



# **Рекомендательные сервисы поддержки научно-технической и научно-образовательной деятельности**

**Руководство пользователя**

**(Информация, необходимая для эксплуатации программного  
обеспечения)**

## Оглавление

<b>1. Рекомендательные сервисы поддержки научно-технической и научно-образовательной деятельности .....</b>	<b>3</b>
<b>2. Функция рекомендации изданий для опубликования научных результатов на основе анализа полного текста препринта научной статьи .....</b>	<b>3</b>
2.1. Назначение метода .....	4
2.2. HTTP-запрос .....	4
2.3. Формат ответа.....	4
2.4. Ошибка валидации .....	6
<b>3. Функция подбора экспертов / рецензентов по заданному полным текстом объекту экспертизы.....</b>	<b>7</b>
3.1. Назначение метода .....	7
3.2. HTTP-запрос .....	7
3.3. Формат ответа.....	8
3.4. Ошибка валидации .....	9
<b>4. Функция подбора релевантных технологий по технологическим запросам.....</b>	<b>9</b>
4.1. Назначение метода .....	10
4.2. HTTP-запрос .....	10
4.3. Формат ответа.....	10
4.4. Ошибка валидации .....	11
<b>5. Функция интерактивного формирования и уточнение поисковых запросов с применением рекомендательной модели ассоциированных понятий предметных областей науки и техники.....</b>	<b>12</b>
5.1. Назначение метода .....	12
5.2. HTTP-запрос .....	12
5.3. Формат ответа.....	13
5.4. Ошибка валидации .....	13
<b>6. Функция автоматической классификации научно-технических и научно-образовательных документов по рубрикатору ГРНТИ.....</b>	<b>14</b>
6.1. Назначение метода .....	14
6.2. HTTP-запрос .....	14
6.3. Формат ответа.....	15
6.4. Ошибка валидации .....	16

## **1. Рекомендательные сервисы поддержки научно-технической и научно-образовательной деятельности**

Рекомендательные сервисы поддержки научно-технической и научно-образовательной деятельности (далее - Сервисы) предназначены для автоматизации процессов проведения исследований, разработки и выполнения экспертиз: поиска технологий, поиска изданий для опубликования результатов, поиск релевантных экспертов и рецензентов.

Сервисы предназначены для решения следующих задач:

- Подбор релевантных технологий по технологическим запросам (на основе смыслового сопоставления текстов научно-технической документации с технологическими запросами и техническими заданиями).
- Подбор экспертов / рецензентов по заданному полным текстом объекту экспертизы (грантовая заявка, текст подготовленной к печати научной статьи, диссертации и др.);
- Рекомендация изданий для опубликования научных результатов на основе анализа полного текста препринта научной статьи.
- Интерактивное формирование и уточнение поисковых запросов с применением рекомендательной модели ассоциированных понятий предметных областей науки и техники.
- Автоматическая классификация научно-технических и научно-образовательных документов по рубрикатору ГРНТИ.

Взаимодействие со всеми сервисами осуществляется через API по протоколу HTTP. Поддерживаемые методы — GET для операций получения данных и POST для операций отправки данных. Все запросы должны содержать корректные HTTP-заголовки. Успешные ответы от сервисов возвращаются с кодом состояния HTTP 200, а данные передаются в теле ответа в формате JSON. Структура запросов и описание параметров для каждой точки доступа приведены в соответствующих подразделах.

## **2. Функция рекомендации изданий для опубликования научных результатов на основе анализа полного текста препринта научной статьи**

Метод POST /journals предназначен для определения научных журналов, наиболее релевантных содержанию загруженной статьи. Сервис выполняет анализ текста документа, сопоставляя его тематическую направленность с профилем изданий, индексируемых в базах научной литературы. Результат представляет собой список журналов, отсортированный по степени релевантности.

Метод используется в сценариях подготовки публикаций, выбора подходящего научного издания, а также при предварительной редакционной экспертизе.

### 2.1. Назначение метода

Метод принимает полный текст научной статьи в виде бинарного файла (PDF, DOCX и др.), выполняет семантический анализ содержания и возвращает перечень журналов, релевантных тематике документа. Дополнительно для каждого возвращённого журнала указываются метаданные (издательство, ISSN, квартиль, индексируемые базы данных) и список статей данного журнала, наиболее похожих на загруженный текст.

### 2.2. HTTP-запрос

**Используемый метод и точка доступа:** POST /journals

**Параметры строки запроса:** отсутствуют

**Тело запроса:** multipart/form-data

Структура тела запроса представлена в таблице 1.

Таблица 1. Параметры тела запроса к функции определения журнала

Поле	Тип	Обязательный	Описание
file	binary (file)	Да	Исходный документ. Файл должен быть загружен как часть формы в бинарном формате

Пример запроса:

```
curl -X POST "https://example.com/journals" \  
-H "Accept: application/json" \  
-F "file=@article.pdf"
```

### 2.3. Формат ответа

При успешной обработке возвращается массив объектов, каждому из которых соответствует один рекомендованный журнал.

**Код 200** — Успешный ответ

**Тип содержимого (Content-type):** application/json

**Тип данных:** array<object>

Структура объекта журнала представлена в таблице 2

Таблица 2. Структура объекта журнала

Поле	Тип	Описание
title	string	Название журнала
publisher	string   null	Издательство
databases	array<string>	Перечень индексирующих баз данных (Scopus, Web of Science и др.)
issn	string   null	Печатный ISSN
eissn	string   null	Электронный ISSN
sjr	number   null	Показатель Scientific Journal Ranking (SJР)
quartile	integer   null	Квартиль SJР (Q1–Q4)
vak_category	integer   null	Категория ВАК
url	string	Ссылка на сайт журнала
neopisk_url	string	Ссылка на карточку журнала в системе «Неоиск»
documents	array<object>	Список статей, релевантных загруженному документу

Структура объекта документа в поле `documents` представлена в таблице 3

Таблица 3. Структура объекта документа журнала

Поле	Тип	Описание
title	string	Название релевантной статьи
urls	array<string>	Ссылки на источники публикации
neopisk_url	string	Ссылка на карточку документа в системе «Неоиске»
rate	number (0–1)	Оценка семантического сходства

Пример ответа:

[

```

{
  "title": "Journal of Applied Computational Science",
  "publisher": "TechPress",
  "databases": ["Scopus", "Web of Science"],
  "issn": "2411-1092",
  "eissn": "2411-1106",
  "sjr": 0.42,
  "quartile": 3,
  "vak_category": 1,
  "url": "https://jacscience.techpress.org",
  "neopoisk_url": "https://neopoisk.ru/catalogue/journal/1096",
  "documents": [
    {
      "title": "Neural Embedding Methods for Sparse Scientific
Corpora",
      "urls": [
        "https://jacscience.techpress.org/article/2023-04-15"
      ],
      "neopoisk_url": "https://neopoisk.ru/documents/884201",
      "rate": 0.87
    },
    ...
  ]
},
...
]

```

## 2.4. Ошибка валидации

### Код 422 — Validation Error

Возвращается при отсутствии файла, некорректном формате запроса или ошибках структуры данных.

Структура ответа представлена в таблице 4

Таблица 4. Структура ответа при ошибке Validation error

Поле	Тип	Описание
detail	array<object>	Список ошибок
detail[.].loc	array<string   integer>	Локация ошибки в запросе

detail[].msg	string	Текст ошибки
detail[].type	string	Код типа ошибки

### 3. Функция подбора экспертов / рецензентов по заданному полным текстом объекту экспертизы

Метод `POST /experts` предназначен для автоматизированного определения экспертов, чья профессиональная деятельность и публикации наиболее релевантны содержанию загруженного документа. Сервис выполняет семантическое сопоставление текста научного материала с массивом авторских публикаций, формируя ранжированный список возможных экспертов или рецензентов.

#### 3.1. Назначение метода

Метод используется для отбора специалистов, которые обладают компетенциями в предметной области, соответствующей тематике документа. Алгоритм анализирует текст входного файла, сопоставляет его с профилями авторов (их публикационной активностью) и возвращает список экспертов, отсортированный по степени релевантности.

#### 3.2. HTTP-запрос

**Используемый метод и точка доступа:** `POST /experts`

**Параметры строки запроса:** отсутствуют

**Тело запроса:** `multipart/form-data`

Структура тела запроса представлена в таблице 5.

Таблица 5. Параметры тела запроса к функции определения экспертов

Поле	Тип	Обязательный	Описание
file	binary (file)	Да	Исходный документ. Файл должен быть загружен как часть формы в бинарном формате

Пример запроса:

```
curl -X POST "https://example.com/experts" \
  -H "Accept: application/json" \
  -F "file=@document.docx"
```

### 3.3. Формат ответа

При успешной обработке возвращается массив объектов, содержащих сведения об экспертах и их релевантных публикациях.

**Код 200** — Успешный ответ

**Тип содержимого (Content-Type):** application/json

**Тип данных:** array<object>

Структура объекта эксперта представлена в таблице 6

Таблица 6. Структура объекта эксперта

Поле	Тип	Описание
name	string	Имя (ФИО) эксперта
documents	array<object>	Список публикаций, наиболее релевантных содержанию загруженного документа

Структура объекта публикации в поле documents представлена в таблице 7

Таблица 7. Структура объекта публикации эксперта

Поле	Тип	Описание
title	string	Название публикации
url	array<string>	Источники, содержащие публикацию
neopisk_url	string	Ссылка на карточку публикации в системе «Неопоиске».
rate	number (0–1)	Оценка семантического сходства с исходным документом.

Пример ответа:

```
[
  {
    "name": "Иванов Сергей Михайлович",
    "documents": [
      {
        "title": "Методы распределённого анализа больших массивов научных текстов",
        "url": [
```



```

        "https://cyberleninka.ru/article/n/145211",
        "https://journals.example.org/comp/"
    ],
    "neopisk_url": "https://neopisk.ru/documents/558201",
    "rate": 0.89
  },
  ...
]
},
...
]

```

### 3.4. Ошибка валидации

#### Код 422 — Validation Error

Возвращается при отсутствии файла, некорректном формате запроса или ошибках структуры данных.

Структура ответа представлена в таблице 8

Таблица 8. Структура ответа при ошибке Validation error

Поле	Тип	Описание
detail	array<object>	Список ошибок
detail[].loc	array<string   integer>	Локация ошибки в запросе
detail[].msg	string	Текст ошибки
detail[].type	string	Код типа ошибки

## 4. Функция подбора релевантных технологий по технологическим запросам

Метод POST /technology предназначен для определения технологических решений и патентных разработок, относящихся к тематике загруженного документа. Сервис выполняет семантический анализ текста и формирует перечень релевантных патентов/технологий с указанием правообладателей и степени тематического совпадения.

Метод применяется при анализе предварительной экспертизе технологических заявок и при поиске аналогов и имеющихся разработок.

#### 4.1. Назначение метода

Метод анализирует текст загруженного документа и сопоставляет его с базой патентных документов. В результате формируется список правообладателей и соответствующих им патентов, релевантных тематике документа. Каждый найденный патент содержит ссылку на источник и оценку семантического сходства.

#### 4.2. HTTP-запрос

**Используемый метод и точка доступа:** POST /technology

**Параметры строки запроса:** отсутствуют

**Тело запроса:** multipart/form-data

Структура тела запроса представлена в таблице 9.

Таблица 9. Параметры тела запроса к функции оценки технологического задела

Поле	Тип	Обязательный	Описание
file	binary (file)	Да	Исходный документ. Файл должен быть загружен как часть формы в бинарном формате

Пример запроса:

```
curl -X POST "https://example.com/technology" \  
-H "Accept: application/json" \  
-F "file=@report.pdf"
```

#### 4.3. Формат ответа

При успешной обработке метод возвращает массив объектов, содержащих сведения о правообладателях и релевантных патентных документах.

**Код 200** — Успешный ответ

**Тип содержимого (Content-Type):** application/json

**Тип данных:** array<object>

Структура объекта правообладателя представлена в таблице 10

Таблица 10. Структура объекта правообладателя

Поле	Тип	Описание
name	string	Правообладатель патентной разработки
documents	array<object>	Список релевантных патентов

Структура объекта патентного документа в поле `documents` представлена в таблице 11

Таблица 11. Структура объекта патентного документа

Поле	Тип	Описание
<code>title</code>	<code>string</code>	Название патентного документа
<code>url</code>	<code>array&lt;string&gt;</code>	Источники, содержащие патент
<code>neopoisk_url</code>	<code>string</code>	Ссылка на карточку патента в системе «Неопоиске».
<code>rate</code>	<code>number (0–1)</code>	Оценка семантического сходства с исходным документом.

Пример ответа:

```
[
  {
    "name": "АО «ТехИнновация»",
    "documents": [
      {
        "title": "Способ семантического анализа технических описаний
(патент RU2790451C1)",
        "url": [
          "https://patents.example.ru/RU2790451C1"
        ],
        "neopoisk_url": "https://neopoisk.ru/document/4484",
        "rate": 0.83
      }
    ]
  },
  ...
]
```

#### 4.4. Ошибка валидации

**Код 422** — Validation Error

Возвращается при отсутствии файла, некорректном формате запроса или ошибках структуры данных.

Структура ответа представлена в таблице 12

Таблица 12. Структура ответа при ошибке Validation error

Поле	Тип	Описание
detail	array<object>	Список ошибок
detail[].loc	array<string   integer>	Локация ошибки в запросе
detail[].msg	string	Текст ошибки
detail[].type	string	Код типа ошибки

## 5. Функция интерактивного формирования и уточнение поисковых запросов с применением рекомендательной модели ассоциированных понятий предметных областей науки и техники

Метод `GET /starsky` предназначен для вычисления набора близких слов и словосочетаний к заданному термину на основе семантических моделей. Результат представляет собой список терминов и их координаты в трёхмерном пространстве. Метод используется при формировании поисковых запросов.

### 5.1. Назначение метода

Метод принимает исходный термин (слово или словосочетание) и возвращает набор ближайших по смысловому расстоянию терминов. Каждый термин проецируется в трёхмерные координаты (x, y, z), что позволяет строить визуальные модели «звёздного неба» — облаков терминов, связанных по смыслу.

### 5.2. HTTP-запрос

**Используемый метод и точка доступа:** `GET /starsky`

**Параметры запроса:** передаются в URL (query parameters)

Параметры запроса представлены в таблице 13

Таблица 13. Параметры запроса к функции получения близких терминов

Поле	Тип	Обязательный	Описание	Ограничения
------	-----	--------------	----------	-------------

query	string	Да	Исходный термин	—
count	integer	Нет	Количество возвращаемых терминов	$\leq 1000$ , по умолчанию 100

Пример запроса:

```
curl -G "https://example.com/starsky" \
  --data-urlencode "query=семантическая модель" \
  --data-urlencode "count=2"
```

### 5.3. Формат ответа

Успешный ответ содержит массив объектов, каждый из которых представляет собой один термин с координатами.

**Код 200** — Успешный ответ

**Тип содержимого (Content-Type):** application/json

**Тип данных:** array<object>

Структура объекта близкого термина представлена в таблице 14

Таблица 14. Структура объекта близкого термина

Поле	Тип	Описание
x	number	Координата термина по оси X
y	number	Координата термина по оси Y
z	number	Координата термина по оси Z
text	string	Сам термин

Пример ответа:

```
[
  { "x": -0.21, "y": 0.44, "z": 0.11, "text": "векторная репрезентация" },
  { "x": -0.18, "y": 0.39, "z": 0.05, "text": "embedding-модель" }
]
```

### 5.4. Ошибка валидации

**Код 422** — Validation Error

Возвращается при отсутствии поля `query`, некорректном формате запроса или ошибках структуры данных.

Структура ответа представлена в таблице 15

Таблица 15. Структура ответа при ошибке Validation error

Поле	Тип	Описание
detail	array<object>	Список ошибок
detail[].loc	array<string   integer>	Локация ошибки в запросе
detail[].msg	string	Текст ошибки
detail[].type	string	Код типа ошибки

## 6. Функция автоматической классификации научно-технических и научно-образовательных документов по рубрикатору ГРНТИ

Метод `POST /grnti` выполняет автоматическую классификацию входного текста по рубрикатору Государственного рубрикатора научно-технической информации (ГРНТИ).

Сервис использует модель семантической классификации, определяющую наиболее вероятные тематические классы на основе анализа полного текста документа.

### 6.1. Назначение метода

Метод принимает произвольный текст (аннотация, полный текст статьи, фрагмент отчета и т.д.), анализирует его содержимое и формирует ранжированный список кодов ГРНТИ, отсортированный по вероятности соответствия.

Результат может использоваться: для подготовки публикаций, автоматического присвоения метаданных, агрегирования документов по тематическим направлениям.

### 6.2. HTTP-запрос

**Используемый метод и точка доступа:** `POST /grnti`

**Параметры строки запроса:** отсутствуют

**Тело запроса:** `application/json`

Параметры запроса представлена в таблице 16

Таблица 16. Параметры тела запроса к функции рубрикации ГРНТИ

Поле	Тип	Обязательный	Описание
text	string	Да	Текст документа, подлежащий классификации

Пример запроса:

```
curl -X POST "https://example.com/grnti" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "В работе анализируются методы обработки естественного
языка..."
  }'
```

### 6.3. Формат ответа

Успешный ответ содержит массив объектов, состоящих из кода класса ГРНТИ и вероятности его соответствия входному документу.

**Код 200** — Успешный ответ

**Формат данных ответа:** application/json

**Тип данных:** array<object>

Структура объекта класса ГРНТИ представлена в таблице 17

Таблица 17. Структура объекта класса ГРНТИ

Поле	Тип	Описание
label	string	Код класса ГРНТИ
score	number	Вероятность соответствия (0-1)

Пример ответа:

```
[
  {
    "label": "20.23.15",
    "score": 0.62
  },
  {
    "label": "20.53.19",
    "score": 0.47
  }
]
```

```
}  
]
```

#### 6.4. Ошибка валидации

##### Код 422 — Validation Error

Возвращается при отсутствии поля `text`, использовании неверного формата данных или передачи пустого текста.

Структура ответа представлена в таблице 18

Таблица 18. Структура ответа при ошибке Validation error

Поле	Тип	Описание
<code>detail</code>	<code>array&lt;object&gt;</code>	Список ошибок
<code>detail[].loc</code>	<code>array&lt;string   integer&gt;</code>	Локация ошибки в запросе
<code>detail[].msg</code>	<code>string</code>	Текст ошибки
<code>detail[].type</code>	<code>string</code>	Код типа ошибки