

Комитет по образованию Правительства Санкт-Петербурга  
**САНКТ-ПЕТЕРБУРГСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ**

**Отчет по практической работе**  
**МДК 01.02 «Разработка мобильных приложений»**  
**Разработка интерактивного приложения для**  
**анализа спектров**

Выполнил  
студент группы 493:  
Лукьянов И. А.  
Преподаватель: Фомин А.В.

Санкт-Петербург 2022

## Структура базы данных

База данных состоит из 4 таблиц:

1. DisplaySettings – хранит настройки отображения.
2. LastTag – хранит последний выбранный тэг.
3. LastPosition – хранит последнее место на спектре и последний выбранный химический элемент.
4. APIEndPoint – хранит конечную точку API.

ER диаграмма представлена на рисунке 1.

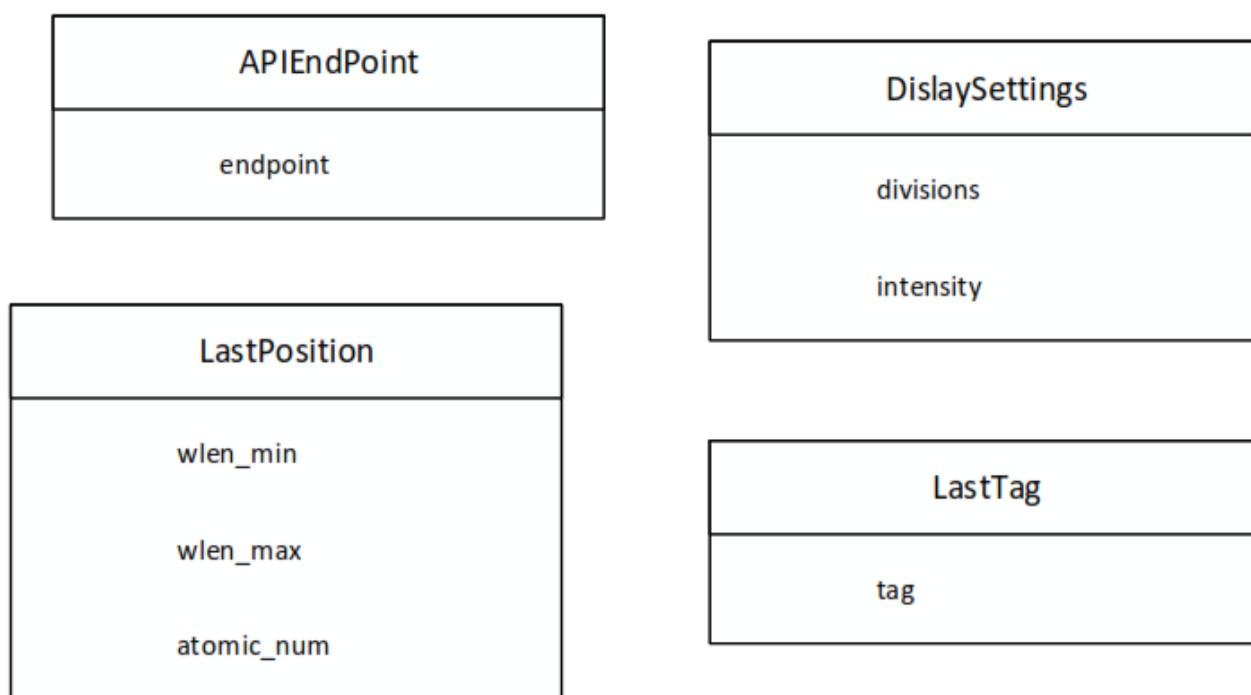


Рисунок 1 – ER диаграмма базы данных

## Таблица DisplaySettings

Содержит сведения о настройках отображения. Таблица состоит из двух столбцов:

1. divisions – вкл./выкл. отображение делений.
2. intensity – интенсивность фонового градиента.

Подробное описание столбцов представлено на рисунке 2.

<i>DisplaySettings</i>		<i>settings of display</i>						
#	name	type	size	default	primary	foreign	unique	description
1	divisions	int	-		no	-	no	enabled/disabled
2	intensity	int	-		no	-	no	intensity of gradient

Рисунок 2 – Описание столбцов таблицы DisplaySettings

## Таблица LastTag

Содержит сведения о последнем сохранённом тэге. Таблица состоит из одного столбца:

1. tag – тэг.

Подробное описание столбцов представлено на рисунке 3.

<i>LastTag</i>		<i>last saved tag</i>						
#	name	type	size	default	primary	foreign	unique	description
1	tag	text	-		no	-	no	tag name

Рисунок 3 – Описание столбцов таблицы LastTag

## Таблица LastPosition

Содержит сведения о последнем месте и масштабе на спектре, а также о последнем выбранном химическом элементе. Таблица состоит из трёх столбцов:

1. wlen\_min – минимальная длина волны.
2. wlen\_max – максимальная длина волны.
3. atomic\_num – выбранный химический элемент.

Подробное описание столбцов представлено на рисунке 4.

<i>LastPosition</i>		<i>last fixed position and atom</i>						
#	name	type	size	default	primary	foreign	unique	description
1	wlen_min	float	-		no	-	no	min wave length
2	wlen_max	float	-		no	-	no	max wave length
3	atomic_num	int	-		no	-	no	atomic number

Рисунок 4 – Описание столбцов таблицы LastPosition

## Таблица APIEndPoint

Содержит сведения о конечной точке API. Таблица состоит из одного столбца:

1. endpoint – конечная точка API.

Подробное описание столбцов представлено на рисунке 5.

<i>APIEndPoint</i>		<i>endpoint of api</i>						
#	name	type	size	default	primary	foreign	unique	description
1	endpoint	text	-		no	-	no	api endpoint

Рисунок 5 – Описание столбцов таблицы APIEndPoint

## Интерфейс приложения

Приложение состоит из 4 форм:

1. Main Form: служит для отображения спектра и изменения масштаба, а также выбора химического элемента.
2. Experiments Form: служит для отображения экспериментов по выбранному тэгу, также на форме имеется возможность изменения конечной точки API.
3. DisplaySettings Form: форма для задания настроек отображения спектра.
4. Tag Form: форма для выбора тэга.

### Форма Main Form

На рисунке 6 показан макет внешнего вида основной формы.

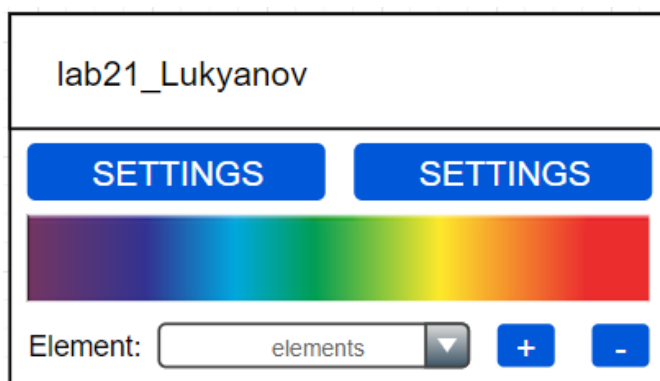


Рисунок 7 – Макет формы Main Form

На рисунке 7 показан внешний вид основной формы в приложении.

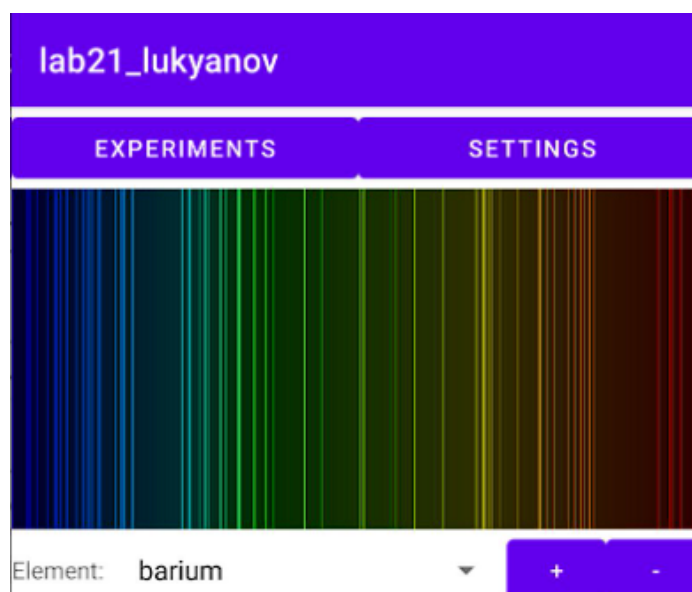
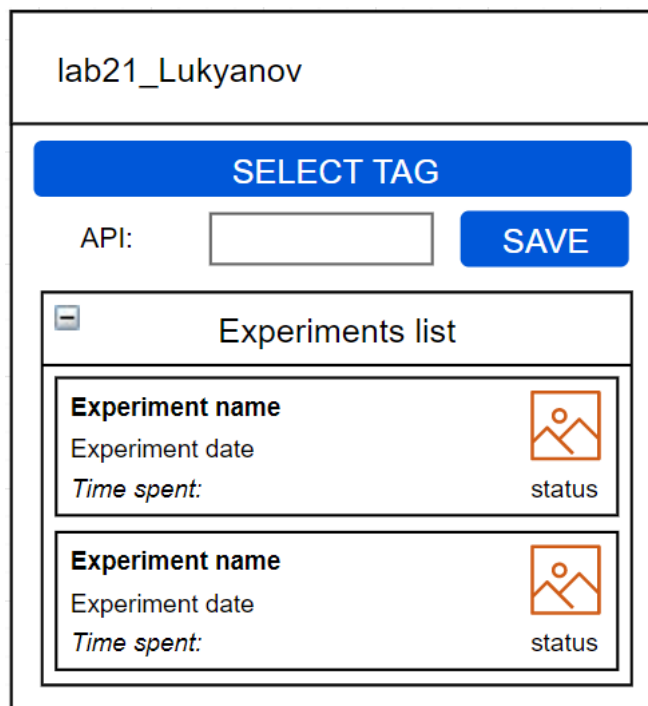


Рисунок 8 – Форма Main Form в приложении

## Форма Experiments Form

На рисунке 9 показан макет внешнего вида формы экспериментов.



lab21\_Lukyanov

**SELECT TAG**

API:  **SAVE**

**Experiments list**



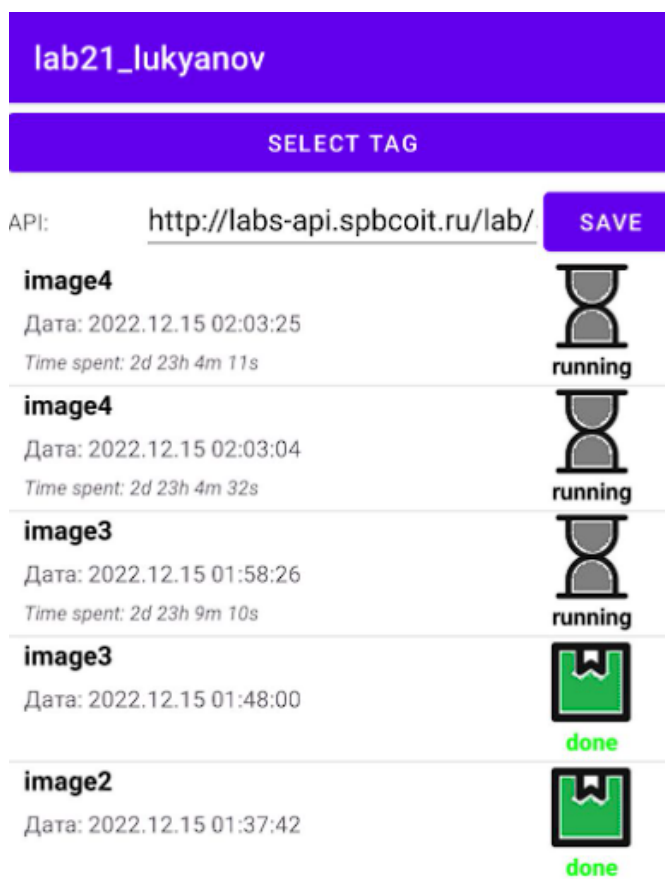
<b>Experiment name</b> Experiment date <i>Time spent:</i>	 status
<b>Experiment name</b> Experiment date <i>Time spent:</i>	 status

Рисунок 9 – Макет формы Experiments Form

На рисунке 10 показан внешний вид формы экспериментов в приложении.



lab21\_lukyanov

**SELECT TAG**

API: <http://labs-api.spbcoit.ru/lab/> **SAVE**






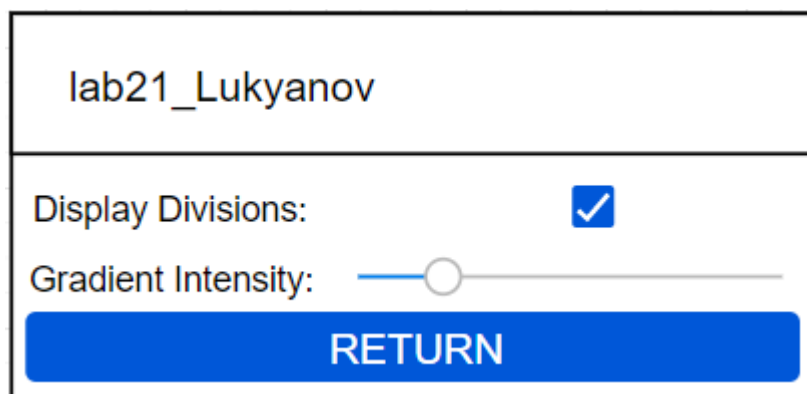
<b>image4</b> Дата: 2022.12.15 02:03:25 <i>Time spent: 2d 23h 4m 11s</i>	 running
<b>image4</b> Дата: 2022.12.15 02:03:04 <i>Time spent: 2d 23h 4m 32s</i>	 running
<b>image3</b> Дата: 2022.12.15 01:58:26 <i>Time spent: 2d 23h 9m 10s</i>	 running
<b>image3</b> Дата: 2022.12.15 01:48:00	 done
<b>image2</b> Дата: 2022.12.15 01:37:42	 done

Рисунок 10 – Форма Experiments Form в приложении

## Форма DisplaySettings Form


На рисунке 11 показан макет внешнего вида формы настроек отображения.



The mockup shows a form with a title bar containing the text "lab21\_Lukyanov". Below the title bar, there are two settings: "Display Divisions:" with a checked checkbox, and "Gradient Intensity:" with a slider control. At the bottom of the form is a large blue button labeled "RETURN".

Рисунок 11 – Макет формы DisplaySettings Form

На рисунке 12 показан внешний вид формы настроек отображения в приложении.

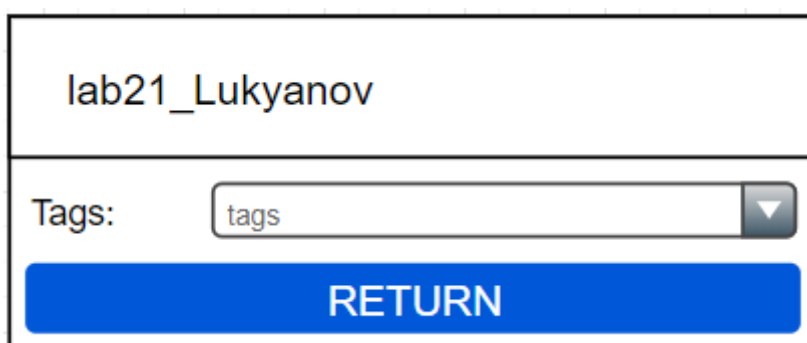


The screenshot shows the form in the application. The "Display Divisions:" checkbox is unchecked. The "Gradient Intensity:" slider is set to a value of 10, indicated by a red dot and the number "(10)" below the slider. The "RETURN" button is red.

Рисунок 12 – Форма DisplaySettings Form в приложении

## Форма Tag Form

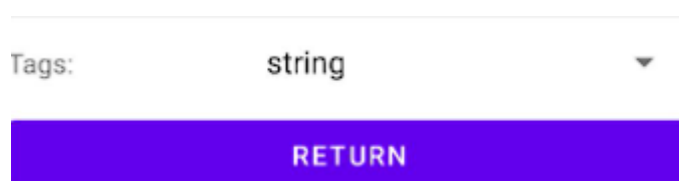
На рисунке 13 показан макет внешнего вида формы выбора тэга.



The mockup shows a form with a title bar containing the text "lab21\_Lukyanov". Below the title bar, there is a "Tags:" label followed by a text input field containing the word "tags" and a dropdown arrow. At the bottom of the form is a large red button labeled "RETURN".

Рисунок 13 – Макет формы Tag Form

На рисунке 14 показан внешний вид формы выбора тэга в приложении.



The screenshot shows the form in the application. The "Tags:" label is followed by a text input field containing the word "string" and a dropdown arrow. The "RETURN" button is red.

Рисунок 14 – Форма Tag Form в приложении

## Описание протокола взаимодействия

Для взаимодействия с базой данных мобильное приложение использует API на основе хранимых процедур и сервера, который организует вызов процедур по протоколу HTTP и обмен данными.

Всего доступно 12 функций, список которых представлен на рисунке 15.

1. /rpc/get\_lines – получение данных спектра по химическому элементу.
2. /rpc/run\_experiments – запуск эксперимента.
3. /rpc/get\_experiment\_data – получить данные эксперимента.
4. /rpc/get\_tags – получить все существующие тэги.
5. /rpc/get\_luminance\_profile – получение профиля яркости эксперимента.
6. /rpc/get\_status – получение статуса эксперимента.
7. /rpc/get\_color\_profile – получение цветового профиля эксперимента.
8. /rpc/get\_experiments – получение списка экспериментов по тэгу.
9. /rpc/nm\_to\_rgb\_range – получение цветов для заданной длины волны.
10. /rpc/set\_calibration – калибровка эксперимента
11. /rpc/get\_calibration – получение калибровки эксперимента.
12. /rpc/get\_elements – получение списка всех химических элементов.



(rpc) get_lines	^	
POST /rpc/get_lines	fetch spectral line data on given chemical element	▼
(rpc) run_experiment	^	
POST /rpc/run_experiment	upload new photo of spectra to database (jpeg in base64, max 4096 bytes) and extract spectral lines from given segment	▼
(rpc) get_experiment_data	^	
POST /rpc/get_experiment_data	get experiment parameters and image data	▼
(rpc) get_tags	^	
POST /rpc/get_tags	return all existing tags	▼
(rpc) get_luminance_profile	^	
POST /rpc/get_luminance_profile	return an array of pixel luminance values in range from 0.0 to 1.0 and their calibrated wavelength	▼
(rpc) get_status	^	
POST /rpc/get_status	get experiment current status (created/running/done)	▼
(rpc) get_color_profile	^	
POST /rpc/get_color_profile	return an array of pixel color values in range from 0.0 to 1.0 and their calibrated wavelength	▼
(rpc) get_experiments	^	
POST /rpc/get_experiments	fetch a list of all experiments with given tag	▼
(rpc) nm_to_rgb_range	^	
POST /rpc/nm_to_rgb_range	calculate a set of rgb colors for given wavelength range	▼
(rpc) set_calibration	^	
POST /rpc/set_calibration	set two calibration points marking point on a segment from 0.0 to 1.0 and corresponding wavelength	▼
(rpc) get_calibration	^	
POST /rpc/get_calibration	return two calibration points on a segment and their corresponding wavelength	▼
(rpc) get_elements	^	
POST /rpc/get_elements	fetch all available chemical elements	▼

Рисунок 15 – Список функций

## 1. Функция /rpc/get\_lines

Обеспечивает возможность получения спектра для заданного химического элемента.

Входных параметры:

— `atomic_num` – номер химического элемента.

Выходными параметрами будет массив данных со следующими характеристиками:

— `wavelength` – длина волны.

— `rel_intensity` – относительная интенсивность.

- red – значение красного цвета.
- green – значение зеленого цвета.
- blue – значение синего цвета.

Пример вызова функции показан на рисунке 16.

The screenshot shows a web interface for calling the `/rpc/get_lines` function. The interface is divided into several sections:

- Parameters:** A section for defining the request parameters. It includes a table with columns "Name" and "Description". The "args" parameter is defined as an "object" with the description "fetch spectral line data on given chemical element". The "args" parameter is required and has a value of `{ "atomic_num": 1 }`. The "Parameter content type" is set to `application/json`. There is a "Cancel" button.
- Execute:** A blue button to execute the request.
- Responses:** A section showing the response details. It includes a "Response content type" dropdown set to `application/json`.
- Curl:** A section showing the equivalent curl command: 

```
curl -X 'POST' \
  'http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get_lines' \
  -H 'accepts: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "atomic_num": 1
  }'
```
- Request URL:** A section showing the request URL: `http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get_lines`
- Server response:** A section showing the server response details. It includes a "Code" field with the value 200 and a "Details" field showing the response body: 

```
{
  "wavelength": 397.06700000000006,
  "rel_intensity": 0,
  "red": 0.716549999999999989,
  "green": 0,
  "blue": 0.283450000000000103
},
{
  "wavelength": 410.174,
  "rel_intensity": 15,
  "red": 0.716549999999999989,
  "green": 0,
  "blue": 0.283450000000000103
}
```

Рисунок 16 – Вызов функции `/rpc/get_lines`

## 2. Функция `/rpc/run_experiment`

Обеспечивает возможность запуска нового эксперимента. А именно загрузки фото спектра (максимальный размер – 4 Кб) в базу данных и выделение линий спектра для заданного сегмента.

Входные параметры:

- b64image – изображение в формате Base64
- note – название эксперимента.
- tag – тэг эксперимента, если такого не существует будет создан.
- x0 – начальная координата по горизонтали.
- x1 – конечная координата по горизонтали.
- y0 – начальная координата по вертикали.
- y1 – конечная координата по вертикали.

Выходные параметры:

- число – уникальный идентификатор созданного эксперимента.

Пример вызова функции показан на рисунке 17.

(rpc) run\_experiment

POST /rpc/run\_experiment upload new photo of spectra to database (jpeg in base64, max 4096 bytes) and extract spectral lines from given segment

Parameters

Name Description

args \* REQUIRED  
object  
(body)

Edit Value | Model

```
{
  "b64image": "string",
  "note": "string",
  "tag": "string",
  "x0": 0,
  "x1": 0,
  "y0": 0,
  "y1": 0
}
```

Cancel

Parameter content type  
application/json

Prefer  
string  
(header)  
Preference  
--

Execute Clear

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/run_experiment' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "b64image": "string",
    "note": "string",
    "tag": "string",
    "x0": 0,
    "x1": 0,
    "y0": 0,
    "y1": 0
  }'
```

Request URL

http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/run\_experiment

Server response

Code	Details
200	Response body

Рисунок 17 – Вызов функции /rpc/run\_experiment

### 3. Функция `/rpc/get_experiment_data`

Обеспечивает возможность получение данных эксперимента.

Входные параметры:

- `experiment` – уникальный идентификатор эксперимента.

Выходные параметры:

- `created_at` – дата и время начала эксперимента.
- `tag` – тэг эксперимента.
- `note` – название эксперимента.
- `status` – статус эксперимента (создан/в процессе/завершён).
- `b64image` – изображение в формате Base64.
- `x0` – начальная координата по горизонтали.
- `x1` – конечная координата по горизонтали.
- `y0` – начальная координата по вертикали.
- `y1` – конечная координата по вертикали.

Пример вызова функции показан на рисунке 18.

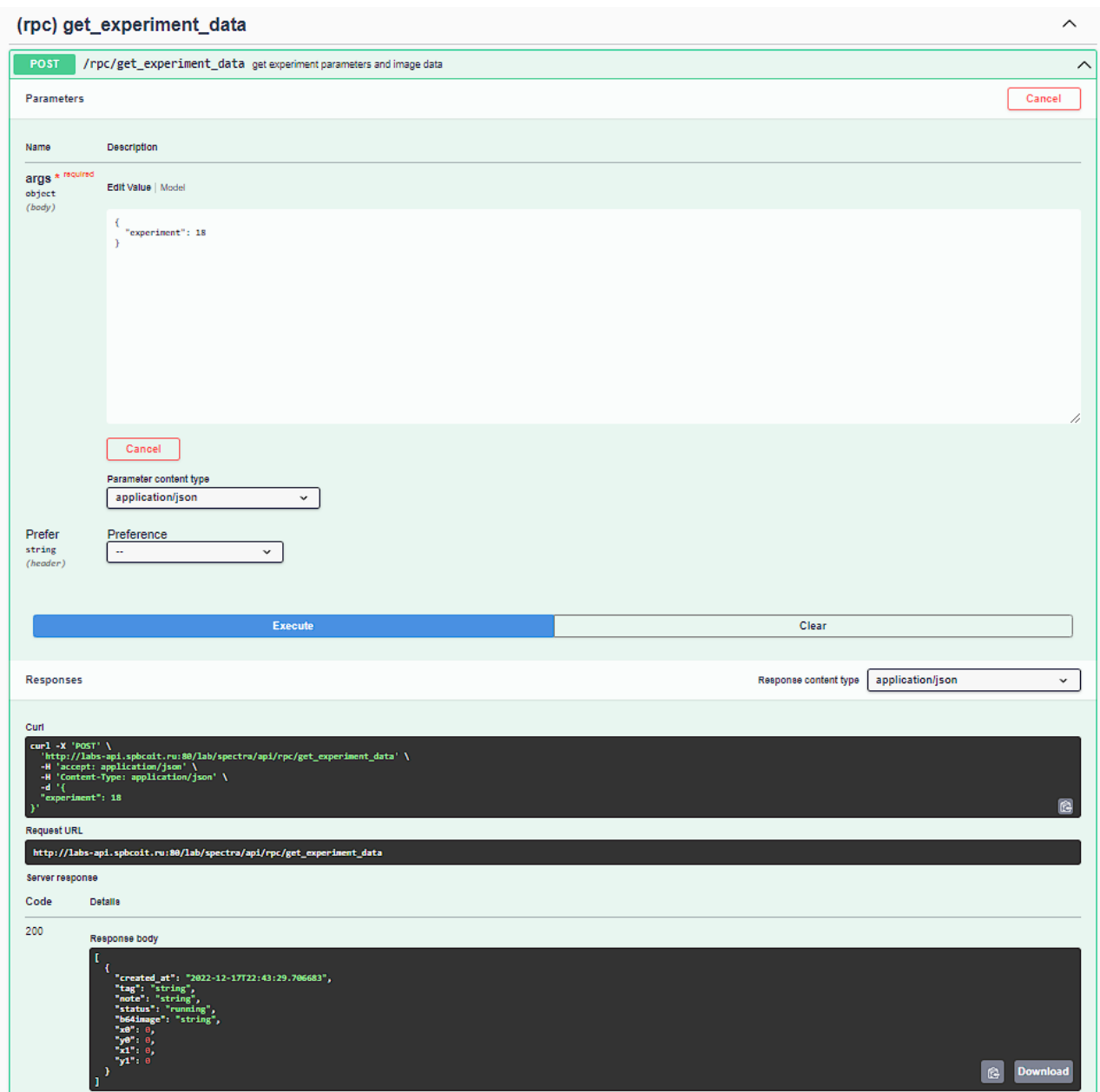


Рисунок 18 – Вызов функции /rpc/get\_experiment\_data

#### 4. Функция /rpc/get\_tags

Обеспечивает возможность получение списка всех существующих тэгов.

Входные параметров нет.

Выходными параметрами будет массив следующих элементов:

— строка – название тэга.

Пример вызова функции показан на рисунке 19.

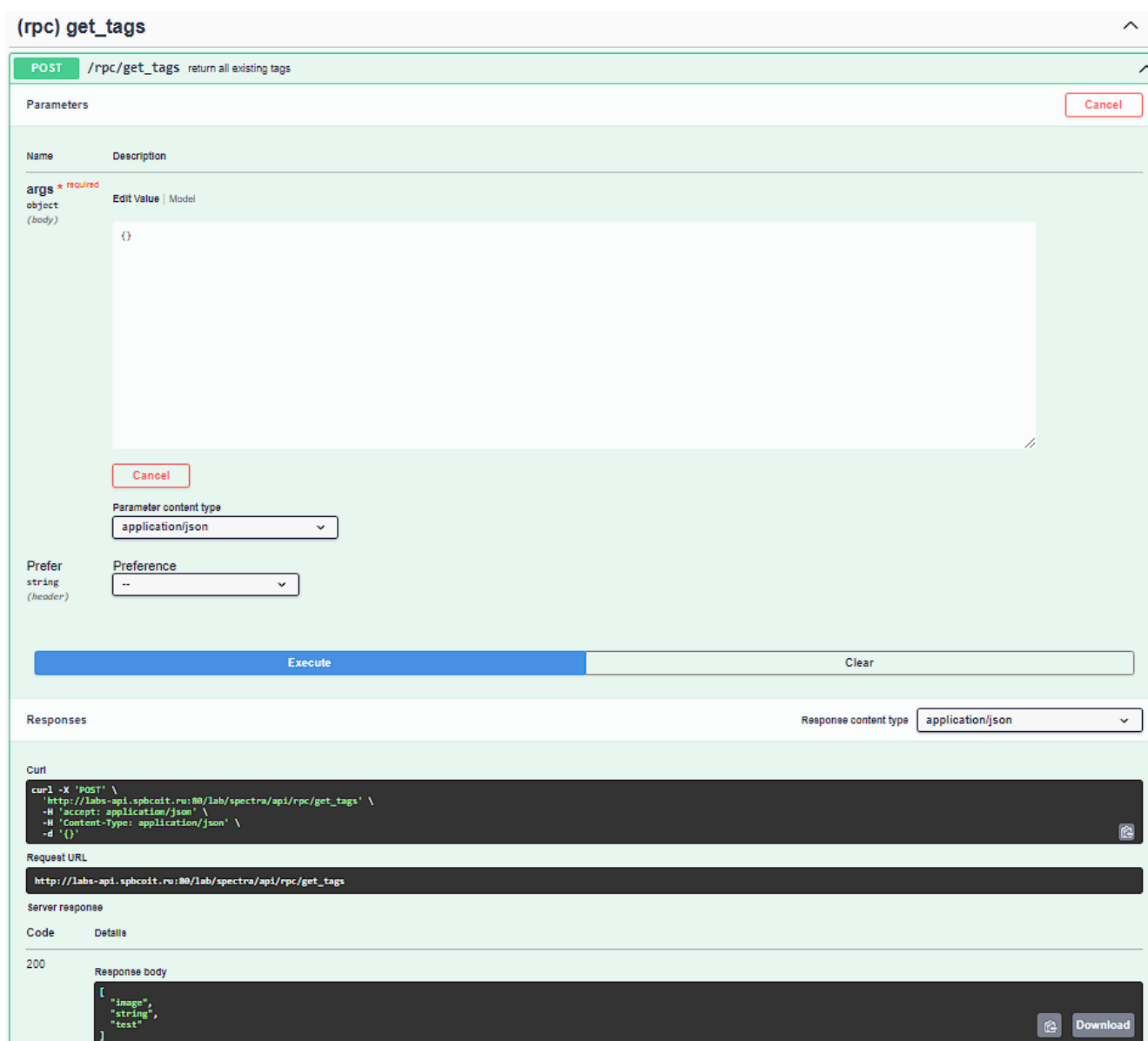


Рисунок 19 – Вызов функции `/rpc/get_tags`

## 5. Функция `/rpc/get_luminace_profile`

Обеспечивает возможность получения профиля яркости эксперимента, если данный эксперимент уже завершён.

Входные параметры:

— `experiment_id` – уникальный идентификатор эксперимента.

Выходными параметрами будет массив следующих элементов:

— `nm` – значение нанометров.

— `lum` – значение яркости.

Пример вызова функции показан на рисунке 20.

(rpc) get\_luminance\_profile

POST

/rpc/get\_luminance\_profile

return an array of pixel luminance values in range from 0.0 to 1.0 and their calibrated wavelength

Parameters

Name

Description

args \* required

object (body)

Edit Value | Model

```
{
  "experiment_id": 1
}
```

Cancel

Parameter content type

application/json

Prefer

string (header)

Preference

--

Execute

Clear

Responses

Response content type

application/json

Curl

```
curl -X 'POST' \
  'http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get_luminance_profile' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "experiment_id": 1
  }'
```

Request URL

http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get\_luminance\_profile

Server response

Code

Details

200

Response body

```
[
  {
    "nm": 380,
    "lum": 0.0104575163396928
  },
  {
    "nm": 380.39100604261973,
    "lum": 0.0104575163396928
  },
  {
    "nm": 380.7820136852395,
    "lum": 0.0104575163396928
  }
]
```

Рисунок 21 – Вызов функции /rpc/get\_luminance\_profile

## 6. Функция /rpc/get\_status

Обеспечивает возможность запуска получения статуса эксперимента.

Входные параметры:

— experiment – уникальный идентификатор эксперимента.

Выходные параметры:

— строка – статус эксперимента (создан/в процессе/завершён).

Пример вызова функции показан на рисунке 22.

15

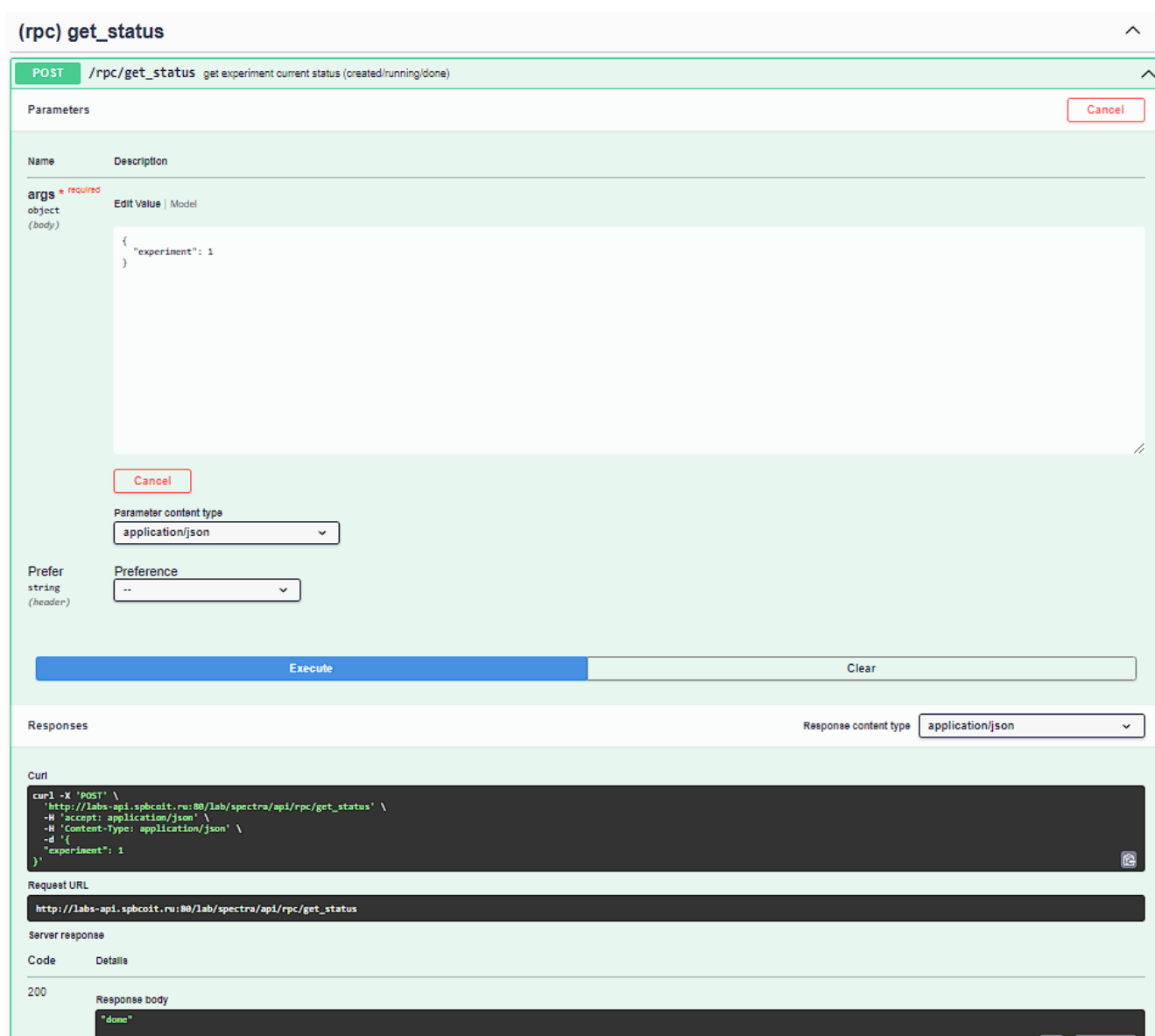


Рисунок 22 – Вызов функции `/rpc/get_status`

## 7. Функция `/rpc/get_color_profile`

Обеспечивает возможность получения профиля цвета эксперимента, если данный эксперимент уже завершён.

Входные параметры:

— `experiment_id` – уникальный идентификатор эксперимента.

Выходными параметрами будет массив следующих элементов:

- `nm` – значение нанометров.
- `red` – значение красного цвета.
- `green` – значение зелёного цвета.
- `blue` – значение синего цвета.



Пример вызова функции показан на рисунке 23.

**(rpc) get\_color\_profile**

**POST** /rpc/get\_color\_profile return an array of pixel color values in range from 0.0 to 1.0 and their calibrated wavelength

Parameters

Name Description

**args** \* required  
object  
(body)

Edit Value | Model

```
{  "experiment_id": 1}
```

Cancel

Parameter content type  
application/json

Prefer  
string  
(header)

Preference  
--

Execute Clear

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  'http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get_color_profile' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "experiment_id": 1
  }'
```

Request URL

http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get\_color\_profile

Server response

Code Details

200

Response body

```
[
  {
    "nm": 380,
    "red": 0.011764705882352941,
    "green": 0.00392156862745098,
    "blue": 0.01568627450980392
  },
  {
    "nm": 380.39190604261973,
    "red": 0.011764705882352941,
    "green": 0.00392156862745098,
    "blue": 0.01568627450980392
  },
  {
    "nm": 380.7820136852395,
```

Рисунок 23 – Вызов функции /rpc/get\_color\_profile

## 8. Функция /rpc/get\_experiments

Обеспечивает возможность запуска получения списка экспериментов по тэгу.

Входные параметры:

— tagname – тэг.

Выходными параметрами будет массив следующих элементов:

— id – уникальный идентификатор эксперимента.

- `created_at` – дата и время создания эксперимента.
- `note` – название эксперимента.
- `status` – статус эксперимента (создан/в процессе/завершён).

Пример вызова функции показан на рисунке 24.

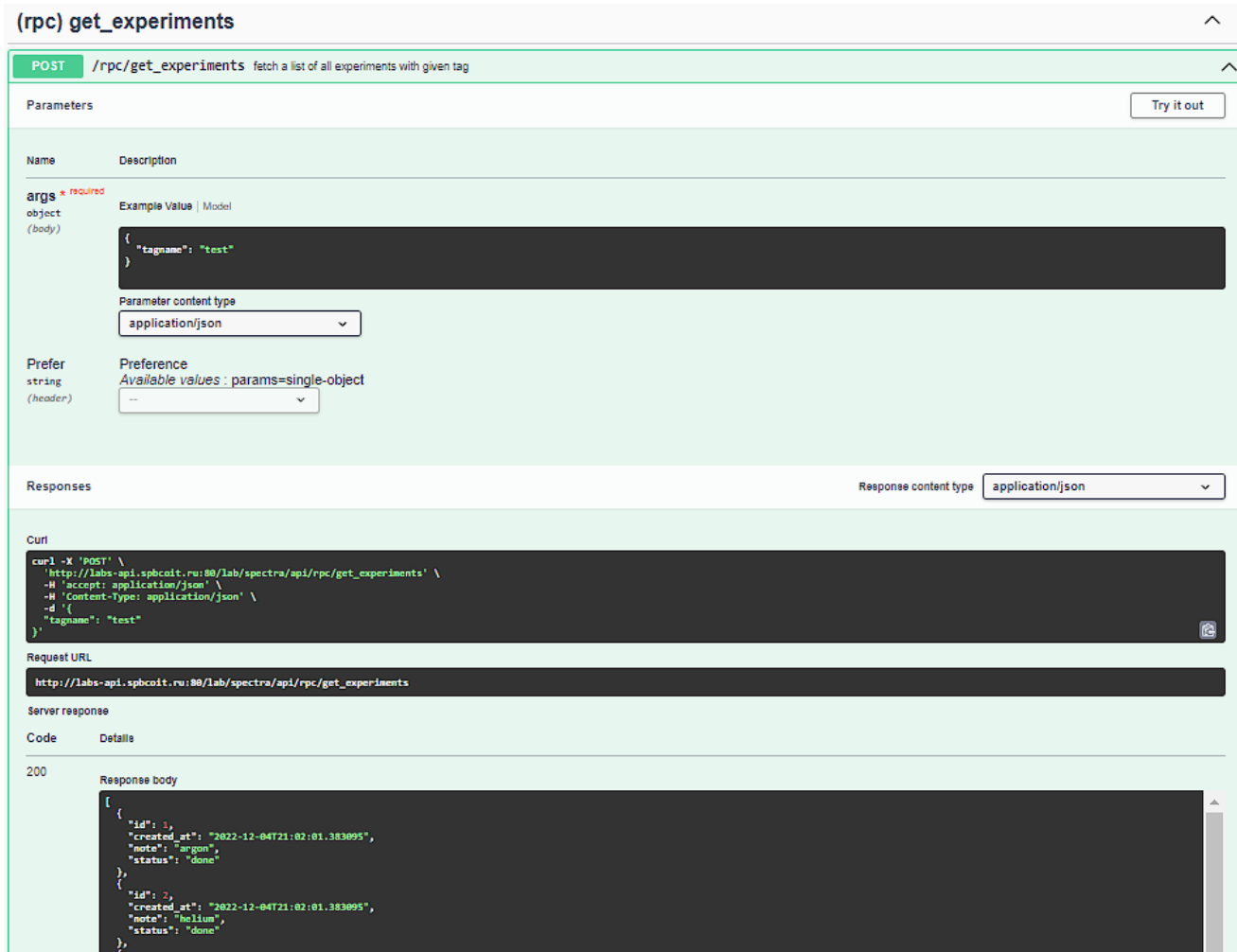


Рисунок 24 – Вызов функции `/rpc/get_experiments`

## 9. Функция `/rpc/nm_to_rgb_range`

Обеспечивает возможность расчета и получения набора цветов для заданной длины волны.

Входные параметры:

- `nm_from` – начало в нанометрах.
- `nm_to` – конец в нанометрах.
- `steps` – число шагов.

Выходными параметрами будет массив следующих элементов:

- `wavelength` – длина волны.

- red – значение красного цвета.
- green – значение зелёного цвета.
- blue – значение синего цвета.

Пример вызова функции показан на рисунке 25.

**(rpc) nm\_to\_rgb\_range**

**POST** /rpc/nm\_to\_rgb\_range calculate a set of rgb colors for given wavelength range

**Parameters**

**Name** **Description**

**args** \* required  
object  
(body)

**Edit Value** | **Model**

```
{
  "nm_from": 380,
  "nm_to": 780,
  "steps": 1220
}
```

**Cancel**

Parameter content type  
application/json

Prefer  
string  
(header)

Preference  
--

**Execute** **Clear**

**Responses** Response content type application/json

**Curl**

```
curl -X 'POST' \
  'http://labs-api.spbcnif.ru:80/lab/spectra/api/rpc/nm_to_rgb_range' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "nm_from": 380,
    "nm_to": 780,
    "steps": 1220
  }'
```

**Request URL**

http://labs-api.spbcnif.ru:80/lab/spectra/api/rpc/nm\_to\_rgb\_range

**Server response**

**Code** **Details**

200

**Response body**

```
[
  {
    "wavelength": 380,
    "red": 0,
    "green": 0,
    "blue": 0
  },
  {
    "wavelength": 380.3201378178835,
    "red": 0.9945310363686885,
    "green": 0,
    "blue": 0.005468963631391451
  }
]
```

Рисунок 25 – Вызов функции /rpc/nm\_to\_rgb\_range

## 10. Функция /rpc/set\_calibration

Обеспечивает возможность запуска калибровки для заданного сегмента и длины волны эксперимента.

Входные параметры:

- experiment – уникальный идентификатор эксперимента.
- new\_i\_max – конец сегмента.
- new\_i\_min – начало сегмента.
- new\_nm\_max – конец длины волны.
- new\_nm\_min – начало длины волны.

Выходные параметры:

- true/false – результат функции.

Пример вызова функции показан на рисунке 26.

(rpc) set\_calibration

POST /rpc/set\_calibration set two calibration points marking point on a segment from 0.0 to 1.0 and corresponding wavelength

Parameters

Cancel

Name	Description
args * required	Edit Value   Model
object (body)	<pre>{   "experiment": 1,   "new_i_max": 1.0,   "new_i_min": 0.0,   "new_nm_max": 780,   "new_nm_min": 380 }</pre>
	Cancel
Parameter content type	application/json
Prefer string (header)	Preference --
Execute Clear	

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/set_calibration' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "experiment": 1,
    "new_i_max": 1.0,
    "new_i_min": 0.0,
    "new_nm_max": 780,
    "new_nm_min": 380
  }'
```

Request URL

```
http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/set_calibration
```

Server response

Code	Details
200	<div>Response body</div> <pre>true</pre>

Рисунок 26 – Вызов функции /rpc/set\_calibration

## 11. Функция /rpc/get\_calibration

Обеспечивает возможность возврата последней калибровки эксперимента.

Входные параметры:

— experiment – уникальный идентификатор эксперимента.

Выходными параметрами будет массив следующих элементов:

— i\_max – конец сегмента.

— i\_min – начало сегмента.

— nm\_max – конец длины волны.

— nm\_min – начало длины волны.

Пример вызова функции показан на рисунке 27.

The screenshot displays a REST client interface for the endpoint `(rpc) get_calibration`. The request is a POST method with a JSON body containing `{ "experiment": 1 }`. The response is a 200 status code with a JSON body containing an array of calibration points: `[ { "i_min": 0, "nm_min": 380, "i_max": 1, "nm_max": 780 } ]`.

**Parameters**

Name	Description
args	required object (body)

Parameter content type: `application/json`

Prefer string (header): `--`

**Responses**

Response content type: `application/json`

**Curl**

```
curl -X 'POST' \
  'http://labs-api.spbcit.ru:80/lab/spectra/api/rpc/get_calibration' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "experiment": 1
  }'
```

**Request URL**

```
http://labs-api.spbcit.ru:80/lab/spectra/api/rpc/get_calibration
```

**Server response**

Code	Details
200	<p>Response body</p> <pre>[   {     "i_min": 0,     "nm_min": 380,     "i_max": 1,     "nm_max": 780   } ]</pre>

Рисунок 27 – Вызов функции /rpc/get\_calibration

## 12. Функция /rpc/get\_elements

Обеспечивает возможность получения списка всех химических элементов нового эксперимента. А именно загрузки фото спектра (максимальный размер – 4 Кб) в базу данных и выделение линий спектра для заданного сегмента.

Входные параметров нет.

Выходными параметрами будет массив следующих элементов:

- `atomic_num` – номер химического элемента.
- `full_name` – полное имя элемента.

Пример вызова функции показан на рисунке 28.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** /rpc/get\_elements (Description: fetch all available chemical elements)
- Parameters:** A table with columns 'Name' and 'Description'. The 'args' parameter is listed as 'object (body)' with a value of '{}'. Below the table, there are dropdowns for 'Parameter content type' (set to 'application/json') and 'Preference' (set to '--').
- Buttons:** 'Execute' and 'Clear' buttons are visible.
- Responses:** A section showing the response details.
  - Response content type:** application/json
  - Curl:**

```
curl -X 'POST' \
  'http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get_elements' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{}'
```
  - Request URL:** http://labs-api.spbcoit.ru:80/lab/spectra/api/rpc/get\_elements
  - Server response:** Code 200, Status OK.
  - Response body:** A JSON array of elements:

```
[
  {
    "atomic_num": 13,
    "full_name": "aluminum"
  },
  {
    "atomic_num": 19,
    "full_name": "argon"
  },
  {
    "atomic_num": 56,
    "full_name": "barium"
  }
]
```

Рисунок 28 – Вызов функции /rpc/get\_elements

## Демонстрация работы приложения

При запуске приложения отображается список экспериментов по последнему выбранному тэгу (рис. 29):

The screenshot shows the application interface for the tag 'lab21\_lukyanov'. At the top, there is a purple header with the tag name. Below it is a 'SELECT TAG' button. The main section displays a list of experiments with the following details:

Experiment Name	Date	Status
argon	Дата: 2022.12.05 00:02:01	done
helium	Дата: 2022.12.05 00:02:01	done
hydrogen	Дата: 2022.12.05 00:02:01	done
krypton	Дата: 2022.12.05 00:02:01	done
mercury	Дата: 2022.12.05 00:02:01	done
neon	Дата: 2022.12.05 00:02:01	done
nitrogen	Дата: 2022.12.05 00:02:01	done
xenon	Дата: 2022.12.05 00:02:01	done

Рисунок 29 – Список экспериментов

Изменим конечную точку API (рис. 30):

The screenshot shows the application interface with the API endpoint being edited. The 'API:' field now contains 'http://labs-api.spbcoitru/lab/ε'. The 'SAVE' button is visible next to the field. The list of experiments is partially visible below:

Experiment Name	Date	Status
argon	Дата: 2022.12.05 00:02:01	done
helium	Дата: 2022.12.05 00:02:01	done
hydrogen	Дата: 2022.12.05 00:02:01	done

Рисунок 30 – Изменение конечной точки API

Теперь проверим загрузку тэгов (рис. 31):

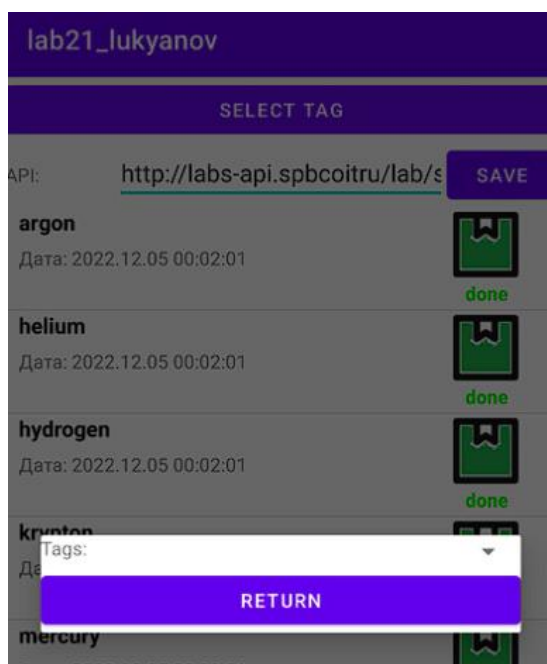


Рисунок 31 – Загрузка тэгов по новой конечной точке API

Имеется возможность выбрать один из существующих тэгов, при этом список экспериментов изменяется до выхода из диалогового окна (рис. 32):

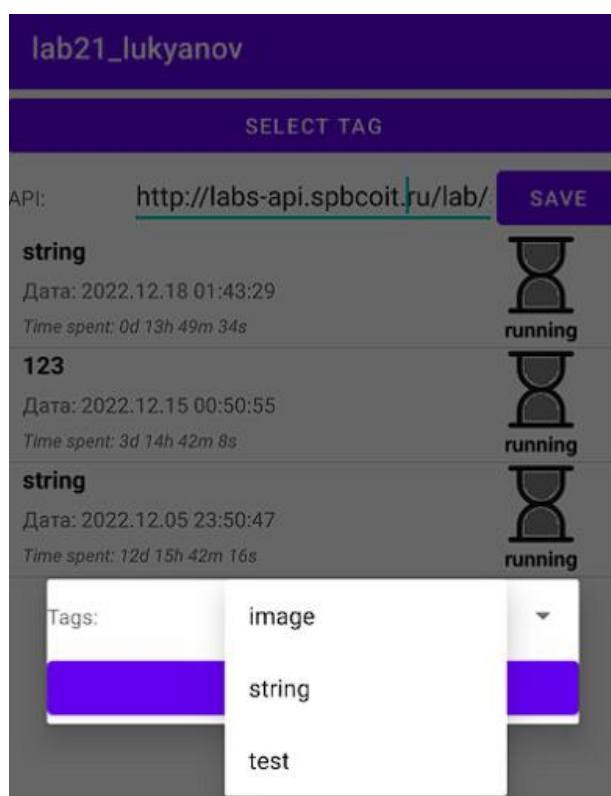


Рисунок 32 – Список тэгов



Эксперименты со статусами «создан» и «в процессе» обновляются каждую секунду и содержат информацию о пройденном с момента создания времени (рис. 33):





lab21_lukyanov		
SELECT TAG		
API:	<a href="http://labs-api.spbcoit.ru/lab/">http://labs-api.spbcoit.ru/lab/</a>	SAVE
<b>image4</b>	Дата: 2022.12.15 02:03:25 Time spent: 3d 13h 30m 39s	 running
<b>image4</b>	Дата: 2022.12.15 02:03:04 Time spent: 3d 13h 31m 0s	 running
<b>image3</b>	Дата: 2022.12.15 01:58:26 Time spent: 3d 13h 35m 38s	 running
<b>image3</b>	Дата: 2022.12.15 01:48:00	 done

Рисунок 33 – Не завершённые эксперименты

При выборе элемента из списка запускается форма спектров с последним сохранённым химическим элементом и настройками отображения (рис. 34):

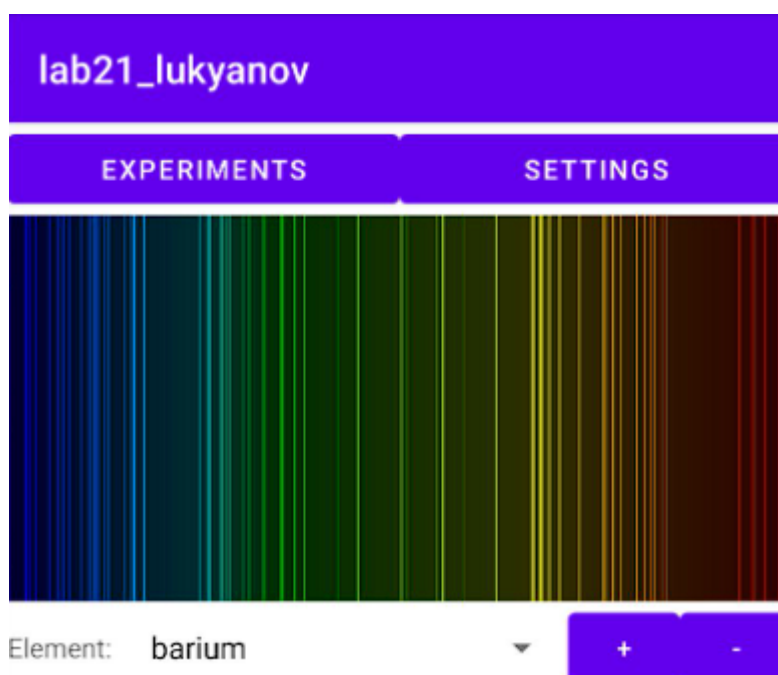


Рисунок 34 – Форма спектров

Имеется возможность перемещения по спектру (рис. 35):

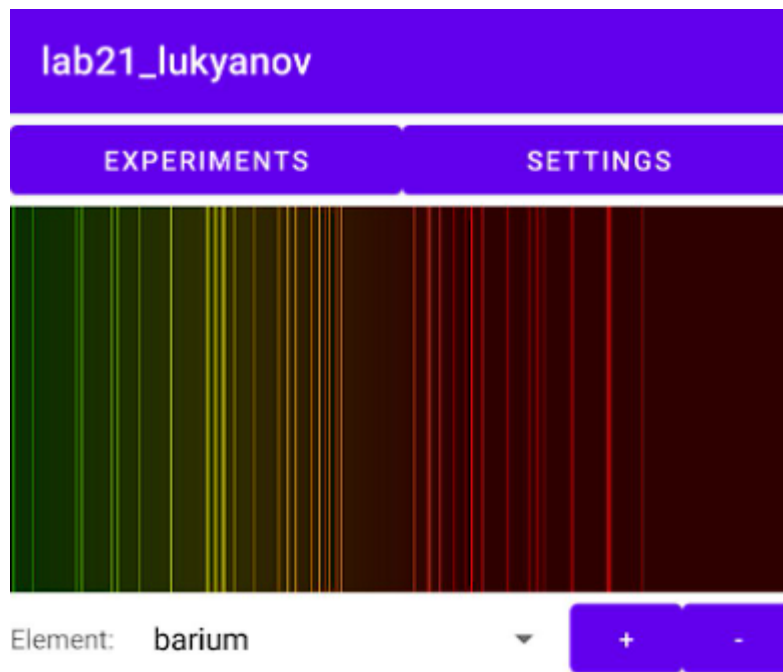


Рисунок 35 – Перемещение по спектру

Имеется возможность изменения масштаба (рис. 36):

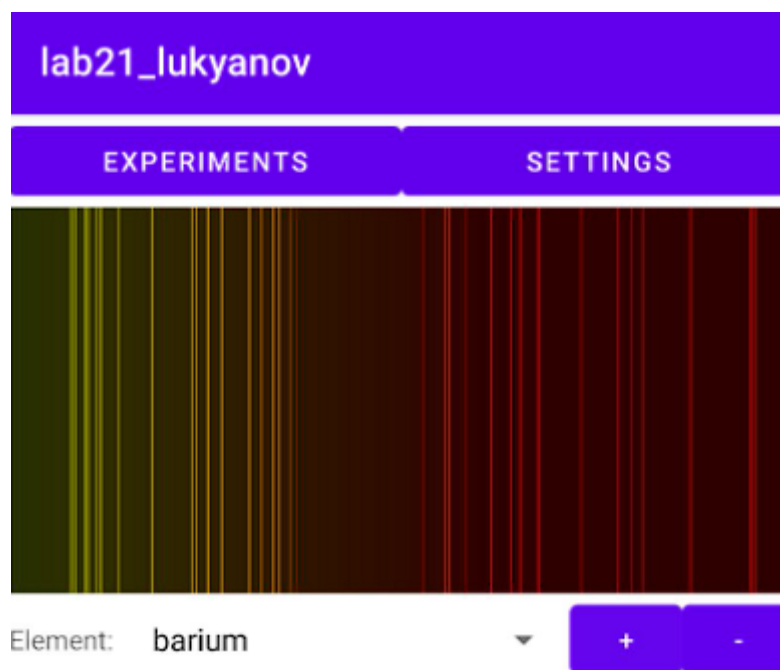


Рисунок 36 – Изменение масштаба

Имеется возможность выбора химического элемента из списка загруженных (рис. 37):

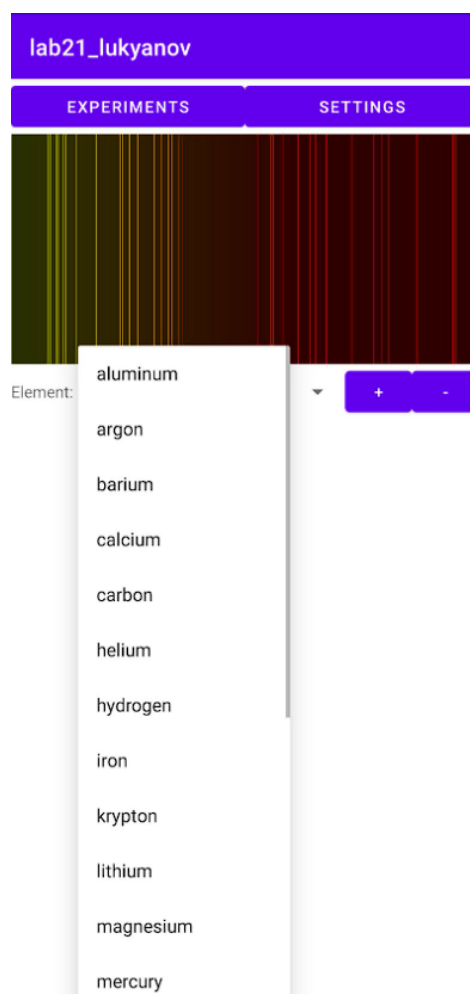


Рисунок 37 – Список химических элементов

Также имеется возможность задания настроек, а именно отображение делений на спектре и регулирование интенсивности фонового градиента (рис. 38):

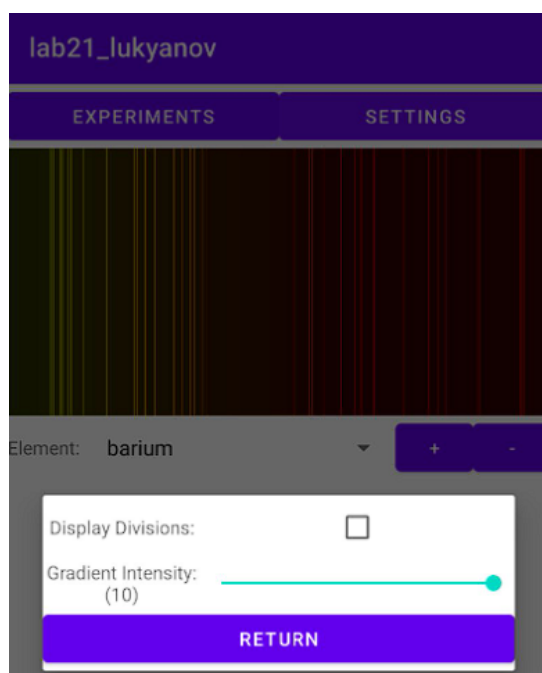


Рисунок 38 – Настройки отображения

Изменённые настройки отображения, они применяются также до выхода из диалогового окна (рис. 39):

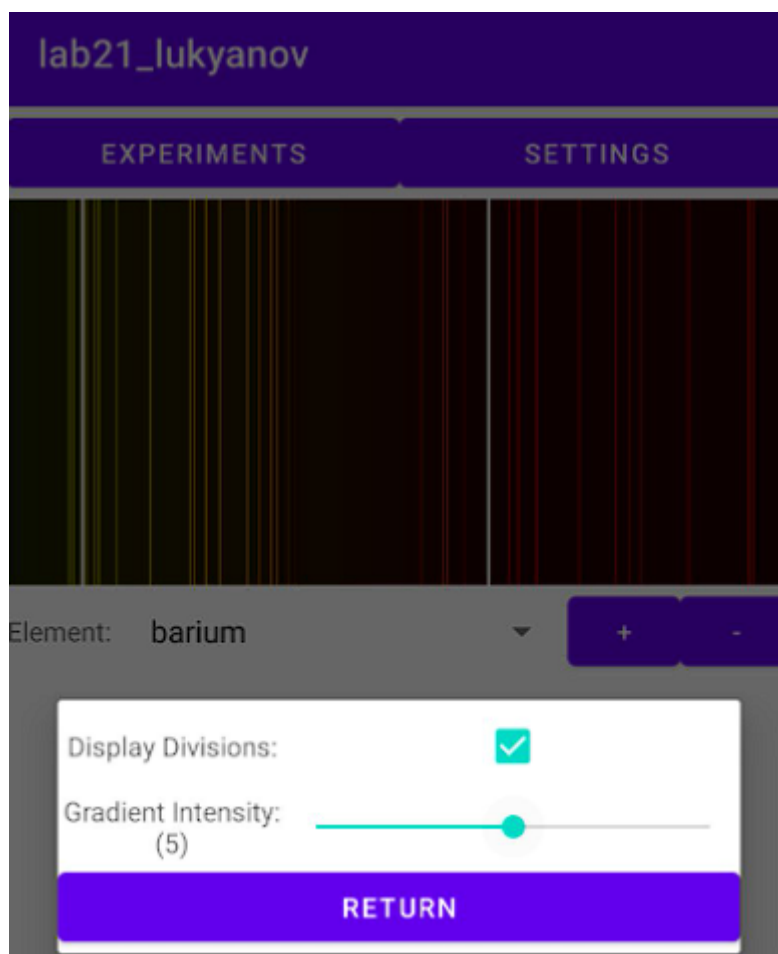


Рисунок 39 – Изменённые настройки

Приложение имеет собственную иконку (рис. 40):

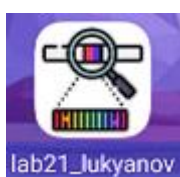


Рисунок 40 – Иконка приложения