

## ГЕОИНФОРМАТИКА

УДК 004.6+004.94+681.518

DOI: 10.30533/0536-101X-2021-65-5-582-591

### К вопросу об использовании каталога DISCOS как источника для агрегирования данных в ГИС околоземного космического пространства. Часть 2

© 2021 г. П.Ю. Орлов\*, А.Е. Луговской

Московский государственный университет геодезии и картографии, Москва, Россия  
\*knight rider3e0@gmail.com

### On the application of DISCOS catalog as a source for data aggregation in GIS of near-Earth space. Part 2

P.Yu. Orlov\*, A.E. Lugovskoi

Moscow State University of Geodesy and Cartography, Moscow, Russia  
\*knight rider3e0@gmail.com

Received April 30, 2021

Revised October 08, 2021

Accepted October 15, 2021

**Keywords:** Near-Earth space, space object, space debris, C#, DISCOSweb API, MVVM, WPF, data parsing.

**Summary.** The first part of this article showed the DISCOS project, initiated by the European Space Agency. In order to use the data stored in this system within the GIS of near-Earth space prototype, it was decided to develop a software tool to extract, structure and store the information about space objects using the DISCOSweb API – DISCOSParser. Authorization on the European Space Agency server is performed using the OAuth 2.0 protocol, the data exchange is carried out via HTTP requests, and the query results are received as strings of JSON format. The development of the software tool prototype was conducted with the object-oriented programming language C#, the MVVM architectural pattern, the WPF graphical subsystem, as well as the Newtonsoft.JSON, Prism.WPF and RestSharp libraries. The user can create multi-parameter queries for single objects and their groups. Nevertheless, DISCOSweb API that retrieves the desired information does not provide a list of unique values for the required attribute. The DISCOSParser's interface is designed to facilitate the work with the database. The Russian translation of the content categories supplied with the corresponding comments allows the ordinary user to better understand the thematic content without specific knowledge. It is recommended to make requests to the DISCOS system once in every 3 seconds.

**Citation:** Orlov P.Yu., Lugovskoi A.E. On the application of DISCOS catalog as a source for data aggregation in GIS of near-Earth space. Part 2. *Izvestia vuzov «Geodesy and Aerophotosurveying»*. 2021, 65 (5): 582–591. [In Russian]. DOI:10.30533/0536-101X-2021-65-5-582-591.

Поступила 30 апреля 2021 г. После доработки 8 октября 2021 г. Принята к публикации 15 октября 2021 г.

**Ключевые слова:** околоземное космическое пространство, космический объект, космический мусор, C#, DISCOSweb API, MVVM, WPF, парсинг данных.

Представлены сведения о разработанном авторами программном средстве DISCOSParser, предназначенном для парсинга разнородных сведений Европейского космического агентства об объектах ракетно-космической техники из базы данных DISCOS, отмечено его конечное целевое назначение. Описаны структура и технологические особенности решения, показан синтаксис запросов к системе и её ответов. Продемонстрирована работа созданного макета DISCOSParser, а также указаны недостатки содержимого базы данных DISCOS, в том числе по результатам тестирования работы интерфейса программирования приложений DISCOSweb API.

**Для цитирования:** Орлов П.Ю., Луговской А.Е. К вопросу об использовании каталога DISCOS как источника для агрегирования данных в ГИС околоземного космического пространства. Часть 2 // Изв. вузов «Геодезия и аэрофотосъемка». 2021. Т. 65. № 5. С. 582–591. DOI:10.30533/0536-101X-2021-65-5-582-591.

## Введение

В первой части данной статьи [1] был проведён анализ и интегральная оценка базы данных (БД) Европейского космического агентства (ЕКА) DISCOS. Представление конечному пользователю её содержимого осуществляется с помощью информационной системы с доступом через web-портал DISCOSweb<sup>1</sup> на базе клиент-серверной архитектуры и web-сервисов RESTful. Данная система используется ЕКА в качестве единого источника справочной информации по каталогизированной ракетно-космической технике (РКТ). В базу данных включены общие сведения по космическим объектам (КО), данные по их разрушению, сведения по запускам и космодромам, ракетам-носителям (РН) и их ступеням, а также начальным и конечным параметрам орбиты. Конечная цель проводимых авторами исследований — разработка единой агрегирующей web-ориентированной геоинформационной системы околоземного космического пространства (ОКП). Предполагается, что она будет включать в себя сведения о КО из всех общедоступных источников, включая рассмотренный проект ЕКА. Далее речь пойдёт о разработке программного средства DISCOSParser для формирования программных запросов и получения сведений Европейского космического агентства о космических объектах из информационной системы DISCOS (рис. 1).

### Общие сведения по разработке

Интерфейс программирования приложений DISCOSweb API<sup>2</sup>, разработанный специ-

алистами ЕКА, позволяет сформировать запросы и получить сведения по всем представленным в БД DISCOS космическим объектам без обращения к web-интерфейсу DISCOSweb (отметим, что документация доступна только авторизованным пользователям портала), что очень удобно для автоматизированного извлечения данных по отдельным категориям объектов на регулярной основе. Для информационного наполнения перспективной ГИС ОКП принято решение разработать программное средство для извлечения структурированной информации с web-ресурса DISCOSweb, просмотра полученных ответов с сервера, их сохранения, а также возможностью стилизации и задания определённых параметров запроса.

С помощью разработанного макета можно получить справочную информацию о запуске космических аппаратов, их регистрации контролирующими организациями, ракетносителях, размерах, массе, форме, владельце космического объекта, а также истории изменения орбитальных параметров для всех когда-либо отслеживаемых и занесённых в каталоги объектов.

Обмен данными между базой данных DISCOS (далее — Сервер) и макетом (далее — Клиент) происходит с помощью HTTP запросов. Клиент обращается к серверу, передавая параметры запроса через URL-параметры. Сервер возвращает ответ по протоколу HTTP, внутри которого содержится строка формата JSON [2]. Строка представляет собой стра-

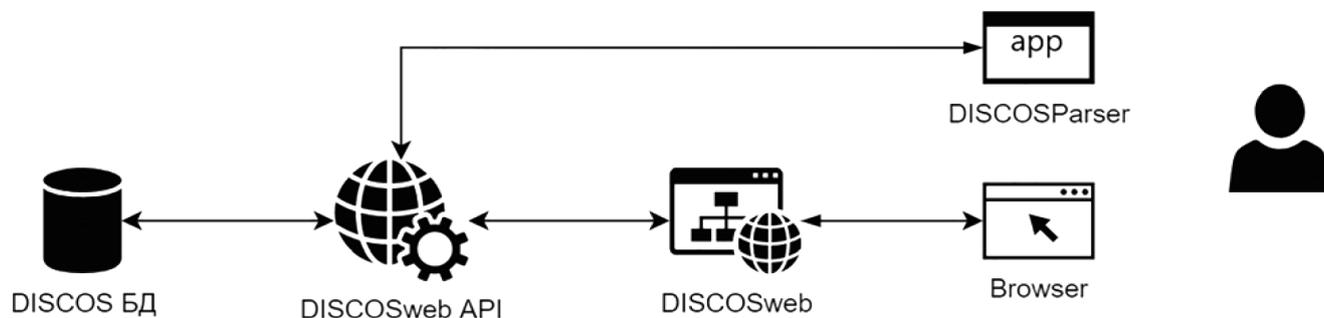


Рис. 1. Общая схема вариантов взаимодействия пользователя с БД DISCOS

Fig. 1. General scheme of user interaction options with DISCOS database

<sup>1</sup>Электронный ресурс: <https://discosweb.esoc.esa.int/>.

<sup>2</sup>Электронный ресурс: <https://discosweb.esoc.esa.int/apidocs>.

ницу с кортежем запрашиваемых объектов. Параметры страницы — количество объектов на странице и её порядковый номер — регулируются при отправке запроса.

Для взаимодействия с API необходимо провести авторизацию по протоколу OAuth 2.0 через уникальный ключ доступа (далее — Токен), который можно сгенерировать в личном кабинете портала DISCOSweb. При успешной авторизации будет получен временный ключ доступа. Его необходимо включать в каждый запрос в виде заголовка URL-Header. По результатам выполнения запроса информации через DISCOSweb API можно получить данные не только непосредственно о запрашиваемом объекте, но также и о связанных с ним объектах (аналогично связанным таблицам портала DISCOSweb).

### Технологии, использованные при разработке

Макет программного средства разрабатывался с помощью объектно-ориентированного языка программирования C# [3, 4] в среде Visual Studio 2019 для работы на компьютерах под управлением операционных систем (ОС) семейства Windows. При создании средства применён шаблон проектирования архитектуры приложений MVVM (Model-View-ViewModel), который позволяет отделить логику приложения от визуальной части с целью упрощения добавления нового функционала или стилизации имеющегося (рис. 2).

Элемент «Model» (модель) описывает используемые в приложении данные и содержит логику работы с ними. Элемент «View» (представление) описывает непосредственно пользовательский интерфейс. В свою очередь, «ViewModel» (модель представления) служит промежуточным звеном между двумя ранее описанными элементами — адаптером модели для взаимодействия с графическим интерфейсом [5].

При разработке также применялась подсистема WPF (Windows Presentation Foundation), которая приспособлена для использования шаблона MVVM [6]. Одно из главных отлич-

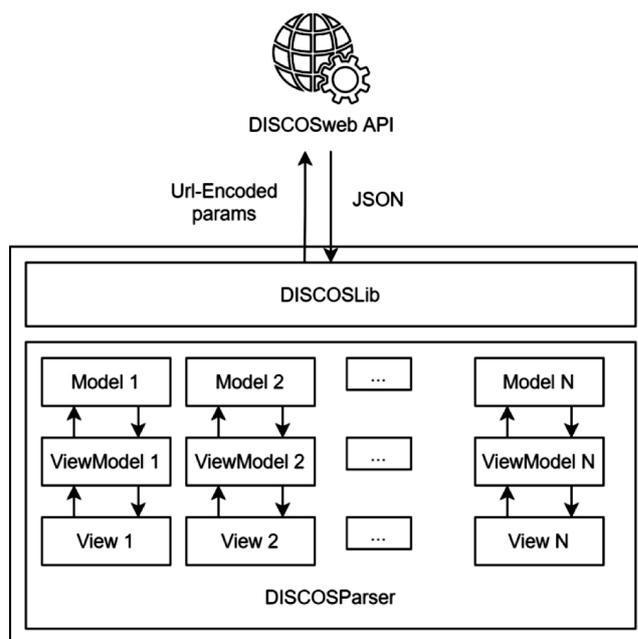


Рис. 2. Структура макета программного средства DISCOSParser, созданного с применением шаблона проектирования архитектуры приложений MVVM

Fig. 2. Layout of DISCOSParser software tool prototype designed with the MVVM software architectural pattern

чий от Windows Forms API, традиционно используемого при создании приложений для ОС Windows на платформе .NET, заключается в том, что интерфейс программы разрабатывается с помощью языка разметки XAML для декларативного программирования приложений, основанного на языке XML. Также WPF использует аппаратное ускорение при отрисовке интерфейса<sup>3</sup>, что позволяет снизить нагрузку на центральный процессор и сделать приложение более отзывчивым.

Макет разработанного программного средства использует три сторонние библиотеки, которые были добавлены в проект с помощью пакетного менеджера NuGet:

Newtonsoft.JSON — предназначена для сериализации / десериализации JSON-строк в объекты классов и наоборот [7];

Prism.WPF — фреймворк [8], применяется для упрощения разработки и дальнейшей поддержки WPF приложений, с шаблоном MVVM [9];

<sup>3</sup>Электронный ресурс: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/introduction-to-wpf?view=netframeworkdesktop-4.8>

RestSharp — обёртка над HTTPClient, позволяет удобно взаимодействовать с web-интерфейсами через протокол HTTP [10].

Минимальные системные требования макета определяет программная платформа .NET Framework, за счёт которой производится запуск. Макет программного средства работает на персональных компьютерах с процессором, поддерживающим инструкции архитектуры x86, разрядность которого составляет 32 или 64 бита. Тактовая частота процессора должна быть не ниже 1 ГГц, а объём оперативной памяти — не менее 512 Мбайт. Операционная система — Windows 7 с установленным пакетом обновлений Service Pack 1. На компьютере пользователя должна быть установлена программная платформа .NET Framework версии 4.7.2 или новее<sup>4</sup>.

### Библиотека DISCOSLib. Параметры запросов

Логика взаимодействия с базой данных DISCOS и графический интерфейс разделены на два проекта. Для взаимодействия с DISCOSweb API используется подключаемая DLL-библиотека DISCOSLib, разработанная в ходе проекта.

DISCOSLib является обёрткой над DISCOSweb API, через которую можно взаимодействовать с web-интерфейсом базы DISCOS. DISCOSLib реализована на языке программирования C# аналогично DISCOSParser. В библиотеке представлены все типы данных объектов, описанные в документации DISCOSweb API, что позволяет приводить ответ сервера к строгой структуре, упрощающей работу с данными (в качестве результата выполнения запроса выступает строка в формате JSON). Реализация отправки запроса и обработки ответа сервера скрыта от пользователя библиотеки. Также библиотека берёт на себя задачу по авторизации пользователя в DISCOSweb API:

а) реализован метод авторизации по протоколу OAuth 2.0 с использованием токена;

б) реализовано автоматическое включение временного ключа доступа в каждый запрос.

Перейдём к описанию подробностей взаимодействия с DISCOSweb API. Входными данными служат несколько строк, которые являются параметрами запроса. Для взаимодействия с сервером используется протокол HTTP. Запрос, отправляемый на сервер, может иметь поля, представленные в табл. 1.

Таблица 1  
Возможные параметры запроса  
на множество объектов

Поле	Пример содержимого
sort	-mass
filter	gt(mass, 200)
page[size]	10
page[number]	1
include	launch
fields[object]	cosparId, satno, name
fields[launch]	epoch

После выполнения запроса с указанными в таблице параметрами на URL-endpoint «/objects» (по этому адресу можно получить доступ к таблице со спутниками) сервер вернёт список спутников, которые имеют массу более 200 кг, при этом среди характеристик будут возвращены только международный идентификатор КОСПАР (COSPAR ID), номер по каталогу USSPACECOM/NORAD (NORAD ID) и наименование аппарата. Отсортирован список будет по массе в порядке убывания. Также в запросе будет содержаться информация о запуске аппаратов, а именно эпоха запуска.

Заполненные поля кодируются как URL параметры запроса. DISCOSweb API позволяет отправить запрос как на множество, так и на конкретный объект по его идентификатору (ID) в базе данных DISCOS.

При запросе на множество параметров (листинг 1) можно указать следующие параметры запроса:

1) sort — указываются атрибуты объекта, по которым необходимо проводить сортировку полученных объектов. Если параметр не указан, то сортировка будет проводиться по внутреннему ID базы данных DISCOS. Если нуж-

<sup>4</sup>Электронный ресурс: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/system-requirements>.

```

01: {
02:   "data": array of Data,
03:   "included": array of
Included,
04:   "links": Link,
05:   "meta": object
06: }

```

**Листинг 1. Структура ответа сервера при запросе множества объектов**

**Listing 1. Server's response structure when requesting multiple objects**

но отсортировать объекты в обратном порядке, то необходимо перед полем поставить знак «-». Поддерживается смешанная сортировка. В таком случае необходимо указать через запятую нужные атрибуты. Пример сортировки по массе: «-mass»;

2) filter — позволяет задать предикаты, по которым должна быть проведена выборка объектов в базе данных. Предусмотрена возможность использования встроенных функций, таких как:

- eq(field, value) — Равенство;
- ne(field, value) — Неравенство;
- lt(field, value) — Менее чем;
- gt(field, value) — Более чем;
- le(field, value) — Менее чем или равно;
- ge(field, value) — Более чем или равно;
- in(field, values) — Вхождение элемента (-ов) в заданном массиве;
- out(field, values) — Отсутствие элемента (-ов) в заданном массиве;
- contains(field, value) — Наличие подстроки с учётом регистра;
- excludes(field, value) — Отсутствие подстроки с учётом регистра;
- icontains(field, value) — Наличие подстроки без учёта регистра;
- iecludes(fields, value) — Отсутствие подстроки без учёта регистра.

Также допускается применение логических операторов, таких как «or» и «and». При формировании предиката возможно использование следующих значений.

1. Константы: «true», «false», «null».
2. Целые и действительные числа. Числа с плавающей запятой допускают запись в экспо-

ненциальной форме.

3. Строки. Кодировка ASCII. Допускаются следующие символы:

U+002D HYPHEN-MINUS, «-»; U+002E FULL STOP, «.»; U+005F LOW LINE, «\_»; U+007E TILDE, «~»; \ – backslash; \' – single quote; \b – ASCII backspace (BS); \f – ASCII formfeed (FF); \n – ASCII linefeed (LF); \r – ASCII carriage return (CR); \t – ASCII horizontal tab (TAB); \v – ASCII vertical tab (VT).

В запросе допускаются следующие типы данных:

- string — строка, заключенная в одинарные кавычки. Пример: '1960-011B';
- number — действительное число;
- boolean — логическое значение — true или false;
- epoch — эпоха, полученная с помощью приведения строки или числа к типу epoch.

Предусмотрено два способа приведения данных к типу epoch:

1. epoch:<number> – UNIX-время в секундах с начала эпохи – 01.01.1970 г., 00:00:00 UTC. Пример: «epoch: 951868800» – будет получена эпоха, соответствующая дате и времени: 01 марта 2000 г. 00:00:00;
2. epoch:<string> – дата, записанная в виде строки в соответствии со стандартом ISO 8601<sup>5</sup>. Пример: «epoch:'1957-10-04 19:29'» – эпоха, соответствующая дате и времени: 04 октября 1957 г. 00:00:00.

При составлении строки для параметра «filter» используется язык запросов RQL. Пример фильтрации объектов массой до 200 кг: «lt(mass,200)»:

page[size] — в связи с тем, что объектов может быть довольно много, они выдаются отдельными коллекциями в виде страниц. В это поле вносится количество объектов, которое должно быть в одной странице. По умолчанию этот параметр равен «30». Пример выдачи 10 объектов на одну страницу запроса: «10»;

page[number] — порядковый номер страницы, которую необходимо получить. По умол-

<sup>5</sup>ГОСТ ИСО 8601–2001 Система стандартов по информации, библиотечному и издательскому делу. Представление дат и времени. Общие требования. М. : Госстандарт России, 2001. 20 с.

Возможные параметры запроса на одиночный объект

Поле	Тип данных	Пример содержимого
fields [object]	Строковое значение	cosparId, mass

чанию параметр равен «1». Пример получения пятой страницы: «5»;

fields — данный параметр позволяет задать в запросе требуемые характеристики объекта, которые нужно получить. По умолчанию возвращаются все доступные характеристики. За счёт того, что этот параметр можно задавать отдельно для конкретных типов объектов, стилизации поддаются как непосредственно запрашиваемые объекты, так и дочерние им — вложенные объекты. К примеру, отправив запрос на получение информации о спутниках с параметром «fields[object]=cosparId», система вернёт объекты только с одной характеристикой — COSPAR ID.

Очевидно, что параметр «fields» можно применить только к тем объектам, которые будут содержаться в ответе на запрос и только к тем характеристикам, которые могут иметь объекты. Тем не менее, в документации к DISCOSweb API в описании параметра «fields» к каждому типу запроса указан одинаковый набор возможных объектов, что не соответствует действительности. Следует также отметить отсутствие средств, позволяющих получить список уникальных значений интересующего атрибута.

При запросе одиночного объекта (листинг 2) структура запроса имеет некоторые отличия. В данном случае необходимо обязательно указать ID объекта, информацию о котором необходимо получить. Пример параметров запроса на одиночный объект представлен в табл. 2.

```
01: {
02:   "data": Data,
03:   "included": array of
Included,
04:   "links": Link,
05:   "meta": object
06: }
```

**Листинг 2. Структура ответа сервера при запросе одиночного объекта**

**Listing 2. Server's response structure when requesting single object**

После выполнения запроса с указанными параметрами на URL-endpoint «/objects/44371» (подставляя после «objects/» необходимый ID, можно получить информацию о конкретных спутниках) сервер вернёт информацию о Международной космической станции (МКС), а именно: её COSPAR ID (1998-067A) и массу (450 000 кг).

### Демонстрация работы DISCOSParser

При запуске DISCOSParser потребуются ввести токен, сгенерированный в личном кабинете портала DISCOSweb (рис. 3, а). С инструкцией по получению токена можно ознакомиться, нажав на кнопку со знаком «?» в окне авторизации (рис. 3, б). Панель быстрого доступа к функционалу разработанного программного средства DISCOSParser представлена на рис. 4. На рис. 5 показан процесс редактирования запроса. Есть возможность выбрать определённые характеристики, по которым можно настроить фильтрацию или сортировку полученных объектов.

На рис. 6 приведены результаты выполненного пользовательского запроса в виде списка. На рис. 7 представлена полученная JSON-структура с информацией о ракете-носителе Восток (8K72) Л1-10, выведшем 19.08.1960 г. на орбиту КА Восток-1 № 2 (Спутник-5) с собаками Белкой и Стрелкой на борту. Нужно отметить, что ЕКА ошибочно относит этот объект к РН типа Восток (8A92M), пуски которых начались лишь с 1964 г.

Все полученные в результате выполнения запроса объекты можно сохранить в отдельном файле на пользовательском компьютере (рис. 8). Для этого необходимо добавить интересующий объект в список сохраняемых, нажав на кнопку с изображением иконки дискеты в правом верхнем углу рабочей области программы. После нажатия откроется окно сохранения

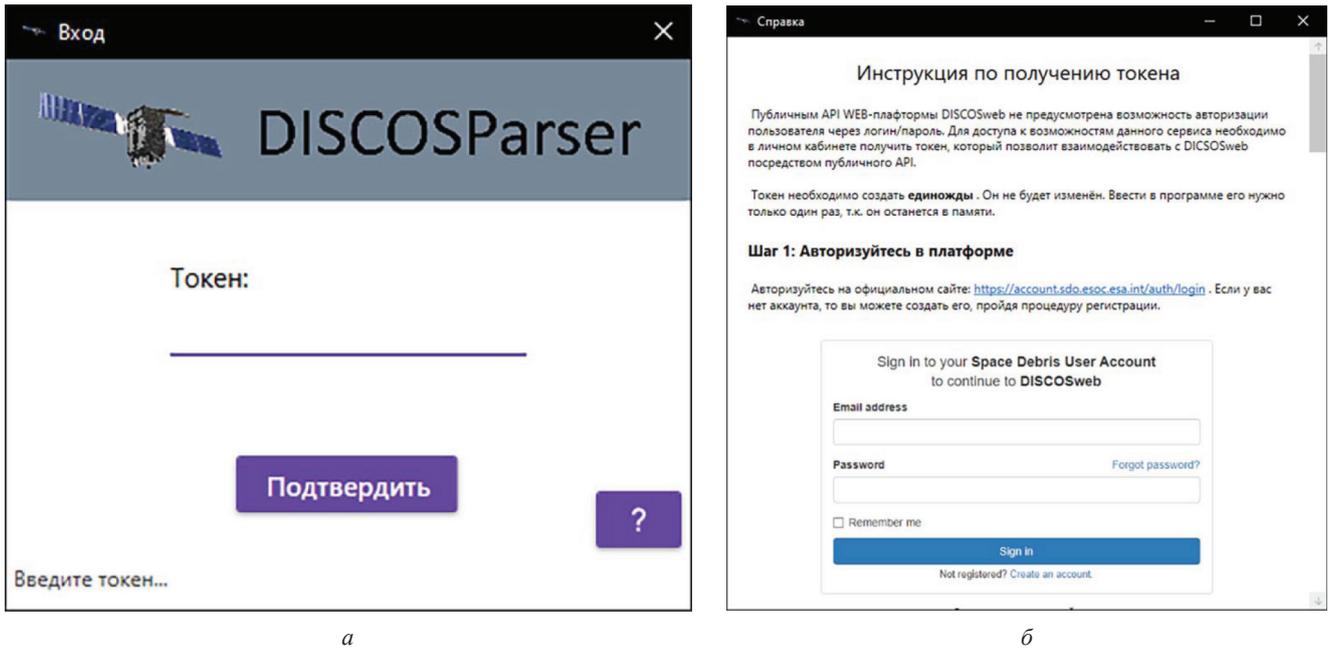


Рис. 3. Окно авторизации (а) и справки (б) макета программного средства DISCOSParser  
 Fig. 3. Authorization window (a) and help window (b) of DISCOSParser software tool prototype

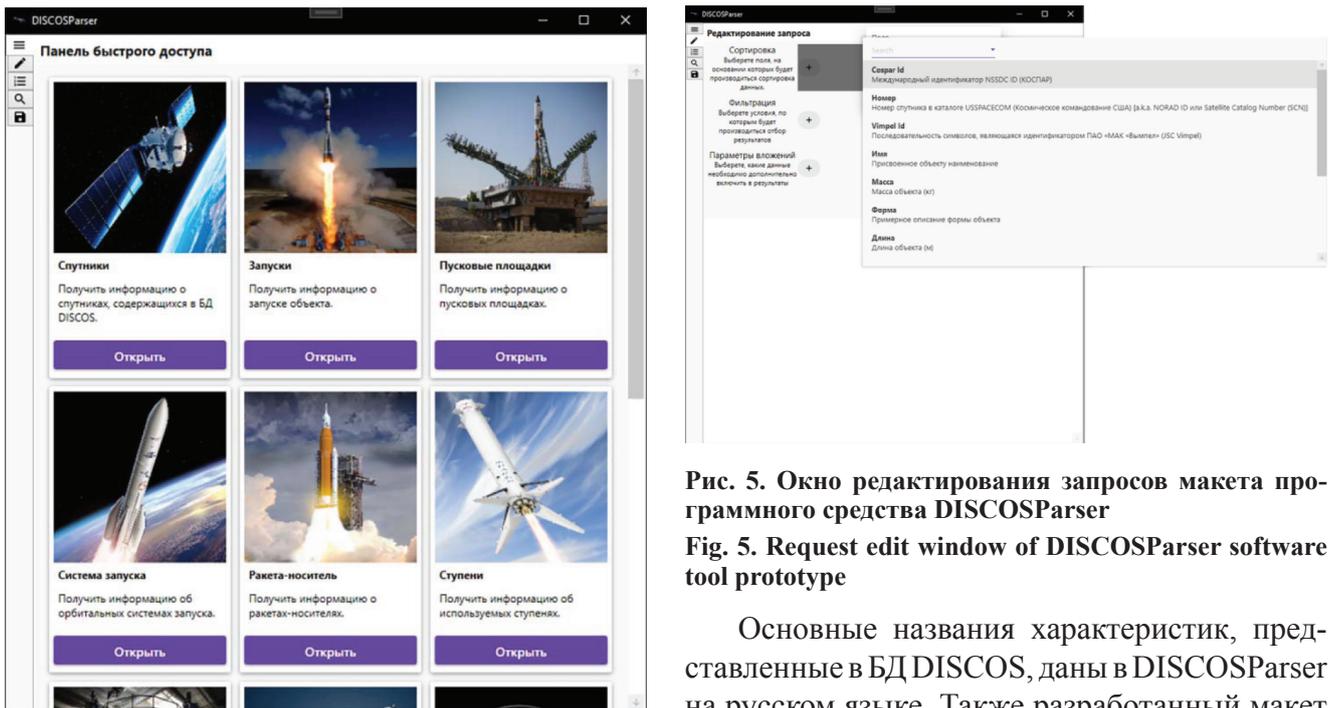


Рис. 4. Главное окно макета программного средства DISCOSParser  
 Fig. 4. Main window of DISCOSParser software tool prototype

запроса. Также пользователь может открыть ранее сохранённый файл с характеристиками космического объекта.

Рис. 5. Окно редактирования запросов макета программного средства DISCOSParser  
 Fig. 5. Request edit window of DISCOSParser software tool prototype

Основные названия характеристик, представленные в БД DISCOS, даны в DISCOSParser на русском языке. Также разработанный макет программы автоматически пытается перевести строку с описанием формы объекта на русский язык. Перевод осуществляется с помощью заранее составленного словаря сокращений. Если программное средство не находит в словаре совпадений, то будет выведено изначальное значение.

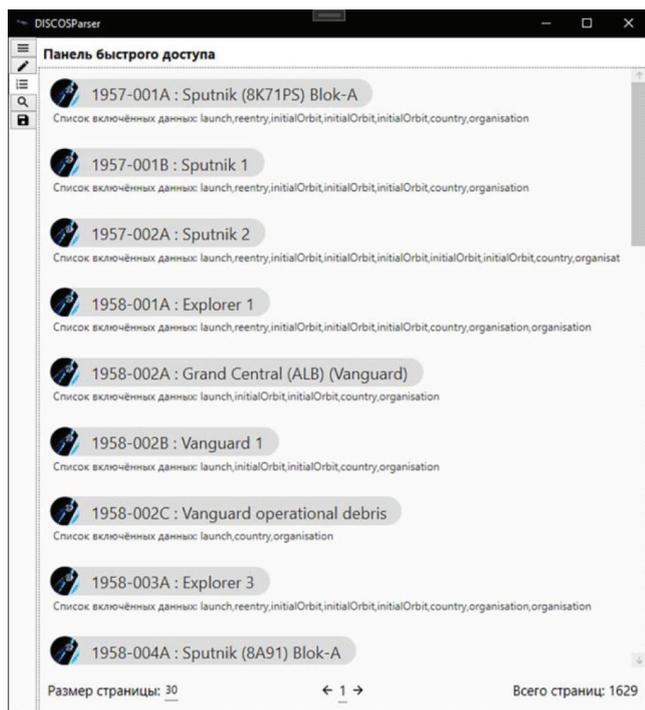


Рис. 6. Список полученных объектов в результате запроса

Fig. 6. List of received objects after request execution

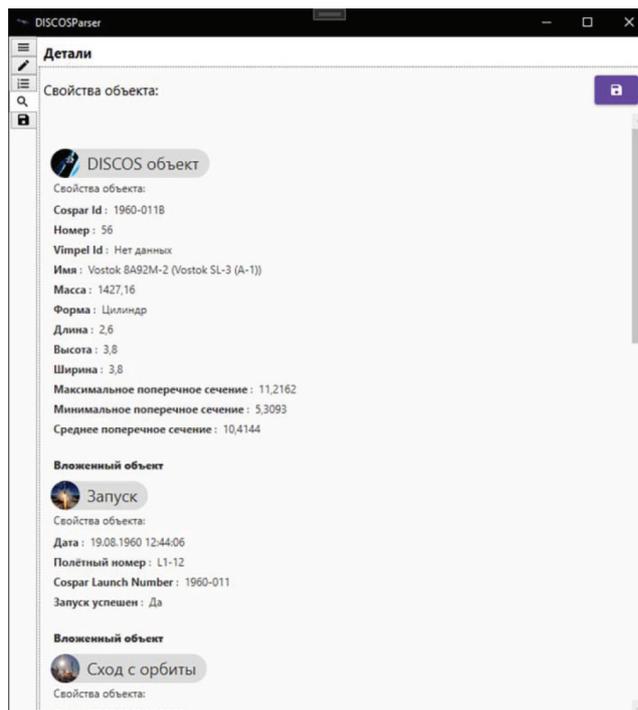


Рис. 7. Информация об объекте РН Восток (8K72) Л1-10

Fig. 7. Information about Vostok (8K72) L1-10 launch vehicle

### Доступ к DISCOSweb API

В полученном по результатам выполнения запроса ответе с сервера через HTTP-заголовки (HTTP-Header), кроме Cookie, также указывается информация, представленная в табл. 3.

Таблица 3

Данные, доступные через возвращаемые HTTP-заголовки

Название заголовка	Формат данных	Описание	Значение по умолчанию
X-RateLimit-Limit	Целое число	Общее доступное количество запросов для клиента.	60
X-RateLimit-Remaining	Целое число	Оставшееся доступное количество запросов для клиента.	20
X-RateLimit-Reset	Целое число (UNIX-Timestamp)	Время сброса количества доступных запросов до максимально доступного (задается через X-RateLimit-Limit).	Время последнего запроса + 60 секунд

Исходя из указанных в табл. 3 значений по умолчанию можно предположить, что DISCOSweb API позволяет отправлять до 20 запросов в минуту или один запрос в 3 с.

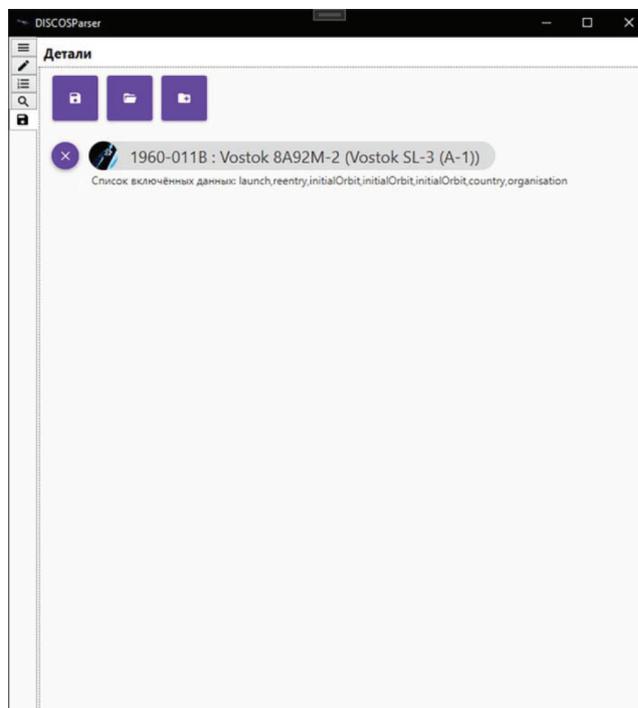


Рис. 8. Окно сохранения данных об объектах

Fig. 8. Object data save window

Для проверки данного предположения проведено тестирование на количество доступных пользователю запросов за определённый интервал времени. Тестирование проводилось с помощью ранее упомянутой библиотеки DISCOSLib.

Характеристики оборудования, на котором проводилось тестирование, следующие:

- ОС: Windows 10 Pro 20H2 x64;
- Процессор: Intel Xeon E3 1270 v3, 3,5 ГГц;
- Оперативная память: 16 Гбайт класса DDR3;
- Скорость Интернет-соединения: 100 Мбит/с.

В табл. 4 представлено время, ушедшее на обработку 20 запросов без задержки после каждого запроса. После отправки 20-го по счёту запроса доступ к API блокировался на 1 минуту.

Т а б л и ц а 4  
Результаты обращений к DISCOSweb API без тайм-аута

№ сеанса	Время, с	Среднее время на обработку запроса, мс	Количество запросов
1	9,683	484	20
2	8,344	417	
3	8,493	424	

В табл. 5 приведено время, ушедшее на обработку 20 запросов с ограничением (тайм-аутом) в 1 сек. после каждого запроса. Аналогично первому эксперименту, после отправки 20-го по счёту запроса доступ к API блокировался на 1 мин.

Т а б л и ц а 5  
Результаты обращений к DISCOSweb API с тайм-аутом в 1 секунду

№ сеанса	Время, с	Среднее время на обработку запроса, мс	Количество запросов
1	29,940	491	20
2	29,173	451	
3	29,936	487	

В табл. 6 представлено время, ушедшее на обработку 20 запросов с ограничением потока в 3 сек. после каждого запроса. В этом случае после отправки 20-го по счёту запроса доступ к API заблокирован не был.

Т а б л и ц а 6  
Результаты обращений к DISCOSweb API с тайм-аутом в 3 секунды

№ сеанса	Общее время, с	Среднее на обработку запроса, мс	Количество запросов
1	69,312	456	20
2	70,019	494	
3	70,120	499	

Таким образом, можно сделать вывод, что доступ к DISCOSweb API не будет блокироваться, если отправлять запросы не чаще одного раза в 3 с. У авторов есть небезосновательное предположение, что доступ портала DISCOSweb к базе данных космических объектов реализован с помощью того же DISCOSweb API, что объясняет сообщения о перегруженности системы в случае слишком частых запросов (перелистывании результатов поиска).

### Заключение

Представлены результаты разработки макета программного средства DISCOSParser, призванного упростить работу с ресурсом DISCOSweb. Использование DISCOSweb API позволяет собрать JSON-структуру различного наполнения по исследуемому объекту (-там) из составного запроса и сохранить нужную пользователю информацию без лишних затрат времени, что также удобно при внесении этих данных в перспективную единую агрегирующую web-ГИС околосемного космического пространства. В дальнейшем предполагается русификация категорий событий разрушений объектов в околосемном космическом пространстве, названий ракет-носителей и их семейств, космодромов и другой информации, требующей перевода на русский язык, а также комментариев специалистов. В перспективе рассматривается возможность получения орбитальных параметров TLE по идентификатору КОСПАР с целью формирования малых трёхмерных сцен (простейших визуализаций) в окне программы, наглядно поясняющих параметры орбиты исследуемых космических объектов. Рассматривается возможность переноса программного средства на персональные компьютеры под управлением других операционных систем.

**Благодарности.** Авторы выражают благодарность доктору технических наук, профессору, заведующему кафедрой геоэкологии РГУ нефти и газа (НИУ) имени И.М. Губкина, Виктору Геннадьевичу Аковецкому за идею в написании данной работы. Работа выполнена в рам-

ках государственного задания № 0708-2020-0001 Минобрнауки России.

**Acknowledgements.** The authors express their gratitude to Viktor Gennadievich Akovetsky, Doctor of Technical Sciences, Professor, Head of the Department of Geocology of the Gubkin

Russian State University of Oil and Gas (National Research University), for the idea of writing this paper. The research was carried out within the state assignment #0708-2020-0001 of the Ministry of Science and Higher Education of the Russian Federation.

## ЛИТЕРАТУРА

1. Орлов П.Ю., Луговской А.Е. К вопросу об использовании каталога DISCOS как источника для агрегирования данных в ГИС околоземного космического пространства. Часть 1 // Изв. вузов «Геодезия и аэрофото-съемка». 2021. Т. 65. № 4. С. 461–472. DOI:10.30533/0536-101X-2021-65-4-461-472.
2. Сейранян А.Т. С #.NET — доступ к методу REST Web API типа GET // Материалы VI Международной междисциплинарной конференции «Инновации и тенденции развития современной науки». Мадрид, Испания: Bubok Publishing S.L., 2021. С. 36–40.
3. Шарп Джон. Microsoft Visual C#. Подробное руководство. СПб.: Питер, 2017. 848 с.
4. Троелсен Эндрю, Джепикс Филипп. Язык программирования C# 7 и платформы .NET и .NET Core. СПб.: ООО «Диалектика», 2018. 1328 с.
5. Назаркин О.А. Разработка графического пользовательского интерфейса в соответствии с паттерном Model-View-Viewmodel на платформе Windows Presentation Foundation. Основные средства WPF. Липецк: Изд-во Липецкого государственного технического университета, 2014. 59 с.
6. Цирлин А.В., Южаков А.А., Верещагина Е.А. Исследование и сравнение современных подходов и паттернов разработки высокоорганизованных приложений // Материалы I Ежегодной международной научной конференции «Фундаментальная и прикладная наука: основные итоги 2015 г.». Северный Чарльстон, Южная Каролина, США : CreateSpace, 2015. С. 40–44.
7. Электронный ресурс: <https://www.newtonsoft.com/json/help>
8. Электронный ресурс: <https://prismlibrary.com/docs/>
9. Южаков А.А., Верещагина Е.А. Исследование современных методов разработки высокоорганизованных расширяемых модульных приложений, обзор и анализ существующих библиотек и MVVM фреймворков // Материалы IX молодежной международной научно-практической конференции студентов, аспирантов и молодых учёных «Шаг в будущее: теоретические и прикладные исследования современной науки». North Charleston, SC, USA : CreateSpace, 2015. С. 12–15.
10. Электронный ресурс: <https://restsharp.dev/api/>

## REFERENCES

1. Orlov P.Yu., Lugovskoi A.E. On the application of DISCOS catalog as a source for data aggregation into GIS of near-Earth space. Part 1. *Izvestia vuzov «Geodesy and Aerophotosurveying»*. 2021, 65 (4): 461–472. [In Russian]. DOI: 10.30533/0536-101X-2021-65-4-461-472.
2. Seyranyan A.T. C#.NET – access GET type REST Web API method. Proceedings of the VI International Multidisciplinary Conference «Prospects and Key Tendencies of Science in Contemporary World». Madrid, Spain : Bubok Publishing S.L., 2021: 36–40. [In Russian].
3. Sharp J. *Microsoft Visual C#. Podrobnoe rukovodstvo*. Microsoft Visual C#: Step by Step. St Petersburg : Piter, 2017: 848 p. [In Russian].
4. Troelsen A, Japikse P. *Yazyk programmirovaniya C# 7 i platformy .NET i .NET Core*. Pro C# 7: With .NET and .NET Core. St. Petersburg : ООО «Dialektika», 2018: 1328 p. [In Russian].
5. Nazarkin O.A. *Razrabotka graficheskogo pol'zovatel'skogo interfeisa v sootvetstvii s patternom Model-View-Viewmodel na platforme Windows*. User interface development in accordance to Model-View-Viewmodel pattern on Windows Presentation Platform. Primary WPF tools. Lipetsk: Izd-vo Lipetskogo gosudarstvennogo tekhnicheskogo universiteta, 2014: 59 p. [In Russian].
6. Tsirlin A.V., Yuzhakov A.A., Vereshchagina E.A. Research and comparison of modern approaches and patterns of highly organized applications development. Proceedings of the I Annual International Scientific Conference «Fundamental and applied sciences: the main results of 2015». North Charleston, SC, USA : CreateSpace, 2015: 40–44. [In Russian].
7. URL: <https://www.newtonsoft.com/json/help>.
8. URL: <https://prismlibrary.com/docs/>.
9. Yuzhakov A.A., Vereshchagina E.A. Research of highly organized extensible modular applications modern methods development, review and analysis of existing libraries and MVVM frameworks. Materials of the IX Youth International Scientific and Practical Conference of Students, Graduate Students and Young Scientists "Step into the Future: Theoretical and Applied Studies of Modern Science" North Charleston, SC, USA : CreateSpace, 2015: 12–15. [In Russian].
10. URL: <https://restsharp.dev/api/>