObjectsChunkRequest { Offset = 0, Count = unitsCount };
var units = unitsClient.GetAllUnitsPaged(request);
```

При успешном выполнении units будет содержать все доступные ОМ.

## Описание структур данных используемых в примерах получения ОМ.

Пример **SpicObjectsChunkRequest** в JS выглядит следующим образом:

```
{
  "offset": 0,
  "count": 20
}
```

*Offset* – это смещение.

*Count* – количество ОМ

Пример **SpicUnitCollectionDto**:

```
{
  "Units": [{
    "Brand": "",
    "Color": "",
    "CompanyId": 264,
    "Description": "???? ?????????? ????????",
    "GarageNumber": "",
    "Model": "",
    "Name": "101503",
```

```

        "OlsonId": "",
        "Owner": "",
        "Power": "",
        "Registration": "",
        "StateNumber": "123456",
        "UnitId": 294,
        "UnitTypeId": 1,
        "VinNumber": "",
        "Year": ""
    }, {
        "Brand": "",
        "Color": "",
        "CompanyId": 264,
        "Description": "???? ?????????? ????????",
        "GarageNumber": "",
        "Model": "",
        "Name": "106641",
        "OlsonId": "",
        "Owner": "",
        "Power": "",
        "Registration": "",
        "StateNumber": "234567",
        "UnitId": 304,
        "UnitTypeId": 1,
        "VinNumber": "",
        "Year": ""
    }, {
        "Brand": "",
        "Color": "",
        "CompanyId": 10124,
        "Description": "",
        "GarageNumber": "",
        "Model": "",
        "Name": "????????????? 845",
        "OlsonId": "",
        "Owner": "",
        "Power": "",
        "Registration": "",
        "StateNumber": "",
        "UnitId": 10981,
        "UnitTypeId": null,
        "VinNumber": "",
        "Year": ""
    }
}

```

Результирующие классы для контрактов данных **SpicObjectsChunkRequest** и **SpicUnitCollectionDto** для **SOAP** можно посмотреть в в классах, сгенерированных после получения метаданных.

[<<Назад  
оглавлению](#)

[К](#)

[Далее>>](#)

