

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»


**СОГЛАСОВАНО**

Научный руководитель, приглашённый  
преподаватель, доцент департамента  
программной инженерии, канд. техн. наук

  
\_\_\_\_\_ А.Д. Брейман  
« 10 » \_\_\_\_\_ мая \_\_\_\_\_ 2023 г.

**УТВЕРЖДЕНО**

Академический руководитель  
образовательной программы  
«Программная инженерия», профессор  
департамента программной инженерии,  
канд. техн. наук

  
\_\_\_\_\_ В.В. Шилов.  
« 10 » \_\_\_\_\_ мая \_\_\_\_\_ 2023 г.

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ  
В УДОБНОЙ ДЛЯ ПОЛЬЗОВАТЕЛЯ ФОРМЕ НА ПРИМЕРЕ POSTGRESQL**


**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.02.07-01 12 01-1-ЛУ**

Исполнитель

студент группы БПИ213

  
\_\_\_\_\_ А.С. Абрамов  
« 8 » \_\_\_\_\_ мая \_\_\_\_\_ 2023 г.

Подп. и дата	
Инв. № дубл	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

УТВЕРЖДЕН

RU.17701729.02.07-01 12 01-1-ЛУ

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ  
В УДОБНОЙ ДЛЯ ПОЛЬЗОВАТЕЛЯ ФОРМЕ НА ПРИМЕРЕ POSTGRESQL**

**Текст программы**

**RU.17701729.02.07-01 12 01-1**

**Листов: 158**

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

## АННОТАЦИЯ

Текст программы - документ, содержащий исходный код программы «Веб-приложение для управления реляционной базой данных в удобной для пользователя форме на примере PostgreSQL», разработанной с использованием языков JavaScript, HTML, CSS, SQL и шаблонизатора EJS.

Приложение позволяет пользователям производить простейшие операции с реляционной базой данных (в частности, PostgreSQL) без особенных знаний её устройства и языка запросов, а для работы требует лишь единоразовую настройку. Основной целью разработки программы является упрощение поддержки программных продуктов - избавление от необходимости постоянного содержания команды специалистов в области баз данных для эффективного решения возникающих проблем и помощи пользователям.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77<sup>[1]</sup>: Виды программ и программных документов.
- 2) ГОСТ 19.102-77<sup>[2]</sup>: Стадии разработки.
- 3) ГОСТ 19.103-77<sup>[3]</sup>: Обозначения программ и программных документов.
- 4) ГОСТ 19.104-78<sup>[4]</sup>: Основные надписи.
- 5) ГОСТ 19.105-78<sup>[5]</sup>: Общие требования к программным документам.
- 6) ГОСТ 19.106-78<sup>[6]</sup>: Требования к программным документам, выполненным печатным способом.
- 7) ГОСТ 19.401-78<sup>[7]</sup>: Текст программы. Требования к содержанию и оформлению.
- 8) ГОСТ 19.602-78<sup>[8]</sup>: Правила дублирования, учета и хранения программных документов, выполненных печатным способом.

Изменения к настоящему документу должны быть оформлены согласно ГОСТ 19.603-78<sup>[9]</sup> и ГОСТ 19.604-78<sup>[10]</sup>.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## СОДЕРЖАНИЕ

<b>1. PACKAGE.JSON.....</b>	<b>6</b>
<b>2. ПОДПРОГРАММА COMMON.....</b>	<b>8</b>
2.1. Директория configs.....	8
2.1.1. Шаблон файла config.json.....	8
2.1.2. Файл config.js.....	10
2.2. Директория postgresSQL.....	10
2.2.1. Файл postgresSQL.js.....	10
2.3. Директория redis.....	14
2.3.1. Файл redis.js.....	14
2.4. Директория utils.....	15
2.4.1. Файл Utils.js.....	15
2.4.2. Директория implementations.....	16
2.4.2.1. Директория Interval.....	16
2.4.2.1.1. Файл Interval.js.....	16
2.4.2.2. Директория object_manipulation_by_path.....	19
2.4.2.2.1. Файл get_field_at_path.js.....	19
2.4.2.2.2. Файл set_field_at_path.js.....	19
<b>3. ПОДПРОГРАММА POSTGRES SQL.....</b>	<b>21</b>
3.1. Файл create.js.....	21
3.2. Файл setup.sql.....	21
3.3. Файл setup.js.....	22
3.4. Файл fill.js.....	23
3.5. Файл delete.js.....	23
<b>4. ПОДПРОГРАММА WEBSITE.....</b>	<b>25</b>
4.1. Директория plugins.....	25
4.1.1. Файл security.js.....	25

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.1.2. Файл request.js.....	29
4.1.3. Файл response.js.....	30
4.1.4. Файл utility.js.....	36
4.1.5. Файл oauth.js.....	38
4.1.6. Файл routes.js.....	43
4.1.7. Файл other.js.....	45
4.2. Директория routes.....	46
4.2.1. Директория utils.....	46
4.2.1.1. Директория database.....	46
4.2.1.1.1. Файл get_database_info.js.....	46
4.2.1.1.2. Файл get_table_layout.js.....	47
4.2.1.1.3. Файл get_table_name.js.....	48
4.2.1.2. Директория schemas.....	48
4.2.1.2.1. Файл types.json.....	48
4.2.1.2.2. Файл schemas.js.....	49
4.2.1.3. Файл compile_websocket_handler.js.....	49
4.2.2. Директория websocket_data_routes.....	51
4.2.2.1. Директория utils.....	51
4.2.2.1.1. Файл table_rows_query.js.....	51
4.2.2.2. Файл database_data.js.....	52
4.2.2.3. Файл table_data.js.....	54
4.2.2.4. Файл table_rows.js.....	56
4.2.2.5. Файл logs.js.....	59
4.2.3. Файл main.js.....	61
4.2.4. Файл database.js.....	62
4.2.5. Файл admin.js.....	69
4.3. Директория static.....	74

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.3.1. Директория css.....	74
4.3.1.1. Файл general.css.....	74
4.3.2. Директория js.....	104
4.3.2.1. Директория utils.....	104
4.3.2.1.1. Файл general.js.....	104
4.3.2.1.2. Файл chart.js.....	108
4.3.2.1.3. Файл table.js.....	112
4.3.2.2. Директория admin.....	125
4.3.2.2.1. Файл logs.js.....	125
4.3.2.3. Файл database.js.....	126
4.3.2.4. Файл table.js.....	129
4.4. Директория views.....	132
4.4.1. Директория general.....	132
4.4.1.1. Файл head.ejs.....	132
4.4.1.2. Файл header.ejs.....	133
4.4.1.3. Файл input.ejs.....	134
4.4.1.4. Файл nav.ejs.....	137
4.4.2. Файл error.ejs.....	139
4.4.3. Директория admin.....	140
4.4.3.1. Файл accounts.ejs.....	140
4.4.3.2. Файл logs.ejs.....	145
4.4.4. Файл database.ejs.....	147
4.4.5. Файл oauth.ejs.....	149
4.4.6. Файл table.ejs.....	150
4.5. Файл index.js.....	154
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	157
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	158

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1. PACKAGE.JSON

```

{
  "name": "dbviewer",
  "type": "module",
  "author": "TTPO100AJIEX",
  "license": "MIT",
  "version": "1.0.0",
  "description": "",
  "scripts": {
    "website": "node website/index.js",
    "db:delete": "node PostgreSQL/delete.js",
    "db:create": "node PostgreSQL/create.js",
    "db:setup": "node PostgreSQL/setup.js",
    "db:fill": "node PostgreSQL/fill.js",
    "db:recreate": "npm run db:delete && npm run db:create && npm run db:setup && npm run db:fill"
  },
  "dependencies": {
    "@fastify/circuit-breaker": "^3.2.0",
    "@fastify/compress": "^6.2.1",
    "@fastify/cookie": "^8.3.0",
    "@fastify/cors": "^8.2.1",
    "@fastify/formbody": "^7.4.0",
    "@fastify/helmet": "^10.1.0",
    "@fastify/rate-limit": "^7.6.0",
    "@fastify/static": "^6.10.1",
    "@fastify/under-pressure": "^8.2.0",
    "@fastify/view": "^7.4.1",

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

"@fastify/websocket": "^7.2.0",
"ajv": "^8.12.0",
"ajv-formats": "^2.1.1",
"bcrypt": "^5.1.0",
"common": "file:common",
"ejs": "^3.1.9",
"fastify": "^4.15.0",
"fastify-plugin": "^4.5.0",
"html-minifier": "^4.0.0",
"ioredis": "^5.3.2",
"pg": "^8.10.0",
"pg-cursor": "^2.9.0",
"pg-format": "^1.0.4",
"postgres-interval": "^4.0.0"
},
"repository": {
  "type": "git",
  "url": "git+https://github.com/TTPO100AJIEX/DBViewer.git"
}
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## 2. ПОДПРОГРАММА COMMON

### 2.1. Директория configs

#### 2.1.1. Шаблон файла config.json

```
{
  "stage": "testing",
  "application": "DBViewer",

  "postgreSQL":
  {
    "internal":
    {
      "host": "localhost",
      "port": 5432,
      "database": "dbviewer",
      "user": "postgres",
      "password": ""
    },
    "target":
    {
      "host": "localhost",
      "port": 5432,
      "database": "",
      "user": "postgres",
      "password": ""
    }
  },
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

"redis":

```
{
  "port": 6379,
  "host": "localhost",
  "database": 0,
  "password": ""
},
```

"website":

```
{
  "host": "localhost.xyz",

  "ssl_key": "common/configs/ssl/privkey.pem",
  "ssl_cert": "common/configs/ssl/fullchain.pem",

  "https_port": 443,

  "secret": "",
  "authentication_expiration": 900000,
  "authorization_expiration": 3600000,
  "user_data_expiration": 300000
},
```

"bcrypt":

```
{
  "saltRounds": 11
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
},
```

```
"default_account":
```

```
{
```

```
  "login": "",
```

```
  "password": ""
```

```
},
```

```
"graph_records": 100,
```

```
"page_size": 30,
```

```
"update_interval": 1000
```

```
}
```

### 2.1.2. Файл config.js

```
import config from "./config.json" assert { type: "json" };
```

```
export default config;
```

## 2.2. Директория postgresSQL

### 2.2.1. Файл postgresSQL.js

```
import config from "common/configs/config.js";
```

```
import Utils from "common/Utils/Utils.js";
```

```
import pg from 'pg';
```

```
import format from 'pg-format';
```

```
pg.types.setTypeParser(pg.types.builtins.INTERVAL, value => new Utils.Interval(value));
```

```
import Cursor from "pg-cursor";
```

```
class PostgreSQLCursor
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  #client; #cursor;

  constructor(client, query, params = [ ], parser = (data) => data)
  {
    this.#client = client;
    this.query = query;
    this.params = params;
    this.parser = parser;
    this.#cursor = client.query(new Cursor(query, params));
  }

  async get(maxRows)
  {
    const data = this.parser(await this.#cursor.read(maxRows));
    return (maxRows === 1) ? data[0] : data;
  }

  async end()
  {
    await this.#cursor.close();
    this.#client.release();
  }
};

```

```
export default class PostgreSQL
```

```

{
  static format = format;
  format = PostgreSQL.format;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

#Connection;

static #DefaultPostgresOptions = { parseInputDatesAsUTC: true, application_name:
config.application };

constructor(options) { this.#Connection = new pg.Pool({ ...PostgreSQL.#DefaultPostgresOptions,
...options }); }

end() { this.#Connection.end(); }

get_raw() { return this.#Connection; }

#parse_response(rows)
{
  for (const key in rows[0])
  {
    const path = key.split('_');
    if (path.length == 1) continue;
    for (let obj of rows)
    {
      const save = obj[key]; delete obj[key];
      Utils.set_field_at_path(obj, path, save);
    }
  }
  return(rows);
}

async query(query, params, { parse = false, one_response = false } = { })
{
  let data = await this.#Connection.query(query, params);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

if (parse) data.rows = this.#parse_response(data.rows);
if (one_response) data.rows = data.rows[0];
return(data.rows);
}
async query_multiple(queries = { })
{
  let final_query = "", plan = [ ];
  for (const key in queries)
  {
    const info = (typeof queries[key] !== "object") ? { query: queries[key] } : queries[key];
    plan.push({ name: key, parse: info.parse ?? false, one_response: info.one_response ?? false });
    final_query += `${info.query}${info.query.endsWith(";") ? "" : ";"}`;
  }
  if (plan.length === 0) return { };

  let data = await this.#Connection.query(final_query);
  if (!Array.isArray(data)) data = [ data ];
  let result = Array.isArray(queries) ? [ ] : { };
  for (let i = 0; i < data.length; i++)
  {
    data[i] = data[i].rows;
    if (plan[i].parse) this.#parse_response(data[i]);
    if (plan[i].one_response) data[i] = data[i][0];
    result[plan[i].name] = data[i];
  }
  return(result);
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

async cursor(query, params, { parse = false } = { })
{
    return new PostgreSQLCursor(await this.#Connection.connect(), query, params, parse ?
this.#parse_response.bind(this) : undefined);
}
};

```

```

const TargetDatabase = new PostgreSQL(config.postgreSQL.target);
const InternalDatabase = new PostgreSQL(config.postgreSQL.internal);
export { TargetDatabase, InternalDatabase };

```

## 2.3. Директория redis

### 2.3.1. Файл redis.js

```

import config from "common/configs/config.js";

import ioredis from 'ioredis';

class Redis
{
    #Connection;

    static #DefaultRedisOptions = {
        connectionName: config.application,
        enableAutoPipeliningP: true
    };

    constructor(options) { this.#Connection = new ioredis({ ...Redis.#DefaultRedisOptions, ...options }); }

    end() { this.#Connection.quit(); }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

get_raw() { return this.#Connection; }

set(key, value) { return this.#Connection.set(key, value); }

set_expire(key, value, expiration) { return this.#Connection.setex(key, expiration / 1000, value); }

set_keepttl(key, value) { return this.#Connection.set(key, value, "KEEPTTL"); }

get(key) { return this.#Connection.get(key); }

get_delete(key) { return this.#Connection.getdel(key); }

async get_expire(key, expiration) { return (await this.#Connection.multi().get(key).expire(key,
expiration / 1000).exec())[0][1]; }

delete(...keys) { return (keys.length == 0) ? [ ] : this.#Connection.del(...keys); }

};

const Cache = new Redis(config.redis);

export { Cache };

```

## 2.4. Директория utils

### 2.4.1. Файл Utils.js

```

import get_field_at_path from "./implementations/object_manipulation_by_path/get_field_at_path.js";
import set_field_at_path from "./implementations/object_manipulation_by_path/set_field_at_path.js";

import Interval from "./implementations/Interval/Interval.js";

class Utils
{
    constructor() { console.error("Utils has been instantiated!"); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
static get_field_at_path = get_field_at_path;
```

```
static set_field_at_path = set_field_at_path;
```

```
static Interval = Interval;
```

```
};
```

```
export default Utils;
```

## 2.4.2. Директория implementations

### 2.4.2.1. Директория Interval

#### 2.4.2.1.1. Файл Interval.js

```
import PostgresInterval from "postgres-interval";
```

```
export default class Interval extends PostgresInterval
```

```
{
```

```
  static #format =
```

```
  /^(([0123456789]+y)?([0123456789]+w)?([0123456789]+d)?([0123456789]+h)?([0123456
```

```
789]+m)?([0123456789]+s)?$/;
```

```
  constructor(input)
```

```
  {
```

```
    switch (typeof input)
```

```
    {
```

```
      case "number":
```

```
      {
```

```
        super();
```

```
        this.seconds = input;
```

```
        break;
```

```
      }
```

```
    }
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

case "object":
{
    super();
    if (input.years) this.years = input.years;
    if (input.months) this.months = input.months;
    if (input.days) this.days = input.days;
    if (input.hours) this.hours = input.hours;
    if (input.minutes) this.minutes = input.minutes;
    if (input.seconds) this.seconds = input.seconds;
    if (input.milliseconds) this.milliseconds = input.milliseconds;
    break;
}
case "string":
{
    if (Interval.#format.test(input))
    {
        super();
        while (input.length != 0)
        {
            const index = input.search(/y|w|d|h|m|s/), number = Number(input.slice(0,
index)), symbol = input[index]; input = input.slice(index + 1);
            switch(symbol)
            {
                case "y": { this.years += number; break; }
                case "w": { this.days += number * 7; break; }
                case "d": { this.days += number; break; }
                case "h": { this.hours += number; break; }
            }
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        case "m": { this.minutes += number; break; }
        case "s": { this.seconds += number; break; }
        default: { }
    }
}
break;
}
}
default: { super(input); }
}
this.normalize();
}

normalize()
{
    if (this.milliseconds >= 1000) { this.seconds += Math.floor(this.milliseconds / 1000);
this.milliseconds %= 1000; }

    if (this.seconds >= 60) { this.minutes += Math.floor(this.seconds / 60); this.seconds
%= 60; }

    if (this.minutes >= 60) { this.hours += Math.floor(this.minutes / 60); this.minutes %=
60; }

    if (this.hours >= 24) { this.days += Math.floor(this.hours / 24); this.hours %= 24; }
    this.days += this.months * 30; this.months = 0;

    if (this.days >= 365) { this.years += Math.floor(this.days / 365); this.days %= 365; }
    if (this.days >= 30) { this.months += Math.floor(this.days / 30); this.days %= 30; }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

toSeconds() { return (((this.years * 365 + this.months * 30 + this.days) * 24 + this.hours)
* 60 + this.minutes) * 60 + this.seconds; }

toFormat()
{
    let answer = "";
    if (this.years !== 0) answer = `${this.years}y`;
    if (this.months + this.days !== 0) answer += `${this.months + this.days}d`;
    if (this.hours !== 0) answer += `${this.hours}h`;
    if (this.minutes !== 0) answer += `${this.minutes}m`;
    if (this.seconds !== 0) answer += `${this.seconds}s`;
    return answer;
}
};

```

#### 2.4.2.2. Директория object\_manipulation\_by\_path

##### 2.4.2.2.1. Файл get\_field\_at\_path.js

```

export default function get_field_at_path(obj, path)
{
    if (!path) return null;
    const realpath = path[0] === "." ? path.slice(1) : path;
    for (let i = 0; i < realpath.length - 1; i++) obj = (obj[realpath[i]] ?? {});
    return obj[realpath.at(-1)];
}

```

##### 2.4.2.2.2. Файл set\_field\_at\_path.js

```

export default function set_field_at_path(obj, path, value)
{
    for (let i = 0; i < path.length - 1; i++) obj = (obj[path[i]] ?? {});
    obj[path.at(-1)] = value;
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
return obj;  
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3. ПОДПРОГРАММА POSTGRESQL

#### 3.1. Файл create.js

```
import config from "common/configs/config.js";
```

```
import pg from 'pg';
```

```
const PostgreSQL = new pg.Pool({
  ...config.postgreSQL.internal,
  database: "postgres",
  parseInputDatesAsUTC: true,
  application_name: config.application
});
```

```
await PostgreSQL.query(`CREATE DATABASE ${config.postgreSQL.internal.database} TEMPLATE
template0 ENCODING UTF8`);
```

```
console.info(`Created database ${config.postgreSQL.internal.database}`);
```

```
PostgreSQL.end();
```

#### 3.2. Файл setup.sql

```
CREATE TABLE users
```

```
(
  id SMALLINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
  login VARCHAR(100) NOT NULL UNIQUE,
  password CHAR(60) NOT NULL,
  read BOOL NOT NULL DEFAULT FALSE,
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

insert BOOL NOT NULL DEFAULT FALSE,
update BOOL NOT NULL DEFAULT FALSE,
delete BOOL NOT NULL DEFAULT FALSE,
admin BOOL NOT NULL DEFAULT FALSE,
permissions CHAR(5) GENERATED ALWAYS AS (
    (CASE WHEN (read OR insert OR update OR delete OR admin) THEN 'R' ELSE '-' END) ||
    (CASE WHEN (insert OR admin) THEN 'I' ELSE '-' END) ||
    (CASE WHEN (update OR admin) THEN 'U' ELSE '-' END) ||
    (CASE WHEN (delete OR admin) THEN 'D' ELSE '-' END) ||
    (CASE WHEN admin THEN 'A' ELSE '-' END)
) STORED
);

CREATE TYPE REQUEST_TYPE AS ENUM ('SELECT', 'INSERT', 'UPDATE', 'DELETE');
CREATE TABLE logs
(
    id SMALLINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    date TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    type REQUEST_TYPE NOT NULL,
    data TEXT NOT NULL,
    userid SMALLINT REFERENCES users(id) MATCH FULL ON DELETE SET NULL ON UPDATE
    CASCADE,
    query TEXT NOT NULL,
    query_params TEXT NOT NULL DEFAULT '[' ]'
);

```

### 3.3. Файл setup.js

```
import { InternalDatabase } from "common/postgreSQL/postgreSQL.js";
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
import fs from 'fs';
await InternalDatabase.query(fs.readFileSync("PostgreSQL/setup.sql", "utf-8"));
console.log(`Created tables`);
```

```
InternalDatabase.end();
```

### 3.4. Файл fill.js

```
import config from "common/configs/config.js";
import { InternalDatabase } from "common/postgreSQL/postgreSQL.js";
import bcrypt from 'bcrypt';

const usersInsertQuery = `INSERT INTO users (login, password, read, insert, update, delete, admin)
VALUES ($1, $2, 'f', 'f', 'f', 'f', 't')`;

await InternalDatabase.query(usersInsertQuery, [ config.default_account.login, await
bcrypt.hash(config.default_account.password, config.bcrypt.saltRounds) ]);

console.log(`Filled tables`);
```

```
InternalDatabase.end();
```

### 3.5. Файл delete.js

```
import config from "common/configs/config.js";
```

```
import pg from 'pg';
```

```
const PostgreSQL = new pg.Pool({
  ...config.postgreSQL.internal,
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
database: "postgres",  
parseInputDatesAsUTC: true,  
application_name: config.application  
});
```

```
await PostgreSQL.query(`DROP DATABASE ${config.postgreSQL.internal.database} WITH (FORCE)`);  
console.info(`Deleted database ${config.postgreSQL.internal.database}`);
```

```
PostgreSQL.end();
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

#### 4. ПОДПРОГРАММА WEBSITE

##### 4.1. Директория plugins

###### 4.1.1. Файл security.js

```
import fastify_plugin from 'fastify-plugin';

import config from 'common/configs/config.js';
import { Cache } from "common/redis/redis.js";

import fastify_helmet from "@fastify/helmet";
import fastify_cors from "@fastify/cors";
import fastify_rate_limit from "@fastify/rate-limit";

function register_security_plugins(app, options, done)
{
  /*-----HELMET-----*/
  app.register(fastify_helmet,
  {
    global: true,
    enableCSPNonces: true,
    contentSecurityPolicy:
    {
      useDefaults: false,
      directives:
      {
        "default-src": [ "none" ],
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

// "child-src": [ "self" ],
"connect-src": [ "self", "report-sample" ],
"font-src": [ "self", "https://fonts.gstatic.com", "report-sample" ],
// "frame-src": [ "self" ],
"img-src": [ "self", "report-sample" ],
// "media-src": [ "self" ],
// "object-src": [ "self" ],
// "prefetch-src": [ "self" ],
"script-src": [ "strict-dynamic", "https://www.gstatic.com", "self", "report-sample" ],
"style-src": [ "self", "https://fonts.googleapis.com", "https://www.gstatic.com",
"report-sample" ],
// "worker-src": [ "self" ],

"base-uri": [ "none" ],

"form-action": [ "self", "report-sample" ],
// "frame-ancestors": [ "self" ],

"report-uri": [ "/csp-violation-report" ],
"report-to": [ "dbviewer" ],
"require-trusted-types-for": [ "script" ],
"upgrade-insecure-requests": [ ]
}
},
crossOriginEmbedderPolicy: true, //require-corp
crossOriginOpenerPolicy: { policy: "same-origin" },
crossOriginResourcePolicy: { policy: "same-origin" },

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

expectCt: { maxAge: 24 * 60 * 60, enforce: true, reportUri: "/ct-violation-report" },
referrerPolicy: { policy: "strict-origin-when-cross-origin" },
hsts: { maxAge: 7 * 24 * 60 * 60, includeSubDomains: false, preload: true },
noSniff: true,
originAgentCluster: true,
dnsPrefetchControl: { allow: true },
frameguard: { action: "SAMEORIGIN" },
hidePoweredBy: true,
xssFilter: true
});

app.addContentTypeParser('application/csp-report', { parseAs: 'string' }, function (request, payload,
done)
{
    try { done(null, JSON.parse(payload)) } catch(err) { err.statusCode = 400; done(err, undefined) }
});

app.addContentTypeParser('application/reports+json', {parseAs: 'string'}, function (request, payload,
done)
{
    try { done(null, JSON.parse(payload)) } catch(err) { err.statusCode = 400; done(err, undefined) }
});

app.addHook("onRequest", (req, res, next) =>
{
    res.header('Report-To', `{"group":"dbviewer","max_age":10886400,"endpoints":[{"url":
"https://${config.website.host}/csp-violation-report"}]}`);
    res.header('Reporting-Endpoints', `dbviewer="/csp-violation-report"`);
    next();
});

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

app.post('/csp-violation-report', (req, res) =>
{
  console.warn(req.body);
  return res.send("Acknowledged");
});
app.post('/ct-violation-report', (req, res) =>
{
  console.warn(req.body);
  return res.send("Acknowledged");
});

```

```

/*-----CORS-----*/

```

```

app.register(fastify_cors,
{
  origin: [ `https://${config.website.host}` ],
  methods: 'GET,POST',
  maxAge: 300
});

```

```

/*-----RATE LIMIT-----*/

```

```

app.register(fastify_rate_limit, {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

global: true,
max: config.stage == "testing" ? 1000000 : 100,
timeWindow: 60000,
ban: config.stage == "testing" ? 1500000 : 150,
continueExceeding: true,
addHeadersOnExceeding: { 'x-ratelimit-limit': false, 'x-ratelimit-remaining': false,
'x-ratelimit-reset': false, 'retry-after': false },
addHeaders: { 'x-ratelimit-limit': false, 'x-ratelimit-remaining': true, 'x-ratelimit-reset': false,
'retry-after': true },
cache: 10000,
redis: Cache.get_raw(),
nameSpace: 'website-rate-limit-',
skipOnError: true,
onExceeded: (req) => console.info(`Rate Limit exceeded by ${req.ip}`)
});

done();
}

```

```
export default fastify_plugin(register_security_plugins, { name: 'security', encapsulate: false });
```

#### 4.1.2. Файл request.js

```

import fastify_plugin from 'fastify-plugin';

import fastify_formbody from "@fastify/formbody";
import fastify_websocket from "@fastify/websocket";

function register_request_plugins(app, options, done)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  /*-----FORMBODY-----*/

  app.register(fastify_formbody);

  /*-----WEBSOCKET-----*/

  app.register(fastify_websocket);

  done();
}

export default fastify_plugin(register_request_plugins, { name: 'request', encapsulate: false });

```

#### 4.1.3. Файл response.js

```

import fastify_plugin from 'fastify-plugin';

import config from 'common/configs/config.js';

import fastify_compress from "@fastify/compress";
import { dirname } from 'path';
import { fileURLToPath } from 'url';
import fastify_static from "@fastify/static";
import view from "@fastify/view";
import ejs from 'ejs';
import html_minifier from "html-minifier";

function register_response_plugins(app, options, done)
{
  /*-----COMPRESS-----*/

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
app.register(fastify_compress);
```

```
/*-----STATIC-----*/
```

```
app.register(fastify_static, {
```

```
  root: dirname(dirname(fileURLToPath(import.meta.url))) + "/static",
```

```
  prefix: "/static",
```

```
  acceptRanges: true,
```

```
  cacheControl: true,
```

```
  dotfiles: 'ignore',
```

```
  etag: true,
```

```
  extensions: false,
```

```
  immutable: false,
```

```
  lastModified: true,
```

```
  maxAge: config.stage === "testing" ? 0 : 604800000,
```

```
  preCompressed: true,
```

```
  wildcard: false,
```

```
  index: false,
```

```
  allowedPath: (pathname, root) =>
```

```
  {
```

```
    return(pathname.startsWith("/css") || pathname.startsWith("/js") ||
```

```
    pathname.startsWith("/images"))
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

    || pathname == "robots.txt" || pathname == "sitemap.xml" || pathname == "favicon.ico" ||
    pathname == "apple-touch-icon.png")
  }
});

```

```

/*-----EJS-----*/
app.register(view, {
  "engine": { "ejs": ejs },
  "production": config.stage != "testing",
  "options":
  {
    "useHtmlMinifier": config.stage != "testing" ? html_minifier : false,
    "htmlMinifierOptions":
    {
      "caseSensitive": true,
      "collapseBooleanAttributes": true,
      "collapseInlineTagWhitespace": true,
      "collapseWhitespace": true,
      "conservativeCollapse": false,
      "continueOnError": false,
      "decodeEntities": true,
      "html5": true,
      "includeAutoGeneratedTags": true,
      "keepClosingSlash": false,

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

"minifyCSS": true,
"minifyJS": true,
"minifyURLs": false,
"preserveLineBreaks": false,
"preventAttributesEscaping": false,
"processConditionalComments": false,
"processScripts": [],
"quoteCharacter": "\"",
"removeAttributeQuotes": false,
"removeComments": true,
"removeEmptyAttributes": true,
"removeEmptyElements": false,
"removeOptionalTags": false,
"removeRedundantAttributes": false,
"removeScriptTypeAttributes": true,
"removeStyleLinkTypeAttributes": true,
"removeTagWhitespace": false,
"sortAttributes": true,
"sortClassName": true,
"trimCustomFragments": false,
"useShortDoctype": true
}
},
"propertyName": "render",
"root": "./website/views/",
"defaultContext": { config }
});

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

app.addHook("onRequest", (req, res, next) =>
{
    req.response = res;
    res.locals = { "req": req, "res": res };
    next();
});

/*-----ERROR-----*/

app.decorateReply("error", function(code, description, opts)
{
    if (!description)
    {
        switch(code)
        {
            case 200: { description = [ "Действие было выполнено успешно!", "Вы будете
перенаправлены обратно в ближайшее время!" ]; break; }

            case 400: { description = [ "Запрос неверен!", "Повторите попытку!" ]; break; }
            case 401: { description = [ "Ошибка авторизации!", "Авторизуйтесь и повторите
попытку!" ]; break; }
            case 402: { description = [ "Запрошенный ресурс доступен только после оплаты!",
"Оплатите доступ к ресурсу и попробуйте снова!" ]; break; }
            case 403: { description = [ "У вас нет доступа к запрошенному ресурсу!", "Повторите
попытку позже или обратитесь в поддержку!" ]; break; }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    case 404: { description = [ "Запрошенный ресурс не найден!", "Убедитесь, что указан
верный адрес!" ]; break; }

    case 429: { description = [ "Вы отправляете слишком много запросов!", "Попробуйте
позже!" ]; break; }

    case 500: { description = [ "Ошибка сервера!", "Повторите попытку позже или обратитесь
в поддержку!" ]; break; }

    case 502: { description = [ "Ошибка взаимодействия!", "Повторите попытку позже или
обратитесь в поддержку!" ]; break; }

    case 503: { description = [ "Сервер перегружен!", "Повторите попытку позже или
обратитесь в поддержку!" ]; break; }

    case 504: { description = [ "Ошибка взаимодействия!", "Повторите попытку позже или
обратитесь в поддержку!" ]; break; }

    case 508: { description = [ "Ошибка обработки запроса!", "Повторите попытку позже или
обратитесь в поддержку!" ]; break; }

    default: { description = [ "Произошла неизвестная ошибка!", "Повторите попытку позже
или обратитесь в поддержку!" ]; }
  }
}

return this.status(isNaN(code) ? 500 : code).render("error.ejs", { "error": code, "title": "",
"description": description, ...opts });
});

app.setNotFoundHandler({ preHandler: app.rateLimit({ max: 25, timeWindow: 60000, ban: 40 }) },
(req, res) => res.error(404));

app.setErrorHandler((error, req, res) =>
{
  if (config.stage === "testing") console.error(error);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
res.error(error.statusCode ?? 500, [ "Ошибка сервера!", error.message ?? error ]);
});
```

```
done();
}
```

```
export default fastify_plugin(register_response_plugins, { name: 'response', encapsulate: false });
```

#### 4.1.4. Файл utility.js

```
import fastify_plugin from 'fastify-plugin';
```

```
import fastify_circuit_breaker from "@fastify/circuit-breaker";
```

```
import under_pressure from '@fastify/under-pressure';
```

```
function register_utility_plugins(app, options, done)
```

```
{
```

```
/*-----CIRCUIT BREAKER-----*/
```

```
app.register(fastify_circuit_breaker, {
```

```
  threshold: 5, timeout: 10000,
```

```
  onCircuitOpen: async (req, res) =>
```

```
{
```

```
  console.warn(`@fastify/circuit-breaker: onCircuitOpen triggered for
```

```
${req.routerPath}${req.query}${JSON.stringify(req.body)}`);
```

```
  return res.error(508);
```

```
},
```

```
onTimeout: async (req, res) =>
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  console.warn(`@fastify/circuit-breaker: onTimeout triggered for
${req.routerPath}${req.query}${JSON.stringify(req.body)}`);
  return res.error(504);
}
});

```

```

/*-----UNDER PRESSURE-----*/
app.register(under_pressure, {
  maxEventLoopDelay: 250,
  maxHeapUsedBytes: 1048576000,
  maxRssBytes: 1048576000,
  maxEventLoopUtilization: 0.8,
  message: 'The server is exhausted! Try again later!',
  retryAfter: 60,
  pressureHandler: (req, res, type, value) =>
  {
    if (type === under_pressure.TYPE_HEAP_USED_BYTES) { console.warn('Heap has been
exhausted: ${value}'); return; }

    if (type === under_pressure.TYPE_RSS_BYTES) { console.warn('RSS has been exhausted:
${value}'); return; }

    return res.error(503);
  },
  exposeStatusRoute:

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  routeResponseSchemaOpts:
    {
      metrics:
        {
          type: 'object',
          properties: { eventLoopDelay: { type: 'number' }, rssBytes: { type: 'number' }, heapUsed: {
type: 'number' }, eventLoopUtilized: { type: 'number' } }
        },
      uptime: { type: 'number' }
    },
  url: "/serverstatus"
},
  healthCheck: async (fastifyInstance) => ({ metrics: fastifyInstance.memoryUsage(), uptime:
process.uptime() }),
});

done();
}

```

```
export default fastify_plugin(register_utility_plugins, { name: 'utility', encapsulate: false });
```

#### 4.1.5. Файл `oauth.js`

```
import fastify_plugin from 'fastify-plugin';
```

```
import config from 'common/configs/config.js';
```

```
import { InternalDatabase } from 'common/postgreSQL/postgreSQL.js';
```

```
import { Cache } from "common/redis/redis.js";
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

import bcrypt from 'bcrypt';
import crypto from 'crypto';

function register_oauth(app, options, done)
{
  app.decorateRequest("session_id", null);
  app.decorateRequest("get_sessionid", function()
  {
    if (this.session_id) return this.session_id;
    if ("__Secure-authorization" in this.cookies)
    {
      let session_id = { valid: false };
      try { session_id = this.unsignCookie(this.cookies["__Secure-authorization"]); } catch(err) { }
      if (session_id.valid && session_id.value.length == 44) this.session_id = session_id.value;
    }
    return this.session_id;
  });

  app.decorateRequest("authentication", null);
  app.decorateRequest("authentication_data", null);
  app.decorateRequest("authenticate", async function()
  {
    if (this.authentication) return this.authentication;
    this.authentication = crypto.randomBytes(16).toString('hex');
    await Cache.set_expire(`authentication-${this.authentication}`, JSON.stringify({ "ip": this.ip,
    "page": this.url } ), config.website.authentication_expiration);
  });

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

    return this.authentication;
  });

  app.decorateRequest("check_authentication", async function(code)
  {
    if (this.authentication_data) return this.authentication_data;
    let authentication = await Cache.get_delete(`authentication-${code}`);
    if (!authentication) return this.authentication_data = null;
    authentication = JSON.parse(authentication);
    if (this.ip !== authentication.ip) return this.authentication_data = null;
    return this.authentication_data = authentication;
  });

  app.decorateRequest("authorization", null);
  app.decorateRequest("authorize", async function()
  {
    if (this.authorization) return this.authorization;
    if (!this.get_sessionid()) return this.authorization = null;
    let authorization = await Cache.get_expire(`session_id-${this.session_id}`,
    config.website.authorization_expiration); // { "user_id": 1 }
    if (!authorization) return this.authorization = null;
    authorization = JSON.parse(authorization);

    if (!authorization.expiration || new Date(authorization.expiration) <= new Date())
    {
      authorization = await InternalDatabase.query(`SELECT id AS user_id, login, permissions
      FROM users WHERE id = $1`, [ authorization.user_id ], { one_response: true });
    }
  });

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    if (!authorization) { await Cache.delete(`session_id-${this.session_id}`); return this.authorization
= null; }

    authorization = { ...authorization, expiration: Date.now() + config.website.user_data_expiration
};

    await Cache.set_keepttl(`session_id-${this.session_id}`, JSON.stringify(authorization));

}

return this.authorization = authorization;

});

app.addHook('preValidation', async (req, res) =>
{
    if (!req.routerPath || !req.routeConfig?.access || req.routeConfig.access === "only_public") return;
    await req.authenticate();
});

app.addHook('preHandler', async (req, res) =>
{
    if (!req.routeConfig?.access || req.routeConfig.access === "only_public") return;
    if (req.method !== "GET" && !(await req.check_authentication(req.body.authentication))) return
res.error(400);
    await req.authorize();
    if (!req.authorization && req.routeConfig.access === "authorization") return res.error(401);
});

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

app.decorateReply("login", async function(login, password)
{
  const req = this.request, res = this;

  if (!(await req.check_authentication(req.body.authentication))) return res.error(400);

  const users = await InternalDatabase.query(` SELECT id, login, password, permissions FROM
users WHERE login = $1`, [ login ]);

  if (users.length === 0) return res.error(401, [ "Ошибка авторизации!", "Такого пользователя нет!"
]);

  for (const user of users)
  {
    if (!(await bcrypt.compare(password, user.password))) continue;

    req.session_id = crypto.randomBytes(32).toString('base64');

    const user_data = { user_id: user.id, login: user.login, permissions: user.permissions, expiration:
Date.now() + config.website.user_data_expiration };

    await Cache.set_expire(`session_id-${req.session_id}`, JSON.stringify(user_data),
config.website.authorization_expiration);

    res.setCookie('__Secure-authorization', req.session_id, { domain: config.website.host, path: '/',
secure: true, httpOnly: true, sameSite: 'Lax', signed: true });

    return res.redirect("/database");
  }

  return res.error(401, [ "Ошибка авторизации!", "Пароль не верный!" ]);
});

app.decorateReply("logout", async function()
{

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    this.clearCookie('__Secure-authorization', { domain: config.website.host, path: '/', secure: true,
httpOnly: true, sameSite: 'Lax', signed: true });

    return this.status(301).redirect("/");

});

```

```

done();
}

```

```

export default fastify_plugin(register_oauth, {
  name: 'oauth', encapsulate: false,
  decorators:
  {
    reply: [ "error" ]
  },
  dependencies: [ 'security', 'utility', 'other', 'request', 'response' ]
});

```

#### 4.1.6. Файл routes.js

```

import fastify_plugin from 'fastify-plugin';

import fs from 'fs';
import path, { dirname } from 'path';
import { fileURLToPath } from 'url';
async function register_routes(app, options)
{

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

app.get('/favicon.ico', (req, res) => res.sendFile("favicon.ico",
dirname(dirname(fileURLToPath(import.meta.url))) + '/static/images/favicon'));

app.get('/apple-touch-icon.png', (req, res) => res.sendFile("apple-touch-icon.png",
dirname(dirname(fileURLToPath(import.meta.url))) + '/static/images/favicon'));

app.get('/robots.txt', (req, res) => res.sendFile("robots.txt",
dirname(dirname(fileURLToPath(import.meta.url)))));

app.get('/sitemap.xml', (req, res) => res.sendFile("sitemap.xml",
dirname(dirname(fileURLToPath(import.meta.url)))));

```

```

for (const filename of fs.readdirSync("website/routes"))
{
  if (!fs.lstatSync(path.join("website/routes", filename)).isFile()) continue;
  const { default: routes } = await import(path.join("../routes", filename));
  for (const route of routes)
  {
    route.config ??= { };
    route.config.access ??= route.access;
    delete route.access;

    if ("paths" in route)
    {
      for (const path of route.paths)
      {
        const { paths: _, ...routeCopy } = route;
        routeCopy.path = path;
        app.route(routeCopy);
      }
    }
  }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
    else
    {
        app.route(route);
    }
}
}
}
}

```

```

export default fastify_plugin(register_routes, {
  name: 'routes', encapsulate: false,
  decorators: { reply: [ "sendFile", "render", "error", "login", "logout" ] },
  dependencies: [ 'security', 'utility', 'other', 'request', 'response', 'oauth' ]
});

```

#### 4.1.7. Файл other.js

```
import fastify_plugin from 'fastify-plugin';
```

```
import config from 'common/configs/config.js';
```

```
import fastify_cookie from "@fastify/cookie";
```

```
function register_other_plugins(app, options, done)
```

```
{
```

```
  /*-----COOKIE-----*/
```

```
  app.register(fastify_cookie, { secret: config.website.secret });
```

```
  done();
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
}
```

```
export default fastify_plugin(register_other_plugins, { name: 'other', encapsulate: false });
```

## 4.2. Директория routes

### 4.2.1. Директория utils

#### 4.2.1.1. Директория database

##### 4.2.1.1.1. Файл get\_database\_info.js

```
export default async function get_database_info(DatabaseConnection)
{
  const [ { database_name }, tables ] = await DatabaseConnection.query_multiple([
    {
      query: `SELECT current_database() AS database_name`,
      one_response: true
    },
    {
      query: `SELECT
        pg_class.oid AS id,
        (CASE WHEN pg_namespace.nspname = 'public' THEN " ELSE
pg_namespace.nspname || '.' END) || pg_class.relname AS name
        FROM pg_class INNER JOIN pg_namespace ON pg_namespace.oid =
pg_class.relnamespace
        WHERE pg_class.relkind = 'r' AND pg_namespace.nspname !=
'information_schema' AND NOT starts_with(pg_namespace.nspname, 'pg_')
      `
    }
  ]);
  return { database_name, tables };
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**4.2.1.1.2. Файл get\_table\_layout.js**

```

export default function get_table_layout(DatabaseConnection, type, input)
{
    const queryString = `
        WITH enums (enumtypeid, enumlabels) AS ( SELECT enumtypeid,
array_agg(enumlabel::text) AS enumlabels FROM pg_enum GROUP BY enumtypeid )
        SELECT
            pg_attribute.attname AS name,
            pg_attribute.atttypeid AS type_id,
            pg_attribute.atttypmod AS type_mod,
            format_type(pg_attribute.atttypeid, pg_attribute.atttypmod) AS type_name,
            pg_attribute.attidentity = 'a' AS flags_IA,
            pg_attribute.attidentity = 'd' AS flags_ID,
            pg_attribute.attnotnull AS flags_NN,
            pg_attribute.atthasdef AS flags_D,
            (pg_attribute.attgenerated = 's') AS flags_G,
            EXISTS( SELECT * FROM pg_constraint WHERE pg_constraint.conrelid =
pg_class.oid AND pg_attribute.attnum = ANY(pg_constraint.conkey) AND
pg_constraint.contype = 'u' ) AS flags_U,
            EXISTS( SELECT * FROM pg_constraint WHERE pg_constraint.conrelid =
pg_class.oid AND pg_attribute.attnum = ANY(pg_constraint.conkey) AND
pg_constraint.contype = 'p' ) AS flags_PK,
            enums.enumlabels AS enumlabels
        FROM pg_attribute INNER JOIN pg_class ON pg_class.oid = pg_attribute.attrelid
        LEFT JOIN enums ON enums.enumtypeid = pg_attribute.atttypeid
        WHERE pg_class.${type} == "name" ? "relname" : "oid" = $1 AND
pg_attribute.attnum > 0 AND NOT pg_attribute.attisdropped
        ORDER BY pg_attribute.attnum ASC`;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

    return DatabaseConnection.query(queryString, [ input ], { parse: true });
}

```

#### 4.2.1.1.3. Файл `get_table_name.js`

```

export default function get_table_name(DatabaseConnection, tableid)
{
    const queryString = `SELECT pg_namespace.nspname AS schema, pg_class.relname
    AS table

        FROM pg_class INNER JOIN pg_namespace ON pg_namespace.oid =
pg_class.relnamespace

        WHERE pg_class.oid = $1`;

    return DatabaseConnection.query(queryString, [ tableid ], { one_response: true });
}

```

#### 4.2.1.2. Директория `schemas`

##### 4.2.1.2.1. Файл `types.json`

```

{
    "authentication": { "type": "string", "minLength": 32, "maxLength": 32 },

    "url": { "type": "string", "format": "uri" },

    "color": { "type": "integer", "minimum": 0, "maximum": 16777215 },

    "interval": { "type": "string", "maxLength": 15, "pattern":
"^[([0123456789]+y)?([0123456789]+w)?([0123456789]+d)?([0123456789]+h)?([0123456
789]+m)?([0123456789]+s)?$" },

    "font": { "type": "string", "maxLength": 64 },

    "integer": { "type": "integer" },

    "smallint": { "type": "integer", "minimum": -32767, "maximum": 32767 },

    "smallint_positive": { "type": "integer", "minimum": 1, "maximum": 32767 },

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

"uinteger": { "type": "integer", "minimum": 0 },
"uinteger_wth_invalid": { "oneOf": [ { "type": "integer", "minimum": 0 }, { "type":
"string", "enum": [ "" ] } ] },

"date": { "type": "string", "format": "date-time" }
}

```

#### 4.2.1.2.2. Файл schemas.js

```

import types from './types.json' assert { type: "json" };
export { types };

```

```

const EMPTY_GET_SCHEMA = { query: { type: "object", additionalProperties: false,
properties: { } } };
export { EMPTY_GET_SCHEMA };

```

#### 4.2.1.3. Файл compile\_websocket\_handler.js

```

import fs from 'fs';
import path from 'path';

import Ajv from "ajv/dist/jtd.js";
const ajv = new Ajv();

export default async function compile_websocket_handler(folder)
{
  let handlers = { }, schemas = { };
  for (const filename of fs.readdirSync(path.join("website/routes", folder)))
  {
    if (!fs.lstatSync(path.join("website/routes", folder, filename)).isFile()) continue;
    const { default: routes } = await import(path.join("../", folder, filename));
  }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

for (const route of routes)
{
    if (!("handler" in route)) throw `load_websocket_routes: ${filename} does not export
handler`;

    if (!("requestNames" in route) && !("requestName" in route)) throw
`load_websocket_routes: ${filename} does not export both requestName and requestNames`;

    if ("requestNames" in route)
    {
        for (const requestName of route.requestNames)
        {
            if (requestName in handlers) throw `load_websocket_routes: route ${requestName}
already registered`;

            handlers[requestName] = route.handler;

            if (route.schema) schemas[requestName] = { properties: { data: route.schema } };
            else schemas[requestName] = { optionalProperties: { data: { } } };
        }
    }
    else
    {
        if (route.requestName in handlers) throw `load_websocket_routes: route
${requestName} already registered`;

        handlers[route.requestName] = route.handler;

        if (route.schema) schemas[route.requestName] = { properties: { data: route.schema } };
        else schemas[route.requestName] = { optionalProperties: { data: { } } };
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }

    const parseRequest = ajv.compileParser({ discriminator: "requestName", mapping: schemas });
    return async function handleRequest(connection, request, message)
    {
        message = parseRequest(message);
        if (message === undefined) throw 'Invalid object received: ' + parseRequest.message;
        await handlers[message.requestName](message.data, connection.socket, request);
    };
}

```

#### 4.2.2. Директория websocket\_data\_routes

##### 4.2.2.1. Директория utils

###### 4.2.2.1.1. Файл table\_rows\_query.js

```

function table_rows_query_conditions(filters)
{
    if (filters.length === 0) return { query: "", params: [ ] };
    return {
        query: `WHERE ${filters.map(filter => filter.comparison === "exact" ? `%I = %L` :
        `%I::text LIKE %L`).join(" AND ")}`,
        params: filters.map(filter => filter.comparison === "exact" ? [ filter.name, filter.value ] :
        [ filter.name, `%${filter.value}%` ]).flat()
    }
}

function table_rows_query_ordering(sorts)
{
    if (sorts.length === 0) return { query: "", params: [ ] };

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

return {
  query: `ORDER BY ${sorts.map(sorting => `%I ${sorting.order == "asc" ? "asc" :
"desc"}\`).join(', ')}\`,
  params: sorts.map(sorting => sorting.name)
}
}

```

```

export default function table_rows_query(columns, table, filters, sorts)
{
  const { query: conditionQuery, params: conditionsParams } =
table_rows_query_conditions(filters);
  const { query: orderingQuery, params: orderingParams } =
table_rows_query_ordering(sorts);
  return {
    query: `SELECT ${columns} FROM ${table} ${conditionQuery}
${orderingQuery}`,
    params: [ ...conditionsParams, ...orderingParams ]
  };
}

```

#### 4.2.2.2. Файл database\_data.js

```
import { TargetDatabase } from "common/postgreSQL/postgreSQL.js";
```

```

const generalDataQueryString = `SELECT
  current_database() AS database_name,
  version() AS postgres_version,
  NOW() - pg_postmaster_start_time() AS uptime,
  pg_size_pretty(pg_database_size(current_database())) AS database_size`;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
const stateDataQueryString = `SELECT state, COUNT(*) AS amount FROM pg_stat_activity
WHERE datname = current_database() GROUP BY state`;
```

```
const activityDataQueryString = `SELECT
    xact_commit AS queries_success, xact_rollback AS queries_cancelled,
    tup_returned AS reads_total, tup_fetched AS reads_index,
    tup_inserted AS updates_inserted, tup_updated AS updates_updated,
tup_deleted AS updates_deleted,
    blks_read + blks_hit AS blocks_total, blks_hit AS blocks_cache,
    active_time AS worktime_worktime,
    stats_reset AS reset
FROM pg_stat_database WHERE datname = current_database()`
```

```
async function database_data(msg, socket, req)
```

```
{
    const [ { database_name, postgres_version, uptime, database_size }, connections_stats, stats ] =
await TargetDatabase.query_multiple([
    { query: generalDataQueryString, one_response: true },
    { query: stateDataQueryString },
    { query: activityDataQueryString, parse: true, one_response: true }
]);
```

```
socket.send(JSON.stringify({
    eventName: "database_data",
    data:
    {
        database_name,
        postgres_version,
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    uptime: uptime.toPostgres(),
    database_size,
    connections:
    {
      active: connections_stats.find(data => data.state === "active")?.amount ?? 0,
      idle: connections_stats.find(data => data.state === "idle")?.amount ?? 0
    },
    ...stats
  }
}));
}

```

```

export default [
  { requestName: "database_data", handler: database_data }
];

```

#### 4.2.2.3. Файл table\_data.js

```

import { TargetDatabase } from "common/postgreSQL/postgreSQL.js";

const tableDataQuery = `SELECT
    (CASE WHEN pg_namespace.nspname = 'public' THEN " ELSE
pg_namespace.nspname || '.' END) || pg_class.relname AS table_name,
    (CASE
      WHEN pg_class.relkind = 'r' THEN 'table'
      ELSE 'unknown'
    END) AS table_type,
    pg_size_pretty(pg_total_relation_size(pg_class.oid)) AS table_size,
    pg_class.reltuples AS table_rows,

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

pg_stat_user_tables.seq_tup_read + pg_stat_user_tables.idx_tup_fetch AS
reads_total,

pg_stat_user_tables.idx_tup_fetch AS reads_index,

pg_stat_user_tables.seq_scan AS scans_sequential,
pg_stat_user_tables.idx_scan AS scans_index,

pg_stat_user_tables.n_tup_ins AS updates_inserted,
pg_stat_user_tables.n_tup_upd AS updates_updated,
pg_stat_user_tables.n_tup_del AS updates_deleted,

pg_stat_user_tables.n_live_tup AS live_live,
pg_stat_user_tables.n_dead_tup AS live_dead
FROM pg_class
INNER JOIN pg_namespace ON pg_namespace.oid = pg_class.relnamespace
LEFT JOIN pg_stat_user_tables ON pg_class.oid = pg_stat_user_tables.relid
WHERE pg_class.oid = $1`;

```

```

async function table_data(msg, socket)
{
    socket.send(JSON.stringify({ eventName: "table_data", data: await
TargetDatabase.query(tableDataQuery, [ msg.id ], { parse: true, one_response: true }) }));
}

```

```

const SCHEMA = { additionalProperties: false, properties: { id: { type: "int32" } } };
export default [

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
{ requestName: "table_data", schema: SCHEMA, handler: table_data }
];
```

#### 4.2.2.4. Файл table\_rows.js

```
import Utils from "common/Utils/Utils.js";
import { TargetDatabase, InternalDatabase } from "common/postgreSQL/postgreSQL.js";
```

```
import get_table_name from "../utils/database/get_table_name.js";
import table_rows_query from "../utils/table_rows_query.js";
```

```
function transform_data(data)
{
  if (typeof data !== 'object') return data;
  if (data instanceof Utils.Interval) return data.toPostgres();

  for (const key in data) data[key] = transform_data(data[key]);
  return data;
}
```

```
async function table_rows(msg, socket, req)
{
  if (!req.authorization.permissions.includes("R")) return;
  const { schema, table } = await get_table_name(TargetDatabase, msg.tableid);
  const { query, params } = table_rows_query('*', "%I.%I", msg.filters, msg.sorts);
  const queryString = TargetDatabase.format(query, schema, table, ...params);

  if (socket.table_rows_message_cursor?.query !== queryString)
  {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

if (socket.table_rows_message_callback)
{
    socket.removeListener('close', socket.table_rows_message_callback);
    await socket.table_rows_message_callback();
}

socket.table_rows_message_cursor = await TargetDatabase.cursor(queryString);
socket.table_rows_message_callback = async () => await
socket.table_rows_message_cursor.end();
socket.on('close', socket.table_rows_message_callback);

await InternalDatabase.query(`INSERT INTO logs (type, data, userid, query, query_params)
VALUES ('SELECT', $1, $2, $3, $4)`,
    [`${schema}.${table}`, req.authorization.user_id, query, JSON.stringify([ schema, table,
...params ]) ]);
}

socket.send(JSON.stringify({ eventName: 'table_rows', data: transform_data(await
socket.table_rows_message_cursor.get(msg.limit)) }));
}

const SCHEMA = {
    additionalProperties: false,
    properties:
    {
        tableid: { type: "int32" },

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

limit: { type: "int32" },
filters:
{
  elements:
  {
    additionalProperties: false,
    properties:
    {
      name: { type: "string" },
      value: { type: "string" },
      comparison: { enum: [ "exact", "substring" ] }
    }
  }
},
sorts:
{
  elements:
  {
    additionalProperties: false,
    properties:
    {
      name: { type: "string" },
      order: { enum: [ "asc", "desc" ] }
    }
  }
},

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
};

export default [
    { requestName: "table_rows", schema: SCHEMA, handler: table_rows }
];

```

#### 4.2.2.5. Файл logs.js

```

import { InternalDatabase } from "common/postgreSQL/postgreSQL.js";

import table_rows_query from "./utils/table_rows_query.js";

async function logs(msg, socket, req)
{
    if (!req.authorization.permissions.includes("A")) return;
    if (msg.sorts.length == 0) msg.sorts = [ { name: "date", order: "desc" } ];
    const { query, params } = table_rows_query(`logs.*, users.login AS username`, "logs INNER
JOIN users ON users.id = logs.userid", msg.filters, msg.sorts);
    const queryString = InternalDatabase.format(query, ...params);

    if (socket.logs_message_cursor?.query != queryString)
    {
        if (socket.logs_message_callback)
        {
            socket.removeListener('close', socket.logs_message_callback);
            await socket.logs_message_callback();
        }

        socket.logs_message_cursor = await InternalDatabase.cursor(queryString);
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

socket.logs_message_callback = async () => await socket.logs_message_cursor.end();

socket.on('close', socket.logs_message_callback);

}

socket.send(JSON.stringify({ eventName: 'logs', data: await
socket.logs_message_cursor.get(msg.limit) }));

}

const SCHEMA = {
  additionalProperties: false,
  properties:
  {
    limit: { type: "int32" },
    filters:
    {
      elements:
      {
        additionalProperties: false,
        properties:
        {
          name: { type: "string" },
          value: { type: "string" },
          comparison: { enum: [ "exact", "substring" ] }
        }
      }
    }
  },

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    sorts:
    {
      elements:
      {
        additionalProperties: false,
        properties:
        {
          name: { type: "string" },
          order: { enum: [ "asc", "desc" ] }
        }
      }
    },
  },
  optionalProperties:
  {
    tableid: { type: "int32" }
  }
};

export default [
  { requestName: "logs", schema: SCHEMA, handler: logs }
];

```

#### 4.2.3. Файл main.js

```

function get_main(req, res) { return req.authorization ? res.redirect("/database") : res.render("oauth.ejs");
}

```

```

function post_oauth(req, res) { return res.login(req.body.login, req.body.password); }

```

```

function get_logout(req, res) { return res.logout(); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

import { types as schema_types, EMPTY_GET_SCHEMA } from "../utils/schemas/schemas.js";
const POST_OAUTH_SCHEMA = {
  body:
  {
    type: "object",
    required: [ "authentication", "login", "password" ],
    additionalProperties: false,
    properties:
    {
      "authentication": schema_types.authentication,
      "login": { type: "string", minLength: 1, maxLength: 100 },
      "password": { type: "string", minLength: 1 }
    }
  }
};
export default [
  { method: "GET", paths: [ "/", "/main" ], access: "public", schema: EMPTY_GET_SCHEMA,
    handler: get_main },
  { method: "POST", path: "/oauth", access: "public", schema: POST_OAUTH_SCHEMA, handler:
    post_oauth },
  { method: "GET", path: "/logout", access: "public", schema: EMPTY_GET_SCHEMA, handler:
    get_logout }
];

```

#### 4.2.4. Файл database.js

```

import config from "common/configs/config.js";
import { TargetDatabase, InternalDatabase } from "common/postgreSQL/postgreSQL.js";

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

import get_database_info from "./utils/database/get_database_info.js";
import get_table_layout from "./utils/database/get_table_layout.js";
import get_table_name from "./utils/database/get_table_name.js";

async function get_database(req, res)
{
  if (!req.authorization.permissions.includes("R")) return res.error(403);
  return res.render("database.ejs", await get_database_info(TargetDatabase));
}

async function get_table(req, res)
{
  if (!req.authorization.permissions.includes("R")) return res.error(403);
  const [ { database_name, tables }, columns ] = await Promise.all([
    get_database_info(TargetDatabase), get_table_layout(TargetDatabase, "id", req.query.id) ]);
  return res.render("table.ejs", { database_name, tables, columns });
}

import compile_websocket_handler from './utils/compile_websocket_handler.js';
const weboscket_data_handler = await compile_websocket_handler("websocket_data_routes");
function websocket_data(connection, req)
{
  const errorHandler = err => { if (config.stage === "testing") console.error(err); };

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
connection.socket.on("message", message => websocket_data_handler(connection, req,
message.toString()).catch(errorHandler));
```

```
const cfg = { graph_records: config.graph_records, page_size: config.page_size, update_interval:
config.update_interval };

connection.socket.send(JSON.stringify({ eventName: "config", data: cfg }));

}
```

```
import Ajv from "ajv/dist/jtd.js";

const ajv = new Ajv();

const actionsParser = ajv.compileParser({

  elements:

  {

    discriminator: "type",

    mapping:

    {

      "INSERT": { properties: { data: { values: { type: "string" } } } },

      "DELETE": { properties: { id: { values: { type: "string" } } } },

      "UPDATE": { properties: { data: { values: { type: "string" } }, id: { values: { type: "string" } } }

    }

  }

});

async function post_data(req, res)

{
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

const actions = actionsParser(req.body.actions);

if (!actions) return res.error(400);

if (actions.length === 0) return res.redirect(`/table?id=${req.body.table}`);

if (!req.authorization.permissions.includes("I") && actions.find(action => action.type === "INSERT"))
return res.error(403);

if (!req.authorization.permissions.includes("U") && actions.find(action => action.type ===
"UPDATE")) return res.error(403);

if (!req.authorization.permissions.includes("D") && actions.find(action => action.type ===
"DELETE")) return res.error(403);


const { schema, table } = await get_table_name(TargetDatabase, req.body.table);

function get_action_query(action)
{
  function build_condition(identifier = { })
  {
    return {
      conditions: Object.values(identifier).map(value => Array.isArray(value) ? `%I =
ARRAY[%L]::text[]` : `%I = %L`).join(" AND "),
      params: Object.entries(identifier).flat()
    }
  }

  switch (action.type)
  {
    case "DELETE":
    {
      const { conditions, params: conditionsParams } = build_condition(action.id);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

return {
  query: `DELETE FROM %I.%I WHERE ${conditions}`,
  params: [ schema, table, ...conditionsParams ]
};
}
case "INSERT":
{
  return {
    query: `INSERT INTO %I.%I (${Object.keys(action.data).fill("%I").join(', ')})
OVERRIDING USER VALUE
VALUES (${Object.values(action.data).map(value => Array.isArray(value) ?
`ARRAY[%L]::text[]` : "%L")})`,
    params: [ schema, table, ...Object.keys(action.data), ...Object.values(action.data) ]
  }
}
case "UPDATE":
{
  const { conditions, params: conditionsParams } = build_condition(action.id);
  return {
    query: `UPDATE %I.%I SET ${Object.values(action.data).map(value =>
Array.isArray(value) ? `%I = ARRAY[%L]::text[]` : "%I = %L").join(", ")} WHERE ${conditions}`,
    params: [ schema, table, ...(Object.entries(action.data).flat()), ...conditionsParams ]
  }
}
default: throw `Unknown action type ${action.type}`;
}
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

let queries = [ ], logs = [ ];
for (const action of actions)
{
    if (action.data) action.data = Object.fromEntries(Object.entries(action.data).map(entry => [
entry[0], entry[1] || null ]));
    const { query, params } = get_action_query(action);
    queries.push(TargetDatabase.format(query, ...params));
    logs.push(InternalDatabase.format(`INSERT INTO logs (type, data, userid, query, query_params)
VALUES (%L, %L, %L, %L, %L)`, action.type, `${schema}.${table}`, req.authorization.user_id, query,
JSON.stringify(params)));
}
await TargetDatabase.query_multiple(queries);
await InternalDatabase.query_multiple(logs);
return res.redirect(`/table?id=${req.body.table}`);
}

```

```

import { types as schema_types, EMPTY_GET_SCHEMA } from "../utils/schemas/schemas.js";
const TABLE_GET_SCHEMA = {
    query:
    {
        type: "object",
        required: [ "id" ],
        additionalProperties: false,
        properties:
        {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    "id": { type: "integer" }
  }
}
};

const DATA_POST_SCHEMA = {
  body:
  {
    type: "object",
    required: [ "authentication", "table", "actions" ],
    additionalProperties: false,
    properties:
    {
      "authentication": schema_types.authentication,
      "table": { type: "integer" },
      "actions": { type: "string" }
    }
  }
};

export default [
  { method: "GET", path: "/database", access: "authorization", schema: EMPTY_GET_SCHEMA,
    handler: get_database },
  { method: "GET", path: "/table", access: "authorization", schema: TABLE_GET_SCHEMA, handler:
    get_table },
  { method: "GET", path: "/data", websocket: true, access: "authorization", schema:
    EMPTY_GET_SCHEMA, handler: websocket_data },
  { method: "POST", path: "/data", access: "authorization", schema: DATA_POST_SCHEMA, handler:
    post_data }
];

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
];
```

#### 4.2.5. Файл admin.js

```
import bcrypt from 'bcrypt';
```

```
import config from 'common/configs/config.js';
```

```
import { InternalDatabase, TargetDatabase } from 'common/postgreSQL/postgreSQL.js';
```

```
import get_database_info from './utils/database/get_database_info.js';
```

```
import get_table_layout from './utils/database/get_table_layout.js';
```

```
async function get_admin_accounts(req, res)
```

```
{
```

```
  if (!req.authorization.permissions.includes("A")) return res.error(403);
```

```
  const [ { database_name, tables }, accounts ] = await Promise.all([
    get_database_info(TargetDatabase), InternalDatabase.query(`SELECT id, login, permissions FROM
    users ORDER BY id ASC`) ]);
```

```
  return res.render("admin/accounts.ejs", { database_name, tables, accounts });
```

```
}
```

```
async function delete_admin_accounts(req, res)
```

```
{
```

```
  if (!req.authorization.permissions.includes("A")) return res.error(403);
```

```
  await InternalDatabase.query(`DELETE FROM users WHERE id = $1`, [ req.body.id ]);
```

```
  return res.redirect("/admin/accounts");
```

```
}
```

```
async function create_admin_accounts(req, res)
```

```
{
```

```
  if (!req.authorization.permissions.includes("A")) return res.error(403);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

const permissions = [ "R", "I", "U", "D", "A" ].map(permission => (req.body.permissions ?? [
]).includes(permission));

const password = await bcrypt.hash(req.body.password, config.bcrypt.saltRounds);

await InternalDatabase.query(`INSERT INTO users (login, password, read, insert, update, delete,
admin) VALUES ($1, $2, $3, $4, $5, $6, $7)`, [ req.body.login, password, ...permissions ]);

return res.redirect("/admin/accounts");
}

async function edit_admin_accounts(req, res)
{
  if (!req.authorization.permissions.includes("A")) return res.error(403);

  const permissions = [ "R", "I", "U", "D", "A" ].map(permission => (req.body.permissions ?? [
]).includes(permission));

  if (req.body.password)
  {
    const password = await bcrypt.hash(req.body.password, config.bcrypt.saltRounds);

    await InternalDatabase.query(`UPDATE users SET (login, password, read, insert, update, delete,
admin) = ($1, $2, $3, $4, $5, $6, $7) WHERE id = $8`, [ req.body.login, password, ...permissions,
req.body.id ]);
  }
  else
  {
    await InternalDatabase.query(`UPDATE users SET (login, read, insert, update, delete, admin) =
($1, $2, $3, $4, $5, $6) WHERE id = $7`, [ req.body.login, ...permissions, req.body.id ]);
  }

  return res.redirect("/admin/accounts");
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

async function get_admin_logs(req, res)
{
  if (!req.authorization.permissions.includes("A")) return res.error(403);

  const [ { database_name, tables }, logsColumns ] = await Promise.all([
    get_database_info(TargetDatabase), get_table_layout(InternalDatabase, "name", "logs") ]);

  const columns = [
    {
      label: 'Тип',
      ...logsColumns.find(column => column.name === 'type')
    },
    {
      label: 'Элементы БД',
      ...logsColumns.find(column => column.name === 'data')
    },
    {
      label: 'Пользователь',
      name: 'username',
      type: { id: 1043, mod: 104 }
    },
    {
      label: 'SQL-запрос',
      ...logsColumns.find(column => column.name === 'query')
    },
    {
      label: 'Параметры запроса',
      ...logsColumns.find(column => column.name === 'query_params')
    },
  ],

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

    {
      label: 'Дата',
      ...logsColumns.find(column => column.name === 'date')
    }
  ];

  return res.render("admin/logs.ejs", { database_name, tables, columns });
}

import { types as schema_types, EMPTY_GET_SCHEMA } from "../utils/schemas/schemas.js";
const ACCOUNTS_DELETE_SCHEMA =
{
  body:
  {
    type: "object",
    required: [ "authentication", "id" ],
    additionalProperties: false,
    properties:
    {
      "authentication": schema_types.authentication,
      "id": schema_types.uinteger
    }
  }
};

const ACCOUNTS_CREATE_SCHEMA =
{

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

body:
{
  type: "object",
  required: [ "authentication", "login", "password" ],
  additionalProperties: false,
  properties:
  {
    "authentication": schema_types.authentication,
    "login": { type: "string", minLength: 1, maxLength: 100 },
    "password": { type: "string", minLength: 1 },
    "permissions": { type: "array", maxItems: 5, uniqueItems: true, items: { type: "string",
minLength: 1, maxLength: 1, enum: [ "R", "I", "U", "D", "A", "-" ] } },
  }
}
};

const ACCOUNTS_EDIT_SCHEMA =
{
  body:
  {
    type: "object",
    required: [ "authentication", "id", "login" ],
    additionalProperties: false,
    properties:
    {
      "authentication": schema_types.authentication,
      "id": schema_types.uinteger,
      "login": { type: "string", minLength: 1, maxLength: 100 },

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    "password": { type: "string" },
    "permissions": { type: "array", maxItems: 5, uniqueItems: true, items: { type: "string",
minLength: 1, maxLength: 1, enum: [ "R", "I", "U", "D", "A", "-" ] } },
  }
}
};

export default [
  { method: "GET", path: "/admin/accounts", access: "authorization", schema:
EMPTY_GET_SCHEMA, handler: get_admin_accounts },
  { method: "GET", path: "/admin/logs", access: "authorization", schema: EMPTY_GET_SCHEMA,
handler: get_admin_logs },

  { method: "POST", path: "/admin/accounts/delete", access: "authorization", schema:
ACCOUNTS_DELETE_SCHEMA, handler: delete_admin_accounts },
  { method: "POST", path: "/admin/accounts/create", access: "authorization", schema:
ACCOUNTS_CREATE_SCHEMA, handler: create_admin_accounts },
  { method: "POST", path: "/admin/accounts/edit", access: "authorization", schema:
ACCOUNTS_EDIT_SCHEMA, handler: edit_admin_accounts }
];

```

### 4.3. Директория static

#### 4.3.1. Директория css

##### 4.3.1.1. Файл general.css

@media (prefers-reduced-motion: reduce)

```

{
  *, *::before, *::after
  {
    animation-duration: 0.01ms !important;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

animation-iteration-count: 1 !important;
transition-duration: 0.01ms !important;
scroll-behavior: auto !important;
}
}

```

@layer normalization

```

{
:root, ::backdrop
{
--brand-color: hsl(214, 90%, 50%);
--background-color: hsl(214, 8%, 8%);
--surface-color: hsl(214, 11%, 11%);
--border-color: hsl(214, 3%, 55%);
--shadow-color: hsl(214, 15%, 30%);

--text-brand-color: hsl(214, 90%, 75%);
--text-major-color: hsl(214, 0%, 100%);
--text-minor-color: hsl(214, 5%, 75%);
--text-placeholder-color: hsl(214, 10%, 60%);
--text-inverse-color: hsl(214, 2%, 10%);

--scrollbar-color: hsla(214, 95%, 50%, 0.85);

--color-red: hsl(353, 75%, 53%);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

--color-yellow: hsl(47, 75%, 53%);

--color-purple: hsl(302, 75%, 53%);

--chart-brand: #0d70f2;

--chart-border: #898c90;

--chart-shadow: #414b58;

--chart-text-brand: #86b8f9;

--chart-text-major: #ffffff;

--chart-text-minor: #bcbfc2;

--chart-red: #f53d53;

--chart-yellow: #f5cd3d;

--chart-purple: #f53def;

accent-color: var(--brand-color); color-scheme: light dark;

--line-height: 1.66;

/\* [ 400; 1900; 3400 ] \*/

--font-size-small: clamp(16rem / 16, (236rem + 16vw) / 240, 18rem / 16); /\* [ 16, 17, 18 ] \*/

--font-size-regular: clamp(16rem / 16, (232rem + 32vw) / 240, 20rem / 16); /\* [ 16, 18, 20 ]

\*/

--font-size-medium: clamp(18rem / 16, (258rem + 48vw) / 240, 24rem / 16); /\* [ 18, 21, 24 ]

\*/

--font-size-big: clamp(21rem / 16, (299rem + 64vw) / 240, 29rem / 16); /\* [ 21, 25, 29 ] \*/

--font-size-large: clamp(25rem / 16, (355rem + 80vw) / 240, 35rem / 16); /\* [ 25, 30, 35 ] \*/

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
--font-size-huge: clamp(30rem / 16, (426rem + 96vw) / 240, 42rem / 16); /* [ 30, 36, 42 ] */
--font-size-enormous: clamp(36rem / 16, (512rem + 112vw) / 240, 50rem / 16); /* [ 36, 43, 50 ] */
```

```
--vertical-sizing-small: calc(var(--font-size-small) / 2.20);
--vertical-sizing-regular: calc(var(--font-size-regular) / 2.00);
--vertical-sizing-medium: calc(var(--font-size-medium) / 1.95);
--vertical-sizing-big: calc(var(--font-size-big) / 1.90);
--vertical-sizing-large: calc(var(--font-size-large) / 1.90);
--vertical-sizing-huge: calc(var(--font-size-huge) / 1.90);
--vertical-sizing-enormous: calc(var(--font-size-enormous) / 1.90);
```

```
--horizontal-sizing-small: calc(var(--font-size-small) / 1.20);
--horizontal-sizing-regular: calc(var(--font-size-regular) / 1.05);
--horizontal-sizing-medium: calc(var(--font-size-medium) / 1.00);
--horizontal-sizing-big: calc(var(--font-size-big) / 1.00);
--horizontal-sizing-large: calc(var(--font-size-large) / 1.00);
--horizontal-sizing-huge: calc(var(--font-size-huge) / 1.00);
--horizontal-sizing-enormous: calc(var(--font-size-enormous) / 1.00);
```

```
--padding-vertical-small: var(--vertical-sizing-small);
--padding-vertical-regular: var(--vertical-sizing-regular);
--padding-vertical-medium: var(--vertical-sizing-medium);
--padding-vertical-big: var(--vertical-sizing-big);
--padding-vertical-large: var(--vertical-sizing-large);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

--padding-vertical-huge: var(--vertical-sizing-huge);
--padding-vertical-enormous: var(--vertical-sizing-enormous);

--padding-horizontal-small: var(--horizontal-sizing-small);
--padding-horizontal-regular: var(--horizontal-sizing-regular);
--padding-horizontal-medium: var(--horizontal-sizing-medium);
--padding-horizontal-big: var(--horizontal-sizing-big);
--padding-horizontal-large: var(--horizontal-sizing-large);
--padding-horizontal-huge: var(--horizontal-sizing-huge);
--padding-horizontal-enormous: var(--horizontal-sizing-enormous);

--padding-horizontal-small-wide: calc(2.5 * var(--horizontal-sizing-small));
--padding-horizontal-regular-wide: calc(2.5 * var(--horizontal-sizing-regular));
--padding-horizontal-medium-wide: calc(2.5 * var(--horizontal-sizing-medium));
--padding-horizontal-big-wide: calc(2.5 * var(--horizontal-sizing-big));
--padding-horizontal-large-wide: calc(2.5 * var(--horizontal-sizing-large));
--padding-horizontal-huge-wide: calc(2.5 * var(--horizontal-sizing-huge));
--padding-horizontal-enormous-wide: calc(2.5 * var(--horizontal-sizing-enormous));

--padding-small: var(--padding-vertical-small) var(--padding-horizontal-small);
--padding-regular: var(--padding-vertical-regular) var(--padding-horizontal-regular);
--padding-medium: var(--padding-vertical-medium) var(--padding-horizontal-medium);
--padding-big: var(--padding-vertical-big) var(--padding-horizontal-big);
--padding-large: var(--padding-vertical-large) var(--padding-horizontal-large);
--padding-huge: var(--padding-vertical-huge) var(--padding-horizontal-huge);
--padding-enormous: var(--padding-vertical-enormous) var(--padding-horizontal-enormous);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

--padding-small-wide: var(--padding-vertical-small)
var(--padding-horizontal-small-wide);

--padding-regular-wide: var(--padding-vertical-regular)
var(--padding-horizontal-regular-wide);

--padding-medium-wide: var(--padding-vertical-medium)
var(--padding-horizontal-medium-wide);

--padding-big-wide: var(--padding-vertical-big) var(--padding-horizontal-big-wide);
--padding-large-wide: var(--padding-vertical-large) var(--padding-horizontal-large-wide);
--padding-huge-wide: var(--padding-vertical-huge) var(--padding-horizontal-huge-wide);
--padding-enormous-wide: var(--padding-vertical-enormous)
var(--padding-horizontal-enormous-wide);

```

```

--margin-vertical-small: calc(1.75 * var(--vertical-sizing-small));
--margin-vertical-regular: calc(1.75 * var(--vertical-sizing-regular));
--margin-vertical-medium: calc(1.75 * var(--vertical-sizing-medium));
--margin-vertical-big: calc(1.75 * var(--vertical-sizing-big));
--margin-vertical-large: calc(1.75 * var(--vertical-sizing-large));
--margin-vertical-huge: calc(1.75 * var(--vertical-sizing-huge));
--margin-vertical-enormous: calc(1.75 * var(--vertical-sizing-enormous));

```

```

--margin-horizontal-small: calc(1.75 * var(--horizontal-sizing-small));
--margin-horizontal-regular: calc(1.75 * var(--horizontal-sizing-regular));
--margin-horizontal-medium: calc(1.75 * var(--horizontal-sizing-medium));
--margin-horizontal-big: calc(1.75 * var(--horizontal-sizing-big));
--margin-horizontal-large: calc(1.75 * var(--horizontal-sizing-large));
--margin-horizontal-huge: calc(1.75 * var(--horizontal-sizing-huge));

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

--margin-horizontal-enormous: calc(1.75 * var(--horizontal-sizing-enormous));

--margin-small:  var(--margin-vertical-small)  var(--margin-horizontal-small);
--margin-regular: var(--margin-vertical-regular) var(--margin-horizontal-regular);
--margin-medium: var(--margin-vertical-medium) var(--margin-horizontal-medium);
--margin-big:    var(--margin-vertical-big)   var(--margin-horizontal-big);
--margin-large:  var(--margin-vertical-large) var(--margin-horizontal-large);
--margin-huge:   var(--margin-vertical-huge)  var(--margin-horizontal-huge);
--margin-enormous: var(--margin-vertical-enormous) var(--margin-horizontal-enormous);

--interactive-min-height: calc(var(--line-height) * var(--font-size-regular) + 2 *
var(--vertical-sizing-regular));

--border-radius-small: 5px;
--border-radius-regular: 10px;
}

:focus-visible { outline: 2px solid var(--brand-color) !important; outline-offset: 1px !important;
border-radius: var(--border-radius-small); }

::selection { background-color: var(--brand-color); color: var(--text-major-color); fill:
var(--text-major-color); }

::target-text { background-color: var(--brand-color); color: var(--text-major-color); fill:
var(--text-major-color); }

::-webkit-calendar-picker-indicator, ::-webkit-clear-button, ::-webkit-inner-spin-button,
::-webkit-outer-spin-button { color: var(--brand-color); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

::-webkit-file-upload-button { -webkit-appearance: button; font: inherit; }

::-webkit-scrollbar { background-color: transparent; }

::-webkit-scrollbar:horizontal { height: max(10px, 0.85vw); }

::-webkit-scrollbar:vertical { width: max(10px, 0.85vw); }

::-webkit-scrollbar-thumb
{
    min-height: 15%; border: calc(max(10px, 0.85vw) / 4) solid transparent; border-radius:
calc(max(10px, 0.85vw) / 2);
    background-clip: content-box; background-color: var(--scrollbar-color);
}

::-webkit-scrollbar-thumb:hover { background-color: var(--brand-color); }

::-webkit-scrollbar-button, ::-webkit-scrollbar-corner { display: none; }

*, *::before, *::after
{
    padding: 0; margin: 0; box-sizing: border-box;
    scroll-behavior: smooth; scrollbar-width: thin; scrollbar-color: var(--scrollbar-color)
var(--background-color);
    transition-duration: 0.25s; transition-timing-function: ease-out; transition-property:
var(--transitioned), background-color, color, border-color, text-decoration-color; --transitioned:
color;
}

* { color: var(--text-major-color); font-family: inherit; font-stretch: inherit; font-style: inherit;
font-variant: inherit; font-weight: inherit; line-height: inherit; }

*::before, *::after { display: inline-block; color: var(--text-major-color); }

body { font-size: var(--font-size-regular); font-weight: 500; font-family: Inter, sans-serif,
system-ui; line-height: var(--line-height); text-align: left; text-size-adjust: 100%; }

main { display: block; padding: var(--padding-large); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
img, picture { flex-shrink: 0; max-inline-size: 100%; display: block; overflow: hidden; }
```

```
img:not([alt]) { outline: 10px solid red; }
```

```
button { appearance: button; }
```

```
strong { font-weight: 700; }
```

```
hr { margin: var(--margin-vertical-regular) 0; border: none; background-color:
var(--text-major-color); height: 1px; }
```

```
[hidden], input[type=hidden] { display: none !important; content-visibility: hidden !important; }
```

```
@media (min-width: 1000px)
```

```
{
```

```
  .desktop_hidden { display: none !important; }
```

```
}
```

```
}
```

```
@layer surface
```

```
{
```

```
  .surface, .surface_rounded { display: block; padding: var(--padding-large); background-color:
var(--surface-color); box-shadow: 0px 0px 0px 1px var(--shadow-color); }
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
.surface_rounded { border-radius: var(--border-radius-regular); }
}
```

```
@layer headings
```

```
{
  h1, h2, h3, figcaption { color: var(--text-major-color); }
  h1 { font-size: var(--font-size-large); font-weight: 800; }
  h2 { font-size: var(--font-size-big); font-weight: 700; }
  h3 { font-size: var(--font-size-medium); font-weight: 600; }
}
```

```
@layer lists
```

```
{
  ol, ul, menu { list-style-position: inside; }
  li > :is(ol, ul, menu) { padding-left: var(--padding-horizontal-regular); padding-right:
var(--padding-horizontal-regular); }
```

```
dt { float: left; font-weight: 600; color: var(--text-minor-color); }
dt::after { content: ':'; margin-right: 0.33ch; color: var(--text-minor-color); }
dd { color: var(--text-brand-color); }
```

```
.no_bullets { list-style-type: none; }
.no_bullets > * { margin-top: calc(var(--vertical-sizing-regular)); margin-bottom:
calc(var(--vertical-sizing-regular)); }
.no_bullets > li { line-height: 0; }
.no_bullets > li > * { line-height: var(--line-height); }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

@layer controls

{

@layer defaults

{

a, button, .control\_inline, .control\_block, .control\_primary, .control\_secondary

{

display: inline-block; text-decoration: underline; text-decoration-color: transparent;

text-align: left; cursor: pointer;

background: transparent; border: thin solid transparent;

}

}

@layer control\_inline

{

/\* Default for <a> is .control\_inline \*/

a:not(.control\_block, .control\_primary, .control\_secondary), .control\_inline { text-decoration: underline; }

:is(a:not(.control\_block, .control\_primary, .control\_secondary), .control\_inline):is(:hover, :visited) { color: var(--text-brand-color); }

:is(a:not(.control\_block, .control\_primary, .control\_secondary), .control\_inline):is(:hover, :visited)::before { color: var(--text-brand-color); }

:is(a:not(.control\_block, .control\_primary, .control\_secondary), .control\_inline):is(:hover, :visited)::after { color: var(--text-brand-color); }

:is(a:not(.control\_block, .control\_primary, .control\_secondary), .control\_inline):is(:hover, :visited) > \* { color: var(--text-brand-color); }

}

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

@layer block_defaults
{
    button:not(.control_inline), .control_block, .control_primary, .control_secondary
    {
        font-size: var(--font-size-small); padding: var(--padding-regular); border-radius:
var(--border-radius-regular); min-height: var(--interactive-min-height);
        text-overflow: ellipsis; overflow-x: hidden; white-space: nowrap;
    }

    :is(button:not(.control_inline), .control_block, .control_primary, .control_secondary) > span {
text-overflow: ellipsis; overflow-x: hidden; white-space: nowrap; }
}

@layer control_block
{
    .control_block:hover, .control_block:hover::before, .control_block:hover::after,
.control_block:hover > * { color: var(--text-brand-color); }

    .control_block:hover { text-decoration-color: currentColor; }
}

@layer control_secondary
{
    :is(a, button).control_secondary { display: inline-flex; align-items: center; justify-content:
center; text-align: center; }

    .control_secondary { background-color: transparent; border-color: var(--brand-color); }

    :is(a, button):not(:disabled).control_secondary:hover { background-color: var(--brand-color);
border-color: transparent; }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        :is(a, button):not(:disabled).control_secondary:hover::selection { background-color:
var(--text-major-color); color: var(--brand-color); fill: var(--brand-color); }

    }

/* Default for <button> is .control_primary */

@layer control_primary
{
    button:not(.control_inline, .control_block, .control_secondary), :is(a, button).control_primary
{ display: inline-flex; align-items: center; justify-content: center; text-align: center; }

    button:not(.control_inline, .control_block, .control_secondary), .control_primary {
background-color: var(--brand-color); border-color: transparent; }

    :is(button:not(.control_inline, .control_block, .control_secondary),
.control_primary)::selection { background-color: var(--text-major-color); color: var(--brand-color);
fill: var(--brand-color); }

    :is(button:not(.control_inline, .control_block, .control_secondary), :is(a,
button).control_primary):not(:disabled):hover { background-color: transparent; border-color:
var(--brand-color); }

}

}

@layer forms
{
    @layer labels
    {
        label { display: block; }

    }

    @layer elements
    {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

@layer defaults

```
{
  input:not([type='checkbox']), textarea, select, .checkboxWrapper
  {
    display: block; width: 100%; min-width: 175px; padding: var(--padding-regular);
margin: 4px 0; min-height: var(--interactive-min-height);

    font-size: var(--font-size-small); background-color: var(--background-color); border: thin
solid var(--border-color); border-radius: var(--border-radius-regular);
  }

  textarea, select[multiple] { resize: vertical; }

  ::placeholder { color: var(--text-placeholder-color); opacity: 1; }

  :is(input:not([type='checkbox']), select, textarea):disabled,
.checkboxWrapper:has(input[type='checkbox']:disabled) { text-align: center; border: thin solid
var(--brand-color); opacity: 1; }

  :is(input:not([type='checkbox']), select, textarea):invalid:not(:empty) { border: thin solid
var(--color-red); }

  input[type='checkbox'] { height: calc(0.8 * var(--interactive-min-height)); aspect-ratio: 1/1;
clip-path: circle(calc(48% - 3px)); }

  .checkboxWrapper { padding-top: 0; padding-bottom: 0; display: flex; align-items: center;
gap: 1ch; }

  .checkboxWrapper:has(input[type='checkbox']):focus-within { outline: 2px solid
var(--brand-color) !important; outline-offset: 1px !important; }

  .checkboxWrapper::after { content: "Не отмечено"; }

  .checkboxWrapper:has(input:checked)::after { content: "Отмечено"; }
}
```

@layer color

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

{
  input[type='color'] { padding: 0; }
  ::-webkit-color-swatch-wrapper { padding: 0; }
  ::-moz-color-swatch-wrapper { padding: 0; }
  ::-webkit-color-swatch { border-radius: var(--border-radius-regular); }
  ::-moz-color-swatch { border-radius: var(--border-radius-regular); }
}

@layer select
{
  select:not([multiple]) { border: none; border-right: 1ch solid transparent !important;
box-shadow: 0 0 0 1px var(--border-color); }

  @-moz-document url-prefix()
  {
    select:not([multiple]) { padding-top: calc(var(--padding-vertical-regular) + 0.33em); }
  }

  option { padding: var(--padding-regular); border-radius: var(--border-radius-small);
margin: 4px 0; font-size: var(--font-size-small); }

  option:checked { background-color: var(--brand-color); }

  option:is(:active, :hover, :focus) { background-color: var(--text-brand-color); color:
var(--text-inverse-color); }

}

@layer forms
{

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    form { display: flex; flex-direction: column; justify-content: stretch; gap:
var(--vertical-sizing-regular); }
    }
}

```

@layer dropdowns

```

{
    @layer opener
    {
        .dropdown_arrow, .dropdown_arrow_horizontal { display: flex; align-items: center; gap:
var(--padding-horizontal-small); }
        :is(.dropdown_arrow, .dropdown_arrow_horizontal)::after
        {
            content: "; margin-left: auto; rotate: 0deg; --transitioned: rotate; flex-shrink: 0;
            height: 0.75em; width: 0.75em; border-width: 3px; border-style: solid; border-color:
transparent currentColor currentColor transparent;
        }
        .dropdown_arrow::after { transform: translateY(-35%) rotate(45deg); }
        .dropdown_arrow_horizontal::after { transform: translateX(-35%) rotate(-45deg); }

        .dropdown_opened::after { rotate: 180deg; }
    }
    @layer window
    {
        .dropdown_inplace { border-radius: var(--border-radius-regular); }

        .dropdown_nobg, .dropdown_bg
    }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
    position: absolute; z-index: 5; display: block; min-width: 175px; width: max-content;
max-width: min(100%, 96vw, 250px);

    padding: var(--padding-medium); filter: drop-shadow(0 0 10px var(--background-color));
border-radius: var(--border-radius-regular);
}

.dropdown_bg { background-color: var(--surface-color); box-shadow: 0px 0px 0px 1px
var(--shadow-color); }

.dropdown_nobg { max-height: min(calc(100vh - 100px), max(300px, calc(50vh - 66px)));
overflow-y: auto; }
}

@layer custom
{
    /*-----STRIPED BUTTON-----*/

    @layer striped_button
    {
        .striped_button
        {
            background: transparent; border: none; padding: 0; height: 40px; width: 40px;
            position: relative; display: inline-flex; flex-direction: column; justify-content:
space-evenly; align-items: center;
        }

        .striped_button > span { background: var(--text-major-color); border-radius:
var(--border-radius-small); width: 80%; height: 4px; }

        .striped_button:not(:disabled):hover > span { background: var(--brand-color); }
    }

    /*-----CROSS BUTTON-----*/

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

@layer cross_button
{
    .cross_button { position: relative; background: transparent; border: none; padding: 0;
height: 40px; width: 40px; }

    .cross_button > span
    {
        position: absolute; top: 50%; left: 10%; transform: translateY(-50%); width: 80%;
height: 4px;
        background: var(--text-major-color); border-radius: var(--border-radius-small);
    }

    .cross_button::after
    {
        position: absolute; left: 50%; top: 10%; transform: translateX(-50%); height: 80%;
width: 4px;
        content: ""; background: var(--text-major-color); border-radius:
var(--border-radius-small);
    }

    .cross_button:not(:disabled):hover > span, .cross_button:not(:disabled):hover::after {
background: var(--brand-color); }
}
}

```

```

@layer tables

```

```

{
    table { min-width: 100%; border-spacing: 0; margin-top: var(--margin-vertical-regular); }

    table thead { position: sticky; z-index: 2; top: calc(-1 * var(--padding-vertical-large) + 1px);
background: var(--surface-color); filter: drop-shadow(0 -10px 0px var(--background-color)); }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

table :is(th, td) { position: relative; padding: var(--padding-regular); width: max-content; }

table th:first-child { border-top-left-radius: var(--border-radius-regular); }
table th:last-child { border-top-right-radius: var(--border-radius-regular); }

table tr > * { border-bottom: thin solid var(--border-color); border-right: thin solid
var(--border-color); }

table > :first-child > :first-child > * { border-top: thin solid var(--shadow-color); }
table tr > :last-child { border-right: thin solid var(--shadow-color); }
table tr > :first-child { border-left: thin solid var(--shadow-color); }
}

@layer dialogs
{
  dialog
  {
    position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%); min-width: min(95vw,
60ch); width: min-content; max-width: min(95vw, 80ch); max-height: 85vh;

    padding: var(--padding-large); background-color: var(--surface-color); box-shadow: 0px 0px
0px 1px var(--shadow-color); border-radius: var(--border-radius-regular); border: none;
  }

  dialog::backdrop { background: var(--surface-color); opacity: 0.5; }
  dialog:not([open]) { display: none !important; }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

@layer body

```
{
    body { background-color: var(--background-color); height: 100vh; display: flex; gap: 1px;
container-type: inline-size; container-name: body; }

    .main_body { flex-grow: 2; container-type: inline-size; container-name: main_body; display:
flex; flex-direction: column; }

    main { overflow: auto; max-width: 100%; }

    main > *:not(table) { position: sticky; left: 0; }
}
```

@layer header

```
{
    header { position: sticky; top: 0; z-index: 3; }
```

@layer main\_body\_header

```
{
    .main_body > header { display: grid; grid-template-columns: auto 1fr auto; gap:
var(--padding-regular); }

    .main_body > header .userinfo { line-height: normal; }

    .main_body > header .navigation { justify-content: center; }
```

@container main\_body (max-width: 900px)

```
{
    .main_body > header { grid-template-columns: auto 1fr; }

    .main_body > header > .navigation { display: none; order: -1; grid-column: 1 / -1; }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

@container main_body (max-width: 700px)
{
    .main_body > header { grid-template-columns: auto; }
    .main_body > header > .navigation { display: none; order: unset; grid-column: 1 / -1; }
}

@layer nav
{
    @layer default
    {
        body > nav { position: sticky; top: 0; height: 100vh; overflow: scroll; width: 320px;
max-width: 100vw; --transitioned: translate; }
    }

    @layer navigation_opener
    {
        #navigation_backdrop { position: fixed; left: 0; top: 0; width: 100vw; height: 100vh; z-index:
10; background: var(--background-color); opacity: 0.5; --transitioned: opacity, z-index; }

        @media not (min-width: 1000px)
        {
            body > nav { position: fixed; top: 0; left: 0; z-index: 11; translate: 0; }
            body > nav.closed { translate: calc(-100% - 1px); }
            #navigation_backdrop.closed { opacity: 0; z-index: -1; }
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
}
```

```
@layer graphs
```

```
{
```

```
  .graphs { display: grid; grid-template-columns: repeat(auto-fit, minmax(min(500px, 100%), 1fr)); gap: var(--margin-regular); }
```

```
  .graphs > div { aspect-ratio: 5/3; min-height: 0; padding: var(--padding-vertical-small) 0; }
```

```
}
```

```
@layer data_table
```

```
{
```

```
  @layer defaults
```

```
{
```

```
  .data_table :is(th, td):has(input:not([type='checkbox']), button, textarea, select, .checkboxWrapper, .dropdown_inplace), .data_table :is(th, td):empty { padding: unset; }
```

```
  .data_table :is(th, td) > :is(input:not([type='checkbox']), button, textarea, select, .checkboxWrapper) { border: none; border-radius: 0; box-shadow: none; margin: 4px; height: 100%; width: calc(100% - 8px); }
```

```
}
```

```
@layer dropdowns
```

```
{
```

```
  .data_table thead .dropdown_opener::after { color: var(--text-major-color); }
```

```
  .data_table thead .dropdown_opener * { text-decoration: underline; text-decoration-color: transparent; color: var(--text-major-color); }
```

```
  .data_table thead .dropdown_opener:hover { text-decoration-color: transparent; }
```

```
  .data_table thead .dropdown_opener:hover * { text-decoration-color: currentColor; }
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
.data_table:not(#logs_table) thead .dropdown_arrow { display: grid; grid-template-columns:
minmax(max-content, 1fr) auto; grid-template-rows: 1fr 1fr; row-gap: 0; }
```

```
.data_table:not(#logs_table) thead .dropdown_arrow::after { grid-row: 1 / -1; grid-column: 2;
margin-left: auto; }
```

```
.data_table thead .dropdown_inplace { border-radius: 0; padding: var(--padding-medium); }
```

```
.data_table thead .dropdown_inplace :is(input, textarea, select) { min-width: calc(175px - 2 *
var(--padding-horizontal-medium)); }

}
```

@layer identity

```
{

.data_table:not(#logs_table) :is(th, td):first-child { width: 150px; position: sticky; left: calc(-1
* var(--padding-horizontal-large)); z-index: 1; background: var(--surface-color); filter:
drop-shadow(-10px 0px 0px var(--background-color)); }
```

```
.data_table tbody [data-identity] > div { display: flex; align-items: center; justify-content:
space-between; gap: var(--padding-horizontal-regular); padding: 0
var(--padding-horizontal-regular); }
```

```
.data_table tbody [data-identity] button { min-height: unset; height: 40px; width: unset;
aspect-ratio: 1/1; }
```

```
.data_table tbody [data-identity] > div > span { margin-left: auto; }

}
```

@layer edits

```
{

.data_table [data-edited]:not([data-deleted]) :is(td, input:not([type='checkbox']), select,
textarea, .checkboxWrapper) { background: var(--color-yellow); color: var(--text-inverse-color); }
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

.data_table [data-edited]:not([data-deleted]) *::after, .data_table
[data-edited]:not([data-deleted]) *::before { color: var(--text-inverse-color); }

.data_table [data-edited]:not([data-deleted]) [data-identity] > div > span { color:
var(--text-inverse-color); }

.data_table [data-edited]:not([data-deleted]) [data-identity] :is(.striped_button:not(:hover) >
span, .striped_button:not(:hover)::after, .cross_button:not(:hover) > span,
.cross_button:not(:hover)::after) { background: var(--text-inverse-color); }

.data_table [data-deleted] :is(td, input:not([type='checkbox']), select, textarea,
.checkboxWrapper) { background: var(--color-red); }

}

}

```

@layer text

{

@layer color

{

```

.text_brand { color: var(--text-brand-color); }

.text_major { color: var(--text-major-color); }

.text_minor { color: var(--text-minor-color); }

.text_placeholder { color: var(--text-placeholder-color); }

.text_inverse { color: var(--text-inverse-color); }

.text_red { color: var(--color-red); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

.text_yellow { color: var(--color-yellow); }

.text_purple { color: var(--color-purple); }

}

@layer size
{
    .text_size_small { font-size: var(--font-size-small); }
    .text_size_regular { font-size: var(--font-size-regular); }
    .text_size_medium { font-size: var(--font-size-medium); }
    .text_size_big { font-size: var(--font-size-big); }
    .text_size_large { font-size: var(--font-size-large); }
    .text_size_huge { font-size: var(--font-size-huge); }
    .text_size_enormous { font-size: var(--font-size-enormous); }
}

@layer alignment
{
    .text_align_left { text-align: left; }
    .text_align_center { text-align: center; }
    .text_align_justify { text-align: justify; }
}

@layer other
{
    .text_nowrap { text-overflow: ellipsis; overflow-x: hidden; white-space: nowrap; }
    .text_nohover { text-decoration: none; cursor: revert; }
    .text_nodecoration { text-decoration: none; }
}
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

/\*-----SIZING-----\*/

@layer sizing

```
{
  .minwidth { width: min-content; }
  .maxwidth { width: 100%; }

  .maxheight { height: 100%; }
}
```

/\*-----BLOCKS-----\*/

@layer blocks

```
{
  .block { display: block; }
  .centered { margin-left: auto; margin-right: auto; }

  .flex_horizontal { display: flex; gap: var(--padding-horizontal-regular); align-items: center; }
  .flex_vertical { display: flex; flex-direction: column; gap: var(--padding-vertical-regular); }
  .flex_fill { flex-grow: 2; }

  .grid_1x2 { display: grid; grid-template-columns: 1fr 1fr; gap: var(--padding-horizontal-regular);
min-width: max-content; }
  .grid_1x2:has([hidden]) { grid-template-columns: 1fr; }

  .grid
{
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

display: grid; font-size: var(--font-size-regular); gap: var(--padding-regular);
grid-template-columns: repeat(auto-fit, minmax(clamp(
    175px + 2 * var(--padding-horizontal-large),
    (100% - 2 * var(--padding-horizontal-large) - (var(--grid-columns, 2) - 1) *
var(--padding-horizontal-regular)) / var(--grid-columns, 2),
    100% - 2 * var(--padding-horizontal-large)
), 1fr));
}

.grid > :is(h1, h2, h3, p) { grid-column: 1 / -1; }
.grid_columns2 { --grid-columns: 2; }
.grid_columns3 { --grid-columns: 3; }

.grid_fill_column { grid-row: 1 / -1; }
.grid_fill_row { grid-column: 1 / -1; }
}

/*-----PADDING/SIZING-----*/

@layer padding
{
    .padding_none { padding: 0; }

    .padding_vertical_small { padding-top: var(--padding-vertical-small); padding-bottom:
var(--padding-vertical-small); }

    .padding_vertical_regular { padding-top: var(--padding-vertical-regular); padding-bottom:
var(--padding-vertical-regular); }

    .padding_vertical_medium { padding-top: var(--padding-vertical-medium); padding-bottom:
var(--padding-vertical-medium); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

.padding_vertical_big    { padding-top: var(--padding-vertical-big);    padding-bottom:
var(--padding-vertical-big);    }

.padding_vertical_large  { padding-top: var(--padding-vertical-large);  padding-bottom:
var(--padding-vertical-large);  }

.padding_vertical_huge   { padding-top: var(--padding-vertical-huge);   padding-bottom:
var(--padding-vertical-huge);   }

.padding_vertical_enormous { padding-top: var(--padding-vertical-enormous); padding-bottom:
var(--padding-vertical-enormous); }

.padding_horizontal_small { padding-left: var(--padding-horizontal-small); padding-right:
var(--padding-horizontal-small); }

.padding_horizontal_regular { padding-left: var(--padding-horizontal-regular); padding-right:
var(--padding-horizontal-regular); }

.padding_horizontal_medium { padding-left: var(--padding-horizontal-medium);
padding-right: var(--padding-horizontal-medium); }

.padding_horizontal_big   { padding-left: var(--padding-horizontal-big);   padding-right:
var(--padding-horizontal-big);   }

.padding_horizontal_large  { padding-left: var(--padding-horizontal-large);  padding-right:
var(--padding-horizontal-large);  }

.padding_horizontal_huge   { padding-left: var(--padding-horizontal-huge);   padding-right:
var(--padding-horizontal-huge);   }

.padding_horizontal_enormous { padding-left: var(--padding-horizontal-enormous);
padding-right: var(--padding-horizontal-enormous); }

.padding_horizontal_small_wide { padding-left: var(--padding-horizontal-small-wide);
padding-right: var(--padding-horizontal-small-wide); }

.padding_horizontal_regular_wide { padding-left: var(--padding-horizontal-regular-wide);
padding-right: var(--padding-horizontal-regular-wide); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

.padding_horizontal_medium_wide { padding-left: var(--padding-horizontal-medium-wide);
padding-right: var(--padding-horizontal-medium-wide); }

.padding_horizontal_big_wide    { padding-left: var(--padding-horizontal-big-wide);
padding-right: var(--padding-horizontal-big-wide); }

.padding_horizontal_large_wide  { padding-left: var(--padding-horizontal-large-wide);
padding-right: var(--padding-horizontal-large-wide); }

.padding_horizontal_huge_wide   { padding-left: var(--padding-horizontal-huge-wide);
padding-right: var(--padding-horizontal-huge-wide); }

.padding_horizontal_enormous_wide { padding-left: var(--padding-horizontal-enormous-wide);
padding-right: var(--padding-horizontal-enormous-wide); }


.padding_small { padding: var(--padding-small); }
.padding_regular { padding: var(--padding-regular); }
.padding_medium { padding: var(--padding-medium); }
.padding_big { padding: var(--padding-big); }
.padding_large { padding: var(--padding-large); }
.padding_huge { padding: var(--padding-huge); }
.padding_enormous { padding: var(--padding-enormous); }


.padding_small_wide { padding: var(--padding-small-wide); }
.padding_regular_wide { padding: var(--padding-regular-wide); }
.padding_medium_wide { padding: var(--padding-medium-wide); }
.padding_big_wide { padding: var(--padding-big-wide); }
.padding_large_wide { padding: var(--padding-large-wide); }
.padding_huge_wide { padding: var(--padding-huge-wide); }
.padding_enormous_wide { padding: var(--padding-enormous-wide); }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
/*-----MARGIN-----*/
```

```
@layer margin
```

```
{

    .margin_vertical_small { margin-top: var(--margin-vertical-small); margin-bottom:
var(--margin-vertical-small); }

    .margin_vertical_regular { margin-top: var(--margin-vertical-regular); margin-bottom:
var(--margin-vertical-regular); }

    .margin_vertical_medium { margin-top: var(--margin-vertical-medium); margin-bottom:
var(--margin-vertical-medium); }

    .margin_vertical_big { margin-top: var(--margin-vertical-big); margin-bottom:
var(--margin-vertical-big); }

    .margin_vertical_large { margin-top: var(--margin-vertical-large); margin-bottom:
var(--margin-vertical-large); }

    .margin_vertical_huge { margin-top: var(--margin-vertical-huge); margin-bottom:
var(--margin-vertical-huge); }

    .margin_vertical_enormous { margin-top: var(--margin-vertical-enormous); margin-bottom:
var(--margin-vertical-enormous); }

    .margin_horizontal_small { margin-left: var(--margin-horizontal-small); margin-right:
var(--margin-horizontal-small); }

    .margin_horizontal_regular { margin-left: var(--margin-horizontal-regular); margin-right:
var(--margin-horizontal-regular); }

    .margin_horizontal_medium { margin-left: var(--margin-horizontal-medium); margin-right:
var(--margin-horizontal-medium); }

    .margin_horizontal_big { margin-left: var(--margin-horizontal-big); margin-right:
var(--margin-horizontal-big); }
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
.margin_horizontal_large { margin-left: var(--margin-horizontal-large); margin-right:
var(--margin-horizontal-large); }

.margin_horizontal_huge { margin-left: var(--margin-horizontal-huge); margin-right:
var(--margin-horizontal-huge); }

.margin_horizontal_enormous { margin-left: var(--margin-horizontal-enormous); margin-right:
var(--margin-horizontal-enormous); }
```

```
.margin_small { margin-left: var(--margin-small); }
.margin_regular { margin-left: var(--margin-regular); }
.margin_medium { margin-left: var(--margin-medium); }
.margin_big { margin-left: var(--margin-big); }
.margin_large { margin-left: var(--margin-large); }
.margin_huge { margin-left: var(--margin-huge); }
.margin_enormous { margin-left: var(--margin-enormous); }
}
```

```
/*-----BORDER RADIUS-----*/
```

```
@layer border_radius
```

```
{
  .border_radius_small { border-radius: var(--border-radius-small); }
  .border_radius_regular { border-radius: var(--border-radius-regular); }
}
```

#### 4.3.2. Директория js

##### 4.3.2.1. Директория utils

##### 4.3.2.1.1. Файл general.js

```
/*-----DROPDOWNS-----*/
```

```
function dropdown_open(dropdown)
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  dropdown.hidden = false;

  const buttons = document.querySelectorAll(`[data-dropdown="${dropdown.id}"]`);

  for (const button of buttons) { button.classList.add("dropdown_opened");
button.setAttribute("aria-expanded", "true"); }

  dropdown.focus();
}

function dropdown_close(dropdown)
{
  dropdown.hidden = true;

  const buttons = document.querySelectorAll(`[data-dropdown="${dropdown.id}"]`);

  for (const button of buttons) { button.classList.remove("dropdown_opened");
button.setAttribute("aria-expanded", "false"); }

}

function toggle_dropdown(ev)
{
  const id = ev.currentTarget.dataset.dropdown;
  const dropdown = document.getElementById(id);
  if (!dropdown) return;
  if (dropdown.hidden) dropdown_open(dropdown);
  else dropdown_close(dropdown);
}

Array.from(document.getElementsByClassName("dropdown_opener")).forEach(el =>
{
  el.addEventListener("click", toggle_dropdown, { "capture": false, "once": false,
"passive": true });

  const dropdown = document.getElementById(el.dataset.dropdown);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

if (!dropdown || dropdown.classList.contains("dropdown_inplace")) return;
dropdown.addEventListener("focusout", ev =>
{
    if (ev.currentTarget.contains(ev.relatedTarget) || (ev.relatedTarget &&
ev.relatedTarget.dataset.dropdown == ev.currentTarget.id)) return;
    dropdown_close(ev.currentTarget.id);
}, { "capture": false, "once": false, "passive": true });
dropdown.addEventListener("keydown", ev =>
{
    if (ev.key != 'Tab') return;
    setTimeout(() => { if (document.activeElement == el) dropdown_close(dropdown.id)
});
}, { "capture": false, "once": false, "passive": true });
});

```

```

/*-----NAVIGATION MENU OPEN/CLOSE-----*/
function open_mobile_navigation()
{
    document.getElementById("navigation").classList.remove("closed");
    document.getElementById("navigation_backdrop").classList.remove("closed");
}
function close_mobile_navigation()
{
    document.getElementById("navigation").classList.add("closed");
    document.getElementById("navigation_backdrop").classList.add("closed");
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
}
```

```
Array.from(document.getElementsByClassName("navigation_opener")).forEach(button =>
{
    button.addEventListener("click", open_mobile_navigation, { "capture": false, "once":
false, "passive": true });
});
Array.from(document.getElementsByClassName("navigation_closer")).forEach(button =>
{
    button.addEventListener("click", close_mobile_navigation, { "capture": false, "once":
false, "passive": true });
});
if (document.getElementById("navigation_backdrop"))
{
    document.getElementById("navigation_backdrop").addEventListener("click",
close_mobile_navigation, { "capture": false, "once": false, "passive": true });
}
```

```
/*-----DIALOGS-----*/
```

```
function toggle_dialog(ev)
{
    const dialog = document.getElementById(ev.currentTarget.dataset.dialog);
    if (dialog.open) return dialog.close();
    dialog.showModal();
    dialog.focus();
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

}
Array.from(document.querySelectorAll("[data-dialog]")).forEach(button =>
{
    button.addEventListener("click", toggle_dialog, { "capture": false, "once": false,
    "passive": true });
});

```

#### 4.3.2.1.2. Файл chart.js

```
export default class Chart
```

```

{
    static #load = new Promise(resolve => Chart.onLoad = resolve);

    #options; #data; #chart;
    constructor(element, records, update_interval, title, vAxisTitle, labels = { })
    {
        this.records = records;
        this.update_interval = update_interval;
        this.labels = labels;

        const styles = getComputedStyle(document.body);
        const colors =
        {
            brand: styles.getPropertyValue("--chart-brand").replaceAll(" ", ""),
            border: styles.getPropertyValue("--chart-border").replaceAll(" ", ""),
            shadow: styles.getPropertyValue("--chart-shadow").replaceAll(" ", ""),
            text_brand: styles.getPropertyValue("--chart-text-brand").replaceAll(" ", ""),
            text_major: styles.getPropertyValue("--chart-text-major").replaceAll(" ", ""),
            text_minor: styles.getPropertyValue("--chart-text-minor").replaceAll(" ", ""),

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

red: styles.getPropertyValue("--chart-red").replaceAll(" ", ""),
yellow: styles.getPropertyValue("--chart-yellow").replaceAll(" ", ""),
purple: styles.getPropertyValue("--chart-purple").replaceAll(" ", "")
};

this.#options = {
  chartArea: { top: '15%', height: '65%' },
  backgroundColor: 'transparent', fontSize: 16, focusTarget: "category",
  colors: [ colors.red, colors.yellow, colors.purple ],
  hAxis:
  {
    format: 'hh:mm:ss', textStyle: { color: colors.text_minor }, slantedText: true,
    gridlines: { color: colors.border }, minorGridlines: { color: colors.shadow },
    title: "Время", titleTextStyle: { fontSize: 18, color: colors.text_brand },
    minValue: new Date(Date.now() - this.records * this.update_interval),
  },
  interpolateNulls: true, lineWidth: 3, pointSize: 0, selectionMode: 'multiple',
  legend: { textStyle: { color: colors.text_major } }, tooltip: { textStyle: { fontName:
'unset' } },
  title: title, titleTextStyle: { fontSize: 21, color: colors.text_major },
  vAxis:
  {
    format: "short",
    textStyle: { color: colors.text_minor }, title: vAxisTitle, titleTextStyle: { fontSize:
18, color: colors.text_brand },
    gridlines: { color: colors.border }, minorGridlines: { color: colors.shadow }
  }
};

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    this.#adjustFontSize();

    window.addEventListener("resize", () => { this.#adjustFontSize(); this.#redraw(); }, {
"capture": false, "once": false, "passive": true });

    Chart.#load.then(this.#firstDrawChart.bind(this, element));
  }

  async #adjustFontSize()
  {
    const defaultFontSize =
Number(getComputedStyle(document.body).getPropertyValue("font-size").slice(0, -2));

    this.#options.fontSize = defaultFontSize - 2;

    this.#options.hAxis.titleTextStyle.fontSize =
this.#options.vAxis.titleTextStyle.fontSize = defaultFontSize;

    this.#options.titleTextStyle.fontSize = defaultFontSize + 2;
  }

  #firstDrawResolve; #firstDraw = new Promise(resolve => this.#firstDrawResolve =
resolve);

  async #firstDrawChart(element)
  {
    this.#data = new google.visualization.DataTable({ cols: [ { type: 'datetime', label:
'date' } ] });

    this.#chart = new google.visualization.LineChart(element);

    this.#firstDrawResolve();
  }

  #redraw() { this.#chart.draw(this.#data, this.#options); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

async addRecord(record)
{
  await this.#firstDraw;

  Object.keys(record).filter(column => this.#data.getColumnIndex(column) ==
-1).forEach(column => this.#data.addColumn('number', this.labels[column] ?? column,
column));

  let row = [ new Date(), ...new Array(this.#data.getNumberOfColumns() - 1) ];
  for (const column in record) row[this.#data.getColumnIndex(column)] =
Number(record[column]);
  this.#data.addRow(row);

  if (this.#options.hAxis.minValue)
  {
    const timeEstimate = new Date(Date.now() - this.records * this.update_interval);
    if (timeEstimate < this.#data.getValue(0, 0)) this.#options.hAxis.minValue =
timeEstimate;
    else delete this.#options.hAxis.minValue;
  }

  this.#data.removeRows(0, this.#data.getNumberOfRows() - 100);
  this.#redraw();
}
};

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
google.charts.load('current', { 'packages': [ 'corechart' ], 'language': 'ru', callback:
Chart.onLoad });
```

#### 4.3.2.1.3. Файл table.js

```
class TableRow
{
  constructor(template)
  {
    const clone = template.content.cloneNode(true);
    this.elements = Array.from(clone.children);

    this.deleteButton = clone.querySelector("[data-identity] button");

    this.changeHandler = this.changed.bind(this);
    this.inputs.forEach(input => input.addEventListener('input', this.changeHandler, {
"capture": false, "once": true, "passive": true }));
  }

  get inputs() { return this.elements.reduce((prev, cur) =>
prev.concat(Array.from(cur.querySelectorAll("[name]"))), [ ]); }

  #changeCallbacks = [ ];
  addChangeCallback(callback) { this.#changeCallbacks.push(callback); }
  removeChangeCallback(callback) { this.#changeCallbacks =
this.#changeCallbacks.filter(cb => cb != callback); }
  changed(ev) { this.#changeCallbacks.forEach(callback => callback(ev.currentTarget)); }

  getData()
  {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

function getInputPair(input)
{
  if (input.type === "checkbox") return [ input.name, input.checked ? "t" : "f"];
  if (input.type === "time" || input.type === "datetime-local") return [ input.name, new
Date(input.value) ];
  return [ input.name, input.value ];
}

return Object.fromEntries(this.inputs.filter(input => input.value !==
input.dataset.initial_value).map(getInputPair));
}

getInitialData() { return Object.fromEntries(this.inputs.map(input => [ input.name,
input.dataset.initial_value ])); }

getIdentifier(head)
{
  let keys = Array.from(head.children[0].children).filter(th =>
(th.children?.[0]?.children?.[1]?.innerText ?? "").includes("[PK]")).map(th =>
th.children[0].children[0].innerText);

  if (keys.length === 0) keys = Array.from(head.children[0].children).filter(th =>
(th.children?.[0]?.children?.[1]?.innerText ?? "").includes("[U]")).map(th =>
th.children[0].children[0].innerText);

  if (keys.length === 0) return this.getInitialData();

  let data = this.getInitialData();
  for (const key in data)
  {
    if (!keys.includes(key)) delete data[key];
  }
  return data;
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
};

class TableInsertRow extends TableRow
{
    constructor(template) { super(template); }

    #removeCallbacks = [ ];

    addRemoveCallback(callback) { this.#removeCallbacks.push(callback); }

    removeRemoveCallback(callback) { this.#removeCallbacks =
this.#removeCallbacks.filter(cb => cb != callback); }

    removed() { this.#removeCallbacks.forEach(callback => callback(this)); }

    showDeleteButton()
    {
        if (!this.deleteButton) return;

        this.deleteButton.addEventListener("click", this.removed.bind(this), { "capture": false,
"once": true, "passive": true });

        this.deleteButton.hidden = false;

        Table.showSaveButton();
    }
};

class TableDisplayRow extends TableRow
{
    constructor(template, data)
    {
        super(template); this.deleted = this.edited = false; this.#fillData(data);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    if (this.deleteButton) this.deleteButton.addEventListener("click",
this.remove.bind(this), { "capture": false, "once": false, "passive": true });
}

```

```

#fillData(data)

```

```

{
  for (const input of this.inputs)
  {
    const key = input.getAttribute("name"); data[key] ??= "";
    generalSwitch: switch (input.tagName)
    {
      case "INPUT":
      {
        switch (input.type)
        {
          case "date":
          {
            if (!data[key]) break generalSwitch;
            data[key] = new Date(data[key]);
            const value =
` ${data[key].getFullYear()}-${toLength(data[key].getMonth() + 1, 2,
"0")}-${toLength(data[key].getDate(), 2, "0")}`;
            input.value = value;
            break generalSwitch;
          }
          case "time":
          {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    if (!data[key]) break generalSwitch;
    data[key] = new Date(data[key]);
    data[key] = new Date(data[key].getTime() -
data[key].getTimezoneOffset() * 60 * 1000);

    const value = `${data[key].getHours()}:${data[key].getMinutes()}`;
    input.value = value;
    break generalSwitch;
}
case "datetime-local":
{
    if (!data[key]) break generalSwitch;
    data[key] = new Date(data[key]);
    data[key] = new Date(data[key].getTime() -
data[key].getTimezoneOffset() * 60 * 1000);

    const value = data[key].toISOString().substring(0,
data[key].toISOString().indexOf("T") + 6);
    input.value = value;
    break generalSwitch;
}
case "checkbox":
{
    input.checked = data[key];
    break generalSwitch;
}
}
}
default:

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    {
      const value = (typeof data[key] === 'object' ? JSON.stringify(data[key]) :
data[key]);
      input.value = value;
    }
  }
  input.dataset.initial_value = input.value;
}
}

```

changed()

```

{
  super.changed();
  this.elements.forEach(row => row.dataset.edited = true);
  this.edited = true;
  Table.showSaveButton();
}

```

remove()

```

{
  if (this.deleted)
  {
    this.elements.forEach(row => delete row.dataset.deleted);
    this.deleteButton.label = "Удалить строку";
    this.deleteButton.classList.remove("cross_button");
    this.deleteButton.classList.add("striped_button");
  }
  else

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  this.elements.forEach(row => row.dataset.deleted = true);
  this.deleteButton.label = "Вернуть строку";
  this.deleteButton.classList.add("cross_button");
  this.deleteButton.classList.remove("striped_button");
}
this.deleted = !this.deleted;
Table.showSaveButton();
}
};

```

export default class Table

```

{
  static #tables = [ ];
  static #saveButton;
  static registerSaveButton(button)
  {
    Table.#saveButton = button;
    button.form.addEventListener("submit", ev =>
    {
      ev.preventDefault();
      ev.currentTarget.elements.actions.value = JSON.stringify(Table.#tables.reduce((acc,
cur) => acc.concat(cur.getUpdateActions()), [ ]));
      ev.currentTarget.submit();
    }, { "capture": false, "once": false, "passive": false });
  }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

static showSaveButton() { this.#saveButton.hidden = false; }

#page_size;
#table; #head; #insertBody; #displayBody;
#insertRow; #displayRow;
constructor(table, page_size, socket, socketEventName = "table_rows")
{
    this.#page_size = page_size;
    this.socket = socket;
    this.socketEventName = socketEventName;

    this.#table = table;
    this.#head = table.querySelector("thead");
    this.#insertBody = table.querySelector("tbody[data-type=insert]");
    this.#displayBody = table.querySelector("tbody[data-type=display]");

    let rowTemplate = table.querySelector("[data-type=row]"); rowTemplate.remove();

    let insertIdentityColumn = rowTemplate.content.querySelector("[data-type=insert]");
    if (insertIdentityColumn) { insertIdentityColumn.parentElement.dataset.identity =
true; insertIdentityColumn.remove(); }

    let displayIdentityColumn =
rowTemplate.content.querySelector("template[data-type=display]");

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

    if (displayIdentityColumn) { displayIdentityColumn.parentElement.dataset.identity =
true; displayIdentityColumn.remove(); }

```

```

    let insertTemplate = rowTemplate.cloneNode(true), displayTemplate =
rowTemplate.cloneNode(true);

    if (insertIdentityColumn)
insertTemplate.content.querySelector("[data-identity]").append(insertIdentityColumn.conte
nt.cloneNode(true));

    if (displayIdentityColumn)
displayTemplate.content.querySelector("[data-identity]").append(displayIdentityColumn.co
ntent.cloneNode(true));

```

```

this.#insertRow = class InsertRow extends TableInsertRow

```

```

{
    constructor() { super(insertTemplate); }
};

```

```

this.#displayRow = class DisplayRow extends TableDisplayRow

```

```

{
    constructor(data) { super(displayTemplate, data); }
};

```

```

this.#setupHead();

```

```

this.#setupInsert();

```

```

this.#setupDisplay();

```

```

Table.#tables.push(this);

```

```

}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

#setupHead()
{
  Array.from(this.#head.querySelectorAll("input, select, textarea")).forEach(input =>
  {
    input.addEventListener("change", this.#setupDisplay.bind(this), { "capture": false,
    "once": false, "passive": true });
  });
}
#getFilters()
{
  return Array.from(this.#head.querySelectorAll(":is(input, select,
  textarea):not(.sortOrder)"))
    .filter(input => input.value)
    .map(input => ({ name: input.dataset.column, value: input.value, comparison:
    input.dataset.comparison }));
}
#getSorts()
{
  let sorts = Array.from(this.#head.querySelectorAll(".sortOrder")).filter(select =>
  select.value !== "no").map(select => ({ name: select.dataset.column, order: select.value
  }));
  if (sorts.length == 0)
  {
    sorts = Array.from(this.#head.children[0].children)
      .filter(th => (th.children?.[0]?.children?.[1]?.innerText ?? "").includes("[PK]"))
      .map(th => ({ name: th.children[0].children[0].innerText, order: 'asc' }));
  }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

if (sorts.length == 0)
{
    sorts = Array.from(this.#head.children[0].children)
        .filter(th => (th.children?.[0]?.children?.[1]?.innerText ?? "").includes("[U]"))
        .map(th => ({ name: th.children[0].children[0].innerText, order: 'asc' }));
}
return sorts;
}

#insertRows = [ ];
#addInsertRowCallback = this.#addInsertRow.bind(this);
#addInsertRow()
{
    if (this.#insertRows.length != 0)
    {
        this.#insertRows.at(-1).removeChangeCallback(this.#addInsertRowCallback);
        this.#insertRows.at(-1).addRemoveCallback(this.#removeInsertRow.bind(this));
        this.#insertRows.at(-1).showDeleteButton();
    }

    this.#insertRows.push(new this.#insertRow());
    this.#insertRows.at(-1).addChangeCallback(this.#addInsertRowCallback);
    this.#insertBody.append(...this.#insertRows.at(-1).elements);
}

#removeInsertRow(row) { row.elements.forEach(row => row.remove());
this.#insertRows = this.#insertRows.filter(r => r != row); }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

#setupInsert() { if (this.#insertBody) this.#addInsertRow(); }

#displayRows = [];
#displayObserver;
#getNextPageData()
{
  return new Promise(resolve =>
  {
    const socketListener = (message) => {
      const msg = JSON.parse(message.data);
      if (msg.eventName === this.socketEventName)
      {
        resolve(msg.data);
        this.socket.removeEventListener("message", socketListener, { "capture": false,
"once": false, "passive": true });
      }
    };
    this.socket.addEventListener("message", socketListener, { "capture": false, "once":
false, "passive": true });
    this.socket.send(JSON.stringify({
      requestName: this.socketEventName,
      data: {
        tableid: Number(this.#table.dataset.tableid),
        limit: this.#page_size,
        filters: this.#getFilters(),
        sorts: this.#getSorts()

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
  }));
});
}
#renderData(data)
{
  for (const row of data)
  {
    this.#displayRows.push(new this.#displayRow(row));
    const identityCell = this.#displayRows.at(-1).elements.map(e =>
e.querySelector("[data-identity]")).filter(e => e)[0];
    if (identityCell) identityCell.children[0].children[1].innerText =
this.#displayRows.length;
    this.#displayBody.append(...this.#displayRows.at(-1).elements);
  }
}
async #loadNextPage()
{
  this.#displayObserver.disconnect();
  const data = await this.#getNextPageData();
  this.#renderData(data);
  if (this.#displayBody.lastElementChild && data.length === this.#page_size)
this.#displayObserver.observe(this.#displayBody.lastElementChild);
}
#setupDisplay()
{
  if (this.#displayObserver) this.#displayObserver.disconnect();

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

else this.#displayObserver = new IntersectionObserver(entries => { if
(entries[0].isIntersecting) this.#loadNextPage() }, { rootMargin: "0px", threshold: 0 });
Array.from(this.#displayBody.children).forEach(row => row.remove());
this.#displayRows = [ ];
this.#loadNextPage();
}

getUpdateActions()
{
return [
...this.#insertRows.slice(0, -1).map(row => ({ type: "INSERT", data: row.getData()
})),
...this.#displayRows.filter(row => row.deleted).map(row => ({ type: "DELETE", id:
row.getIdentifier(this.#head) })),
...this.#displayRows.filter(row => row.edited && !row.deleted).map(row => ({
type: "UPDATE", data: row.getData(), id: row.getIdentifier(this.#head) }))
];
}
};

```

#### 4.3.2.2. Директория admin

##### 4.3.2.2.1. Файл logs.js

```

import Table from "/static/js/utis/table.js";

var config = undefined, table = undefined;

const socket = new WebSocket(`wss://${location.host}/data`);

socket.addEventListener("message", message =>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
  const msg = JSON.parse(message.data);
  switch (msg.eventName)
  {
    case "config":
    {
      config = msg.data;
      table = new Table(document.getElementById("logs_table"), config.page_size,
socket, "logs");
      break;
    }
    case "logs": { break; }
    default:
    {
      console.warn("Websocket - unknown eventName received " + msg.eventName);
    }
  }
}, { "capture": false, "once": false, "passive": true });

```

#### 4.3.2.3. Файл database.js

```
import Chart from "/static/js/utils/chart.js";
```

```
var config = undefined, charts = undefined, socket = new
```

```
WebSocket(`wss://${location.host}/data`);
```

```
function requestData()
```

```
{
```

```
  socket.send(JSON.stringify({ requestName: "database_data" }));
```

```
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

socket.addEventListener("message", message =>
{
  const msg = JSON.parse(message.data);
  switch (msg.eventName)
  {
    case "config":
    {
      config = msg.data;
      charts = {
        connections: new Chart( document.getElementById("connections_graph"),
                                config.graph_records,
                                config.update_interval,
                                "Соединения с базой данных",
                                "Количество соединений",
                                { active: "Активные", idle: "Бездействующие" }),
        queries: new Chart( document.getElementById("queries_graph"),
                            config.graph_records,
                            config.update_interval,
                            "Запросы к базе данных",
                            "Количество запросов",
                            { success: "Успешные", cancelled: "Отменённые" }),
        reads: new Chart( document.getElementById("reads_graph"),
                           config.graph_records,
                           config.update_interval,
                           "Статистика получения записей",
                           "Количество записей",
                           { total: "Всего", index: "По индексу" }),
      }
    }
  }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

updates: new Chart( document.getElementById("updates_graph"),
    config.graph_records,
    config.update_interval,
    "Изменения записей базы данных",
    "Количество записей",
    { inserted: "Добавленных", updated: "Обновлённых", deleted:
"Удалённых" })),

blocks: new Chart( document.getElementById("blocks_graph"),
    config.graph_records,
    config.update_interval,
    "Количество обращений к диску",
    "Количество обращений",
    { total: "Всего", cache: "Кэш" })),

worktime: new Chart(document.getElementById("worktime_graph"),
    config.graph_records,
    config.update_interval,
    "Время исполнения запросов базой данных",
    "Время (ms)",
    { worktime: "Время исполнения запросов" })

};

requestData();

break;
}

case "database_data":
{
    document.getElementById("database_name").innerText = msg.data.database_name;
    document.getElementById("postgres_version").innerText = msg.data.postgres_version;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

document.getElementById("uptime").innerText = msg.data.uptime;
document.getElementById("database_size").innerText = msg.data.database_size;
document.getElementById("stats_reset").innerText = msg.data.reset || "Никогда";
for (const chartName in charts) charts[chartName].addRecord(msg.data[chartName]);
setTimeout(requestData, config.update_interval);

break;
}
default:
{
  console.warn("Websocket - unknown eventName received " + msg.eventName);
}
}
}, { "capture": false, "once": false, "passive": true });

```

#### 4.3.2.4. Файл table.js

```

import Chart from "/static/js/Utils/chart.js";
import Table from "/static/js/Utils/table.js";

Table.registerSaveButton(document.getElementById("save_changes"));

var config = undefined, charts = undefined, table = undefined, socket = new
WebSocket(`wss://${location.host}/data`);

function requestData()
{
  socket.send(JSON.stringify({ requestName: "table_data", data: { id: Number(new
URLSearchParams(location.search).get("id")) } }));
}

socket.addEventListener("message", message =>
{

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

const msg = JSON.parse(message.data);
switch (msg.eventName)
{
  case "config":
  {
    config = msg.data;
    charts = {
      reads: new Chart( document.getElementById("reads_graph"),
        config.graph_records,
        config.update_interval,
        "Статистика получения записей таблицы",
        "Количество записей",
        { total: "Всего", index: "По индексу" } ),
      scans: new Chart( document.getElementById("scans_graph"),
        config.graph_records,
        config.update_interval,
        "Количество поисков по таблице",
        "Количество сканирований",
        { sequential: "Последовательные сканирования", index: "По индексу" } ),
      updates: new Chart( document.getElementById("updates_graph"),
        config.graph_records,
        config.update_interval,
        "Изменения записей таблицы",
        "Количество записей",
        { inserted: "Добавленных", updated: "Обновлённых", deleted:
"Удалённых" } ),
      live: new Chart(document.getElementById("live_graph"),

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        config.graph_records,
        config.update_interval,
        'Количество "живых" и "мёртвых" записей таблицы',
        "Количество записей",
        { live: "Живых", dead: "Мёртвых" })
    };
    table = new Table(document.getElementById("data_table"), config.page_size, socket);
    requestData();
    break;
}
case "table_data":
{
    document.getElementById("table_name").innerText = msg.data.table.name;
    document.getElementById("table_type").innerText = msg.data.table.type;
    document.getElementById("table_size").innerText = msg.data.table.size;
    document.getElementById("table_rows").innerText = msg.data.table.rows;
    for (const chartName in charts) charts[chartName].addRecord(msg.data[chartName]);
    setTimeout(requestData, config.update_interval);
    break;
}
case "table_rows": { break; }
default:
{
    console.warn("Websocket - unknown eventName received " + msg.eventName);
}
}
}, { "capture": false, "once": false, "passive": true });

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

#### 4.4. Директория views

##### 4.4.1. Директория general

###### 4.4.1.1. Файл head.ejs

```

<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Inter:wght@100..900&display=swap"
nonce="<%=res.cspNonce.style%>">

<link rel="stylesheet" href="/static/css/general.css" nonce="<%=res.cspNonce.style%>">
<script src="/static/js/utills/general.js" type="module"
nonce="<%=res.cspNonce.script%>"></script>

<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="canonical" href="https://<%=config.website.host%><%=req.routerPath%>">

<link rel="icon" type="image/svg+xml" href="/static/images/favicon/favicon.svg" sizes="any">
<link rel="icon" type="image/x-icon" href="/static/images/favicon/favicon.ico" sizes="32x32">
<link rel="icon" type="image/png" href="/static/images/favicon/favicon_48x48.png"
sizes="48x48">
<link rel="icon" type="image/png" href="/static/images/favicon/favicon_96x96.png"
sizes="96x96">
<link rel="icon" type="image/png" href="/static/images/favicon/favicon_144x144.png"
sizes="144x144">

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
<link rel="icon" type="image/png" href="/static/images/favicon/favicon_192x192.png"
sizes="192x192">
```

```
<link rel="apple-touch-icon" type="image/png"
href="/static/images/favicon/apple-touch-icon.png" sizes="180x180">
```

```
<meta name="supported-color-schemes" content="dark">
```

#### 4.4.1.2. Файл header.ejs

```
<header class="surface">
```

```
<div class="siteinfo flex_horizontal">
  <% if (locals.has_nav)
  { %>
    <button class="striped_button navigation_opener desktop_hidden" title="Развернуть
меню навигации">
      <span></span><span></span><span></span>
    </button>
  <% } %>
  <div class="userinfo text_nowrap">
    <div class="text_size_medium"><%=req.authorization.login%></div>
    <%
    const perms_title = ((req.authorization.permissions.includes("R") ? "Чтение&#10;" : "") +
      (req.authorization.permissions.includes("I") ? "Вставка&#10;" : "") +
      (req.authorization.permissions.includes("U") ? "Обновление&#10;" : "") +
      (req.authorization.permissions.includes("D") ? "Удаление&#10;" : "") +
      (req.authorization.permissions.includes("A") ? "Администрирование&#10;" :
      "")) || "Отсутствует";
    %>
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    <div class="text_size_small" title="<%=perms_title%>"> <span
class="text_minor">Доступ:</span> <span
class="text_brand"><%=req.authorization.permissions%></span> </div>

```

```

    </div>

```

```

</div>

```

```

<div class="navigation"></div>

```

```

<div class="grid_1x2">

```

```

    <% if (req.routerPath == "/admin/accounts") { %> <button class="padding_regular_wide"
data-dialog="create_account">Создать</button> <% }

```

```

    else { %> <button id="save_changes" class="padding_regular_wide" form="changes"
hidden>Сохранить</button> <% } %>

```

```

    <a href="/logout" class="control_primary padding_regular_wide">Выйти</a>

```

```

    </div>

```

```

</header>

```

```

<% if (req.routerPath.startsWith("/table"))

```

```

{ %>

```

```

    <form id="changes" method="POST" action="/data" hidden>

```

```

        <input type="hidden" name="actions" value="[]">

```

```

        <input type="hidden" name="table" value="<%=req.query.id%>">

```

```

        <input type="hidden" name="authentication" value="<%=req.authentication%>">

```

```

    </form>

```

```

<% } %>

```

#### 4.4.1.3. Файл input.ejs

```

<%

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

let attributes = `name="\${column.name}" data-column="\${column.name}" autocomplete="off" `;

if (column.type.id === 16 || column.enumlabels) attributes += `data-comparison="exact" `;
else attributes += `data-comparison="substring" `;

if (!req.authorization.permissions.includes("U") || (identityType === 'body' && (column.flags?.ia ||
column.flags?.id || req.routerPath === "/admin/logs"))) attributes += "disabled ";

switch (column.type.id)
{
  case 16:
  {
    if (identityType === 'head')
    { %>
      <select <%-attributes%>>
        <option value="">Не выбрано</option>
        <option value="false">Не отмечено</option>
        <option value="true">Отмечено</option>
      </select>
    <% }
    else
    { %>
      <label class="checkboxWrapper"> <input type="checkbox" <%-attributes%>> </label>
    <% }
    break;
  }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

case 20: { %> <input type="number" placeholder="Введите число" step="1"
min="-9223372036854775807" max="9223372036854775807" <%-attributes%>> <% break; }

case 21: { %> <input type="number" placeholder="Введите число" step="1" min="-32767 "
max="32767 " <%-attributes%>> <% break; }

case 23: { %> <input type="number" placeholder="Введите число" step="1"
min="-2147483648" max="2147483648" <%-attributes%>> <% break; }

case 700: case 701: case 1700: { %> <input type="number" placeholder="Введите число"
<%-attributes%>> <% break; }

case 1114: case 1184: { %> <input type="datetime-local" placeholder="Введите дату и время"
<%-attributes%>> <% break; }

case 1082: { %> <input type="date" placeholder="Введите дату" <%-attributes%>> <% break;
}

case 1083: case 1266: { %> <input type="time" placeholder="Введите время"
<%-attributes%>> <% break; }

case 18: case 1043: { %> <input type="text" placeholder="Введите данные"
maxlength="<%=column.type.mod - 4%>" <%-attributes%>> <% break; }

default:
{
if (column.enumlabels)
{ %>
<select <%-attributes%>>
<option value="">Не выбрано</option>
<% for (const key of column.enumlabels) { %> <option
value="<%=key%>"><%=key%></option> <% } %>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

</select>

<% }

else

{ %>

    <textarea placeholder="Введите данные" rows="1" <%-attributes%>></textarea>

<% }

break;

}

} %>

```

#### 4.4.1.4. Файл nav.ejs

```

<nav class="surface closed" id="navigation">

    <h2 class="block_minheight">Навигация</h2>

    <menu class="no_bullets margin_vertical_enormous">

        <li class="desktop_hidden"><button class="control_secondary maxwidth text_align_left
navigation_closer" href="#"><span>Закрыть меню</span></div></li>

        <li class="desktop_hidden"><hr></li>

        <% if (req.authorization.permissions.includes("A"))

        { %>

            <li>

                <button class="<%=req.routerPath.startsWith('/admin') ? " : 'control_block' %>
maxwidth text_align_left dropdown_opener dropdown_arrow dropdown_opened"
data-dropdown="nav_admin_dropdown"><span>Администрирование</span></button>

                <menu id="nav_admin_dropdown" class="no_bullets dropdown_inplace"
tabindex="-1">

                    <% if (req.routerPath == "/admin/accounts") { %> <li><div class="control_secondary
maxwidth"><span>Аккаунты</span></div></li> <% }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

else { %> <li><a class="control_block maxwidth"
href="/admin/accounts"><span>Аккаунты</span></a></li> <% }

if (req.routerPath == "/admin/logs") { %> <li><div class="control_secondary
maxwidth"><span>История</span></div></li> <% }

else { %> <li><a class="control_block maxwidth"
href="/admin/logs"><span>История</span></a></li> <% } %>

</menu>

</li>

<% } %>

<% if (req.routerPath == "/database") { %> <li><div class="control_primary
maxwidth"><span>База данных: <%=database_name%></span></div></li> <% }

else { %> <li><a class="control_block maxwidth" href="/database"><span>База данных:
<%=database_name%></span></a></li> <% } %>

<li>

<button class="<%=req.routerPath.startsWith('/table') ? " : 'control_block' %> maxwidth
text_align_left dropdown_opener dropdown_arrow dropdown_opened"
data-dropdown="nav_tables_dropdown"><span>Таблицы</span></button>

<menu id="nav_tables_dropdown" class="no_bullets dropdown_inplace" tabindex="-1">

<% for (const table of tables)

{

if (req.query.id == table.id) { %> <li><div class="control_secondary
maxwidth"><span><%=table.name%></span></div></li> <% }

else { %> <li><a class="control_block maxwidth"
href="/table?id=<%=table.id%>"><span><%=table.name%></span></a></li> <% }

} %>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
</menu>
```

```
</li>
```

```
</menu>
```

```
</nav>
```

```
<div id="navigation_backdrop" class="closed desktop_hidden"></div>
```

#### 4.4.2. Файл error.ejs

```
<!DOCTYPE html>
```

```
<html lang='ru'>
```

```
<head>
```

```
<meta name="robots" content="noindex">
```

```
<title><%=error%> <%=title%> - DBViewer</title>
```

```
<meta name="description" content="">
```

```
<style nonce="<%=res.cspNonce.style%>">
```

```
body { display: flex; align-items: center; justify-content: center; text-align: center; }
```

```
main { max-width: min(80ch, 90vw); }
```

```
</style>
```

```
<%- include("./general/head.ejs", locals) %>
```

```
</head>
```

```
<body>
```

```
<main tabindex="-1">
```

```
<h1 class="text_size_enormous"><%=error%> <%=title%></h1>
```

```
<p>
```

```
<% if (Array.isArray(description))
```

```
{
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    for (const line of description) { %> <%=line%><br> <% }
  }
  else
  { %>
    <%=description%>
    <% } %>
  }
</p>
<a href="/" class="control_primary padding_regular_wide margin_vertical_regular">На
главную</a>
</main>
</body>
</html>

```

#### 4.4.3. Директория admin

##### 4.4.3.1. Файл accounts.ejs

```

<% locals.has_nav = true; %>
<!DOCTYPE html>
<html lang='ru'>
  <head>
    <title>Страница управления аккаунтами</title>
    <meta name="description" content="Страница управления аккаунтами">
    <%- include("../general/head.ejs", locals) %>
  </head>
  <body>
    <%- include("../general/nav.ejs", locals) %>

    <div class="main_body">
      <%- include("../general/header.ejs", locals) %>
    </div>
  </body>
</html>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
<main tabindex="-1">
```

```
<h1>Страница управления аккаунтами</h1>
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th>Имя аккаунта</th>
```

```
<th>Уровень доступа</th>
```

```
<th>Управление</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<% for (const account of accounts)
```

```
{ %>
```

```
<tr>
```

```
<td><%=account.login%></td>
```

```
<%
```

```
const perms_title = ((account.permissions.includes("R") ? "Чтение&#10;" : "")
```

```
+
```

```
(account.permissions.includes("I") ? "Вставка&#10;" : "") +
```

```
(account.permissions.includes("U") ? "Обновление&#10;" : "") +
```

```
(account.permissions.includes("D") ? "Удаление&#10;" : "") +
```

```
(account.permissions.includes("A") ?
```

```
"Администрирование&#10;" : "")) || "Отсутствует";
```

```
%>
```

```
<td title="<%-perms_title%>"><%=account.permissions%></td>
```

```
<td>
```

```
<div class="grid_1x2">
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<button
data-dialog="edit_account_<%=account.id%>">Редактировать</button>

<form method="POST" action="/admin/accounts/delete">

  <input type="hidden" name="authentication"
value="<%=req.authentication%>">

  <input type="hidden" name="id" value="<%=account.id%>">

  <button>Удалить</button>

</form>

</div>

<dialog id="edit_account_<%=account.id%>" tabindex="-1">

  <form method="POST" action="/admin/accounts/edit" class="grid
grid_columns2">

    <h2>Редактирование аккаунта</h2>

    <label>

      Имя аккаунта:

      <input type="text" name="login" value="<%=account.login%>"
placeholder="Введите не более 100 символов" maxlength="100" required>

    </label>

    <label>

      Пароль:

      <input type="password" name="password" placeholder="Пароль"
maxlength="100">

    </label>

    <label class="grid_fill_row">

      Уровень доступа:

      <select multiple name="permissions">

        <option value="R" <%=account.permissions.includes("R") ?
'selected' : "%>>Чтение</option>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        <option value="I" <%=account.permissions.includes("I") ?
'selected' : "%>>Вставка</option>

        <option value="U" <%=account.permissions.includes("U") ?
'selected' : "%>>Обновление</option>

        <option value="D" <%=account.permissions.includes("D") ?
'selected' : "%>>Удаление</option>

        <option value="A" <%=account.permissions.includes("A") ?
'selected' : "%>>Администрирование</option>

    </select>

</label>

<button type="button"
data-dialog="edit_account_<%=account.id%>">Отменить</button>

<button>Сохранить</button>

<input type="hidden" name="authentication"
value="<%=req.authentication%>">

    <input type="hidden" name="id" value="<%=account.id%>">

</form>

</dialog>

</td>

</tr>

<% } %>

</tbody>

</table>

</main>

</div>

```

```

<dialog id="create_account" tabindex="-1">

```

```

    <form method="POST" action="/admin/accounts/create" class="grid grid_columns2">

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

<h2>Создание аккаунта</h2>

<label>

    Имя аккаунта:

    <input type="text" name="login" placeholder="Введите не более 100 символов"
maxlength="100" required>

</label>

<label>

    Пароль:

    <input type="password" name="password" placeholder="Пароль" required>

</label>

<label class="grid_fill_row">

    Уровень доступа:

    <select multiple name="permissions">

        <option value="R">Чтение</option>

        <option value="I">Вставка</option>

        <option value="U">Обновление</option>

        <option value="D">Удаление</option>

        <option value="A">Администрирование</option>

    </select>

</label>

<button type="button" data-dialog="create_account">Отменить</button>

<button>Сохранить</button>

<input type="hidden" name="authentication" value="<%=req.authentication%>">

</form>

</dialog>

</body>

</html>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**4.4.3.2. Файл logs.ejs**

```
<% locals.has_nav = true; %>
```

```
<!DOCTYPE html>
```

```
<html lang='ru'>
```

```
<head>
```

```
<title>История запросов</title>
```

```
<meta name="description" content="История запросов">
```

```
<script src="/static/js/admin/logs.js" type="module"
```

```
nonce="<%=res.cspNonce.script%>"></script>
```

```
<%- include("../general/head.ejs", locals) %>
```

```
</head>
```

```
<body>
```

```
<%- include("../general/nav.ejs", locals) %>
```

```
<div class="main_body">
```

```
<%- include("../general/header.ejs", locals) %>
```

```
<main tabindex="-1">
```

```
<h1>История запросов</h1>
```

```
<table id="logs_table" class="data_table" data-tableid="<%=req.query.id%>">
```

```
<thead>
```

```
<tr>
```

```
<% for (const column of columns)
```

```
{ %>
```

```
<th> <button class="control_block text_size_regular dropdown_opener
```

```
dropdown_arrow"
```

```
data-dropdown="data_table_head_<%=column.name%>"><span><%=column.label ||
```

```
column.name%></span></button> </th>
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        <% } %>
    </tr>
    <tr>
        <% for (const column of columns)
        { %>
            <td>
                <div id="data_table_head_<%=column.name%>" class="dropdown_inplace
flex_vertical" tabindex="-1" hidden>
                    <label> Фильтрация: <%- include("../general/input.ejs", { ...locals,
column, identityType: 'head' }) %> </label>
                    <label>
                        Сортировка:
                        <select class="sortOrder" data-column="<%=column.name%>">
                            <option value="asc">По возрастанию</option>
                            <option value="desc">По убыванию</option>
                            <option value="no" selected>Без сортировки</option>
                        </select>
                    </label>
                </div>
            </td>
        <% } %>
    </tr>
</thead>
<tbody data-type="display"></tbody>
<template data-type="row">
    <tr>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        <% for (const column of columns) { %> <td> <%- include("../general/input.ejs",
{ ...locals, column, identityType: 'body' }) %> </td> <% } %>

    </tr>

</template>

</table>

</main>

</div>

</body>

</html>

```

#### 4.4.4. Файл database.ejs

```

<% locals.has_nav = true; %>

<!DOCTYPE html>

<html lang='ru'>

  <head>

    <script src="https://www.gstatic.com/charts/loader.js" nonce="<%=res.cspNonce.script%>"
defer></script>

    <title><%=database_name%></title>

    <meta name="description" content="База данных <%=database_name%>">

    <script src="/static/js/database.js" type="module" nonce="<%=res.cspNonce.script%>"></script>

    <%- include("../general/head.ejs", locals) %>

  </head>

  <body>

    <%- include("../general/nav.ejs", locals) %>

    <div class="main_body">

      <%- include("../general/header.ejs", locals) %>

      <main tabindex="-1">

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

<h1>База данных <span id="database\_name"><%=database\_name%></span></h1>

<dl class="margin\_vertical\_regular">

<dt>Версия PostgreSQL</dt>

<dd id="postgres\_version">&nbsp;</dd>

<dt>Время работы</dt>

<dd id="uptime">&nbsp;</dd>

<dt>Объём памяти</dt>

<dd id="database\_size">&nbsp;</dd>

<dt>Последний сброс статистики</dt>

<dd id="stats\_reset">&nbsp;</dd>

</dl>

<p class="margin\_vertical\_regular">Ниже представлена статистика нагрузки на сервер базы данных, которая может быть полезна при диагностике ошибок БД администратором:</p>

<div class="graphs">

<div class="surface\_rounded" id="connections\_graph"></div>

<div class="surface\_rounded" id="queries\_graph"></div>

<div class="surface\_rounded" id="reads\_graph"></div>

<div class="surface\_rounded" id="updates\_graph"></div>

<div class="surface\_rounded" id="blocks\_graph"></div>

<div class="surface\_rounded" id="worktime\_graph"></div>

</div>

</main>

</div>

</body>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
</html>
```

#### 4.4.5. Файл oauth.ejs

```
<!DOCTYPE html>
```

```
<html lang='ru'>
```

```
<head>
```

```
<title>Авторизация</title>
```

```
<meta name="description" content="Авторизация">
```

```
<style nonce="<%=res.cspNonce.style%>">
```

```
body { display: flex; align-items: center; justify-content: center; }
```

```
input { min-width: 20vw; }
```

```
</style>
```

```
<%- include("./general/head.ejs", locals) %>
```

```
</head>
```

```
<body>
```

```
<main class="surface_rounded" tabindex="-1">
```

```
<h1>Авторизация</h1>
```

```
<form action="/oauth" method="post">
```

```
<label for="login">
```

Логин

```
<input type="text" placeholder="Введите логин..." name="login" id="login"
```

```
maxlength="100">
```

```
</label>
```

```
<label for="password">
```

Пароль

```
<input type="password" placeholder="Введите пароль..." name="password"
```

```
id="password" maxlength="100">
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
</label>
```

```
<button>Авторизоваться</button>
```

```
<input type="hidden" name="authentication" value="<%=req.authentication%>">
```

```
</form>
```

```
</main>
```

```
</body>
```

```
</html>
```

#### 4.4.6. Файл table.ejs

```
<% locals.has_nav = true; %>
```

```
<!DOCTYPE html>
```

```
<html lang='ru'>
```

```
<head>
```

```
<script src="https://www.gstatic.com/charts/loader.js" nonce="<%=res.cspNonce.script%>"
defer></script>
```

```
<title><%=tables.find(table => table.id == req.query.id).name%></title>
```

```
<meta name="description" content="Таблица <%=tables.find(table => table.id ==
req.query.id).name%>">
```

```
<script src="/static/js/table.js" type="module" nonce="<%=res.cspNonce.script%>"></script>
```

```
<%- include("./general/head.ejs", locals) %>
```

```
</head>
```

```
<body>
```

```
<%- include("./general/nav.ejs", locals) %>
```

```
<div class="main_body">
```

```
<%- include("./general/header.ejs", locals) %>
```

```
<main tabindex="-1">
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
<h1>Таблица <span id="table_name"><%=tables.find(table => table.id ==
req.query.id).name%></span></h1>
```

```
<dl class="margin_vertical_regular">
```

```
<dt>Тип</dt>
```

```
<dd id="table_type">&nbsp;</dd>
```

```
<dt>Объём памяти</dt>
```

```
<dd id="table_size">&nbsp;</dd>
```

```
<dt>Количество записей в таблице</dt>
```

```
<dd id="table_rows">&nbsp;</dd>
```

```
</dl>
```

```
<p class="margin_vertical_regular">Ниже представлена статистика нагрузки на сервер
базы данных, которая может быть полезна при диагностике ошибок БД администратором:</p>
```

```
<div class="graphs">
```

```
<div class="surface_rounded" id="reads_graph"></div>
```

```
<div class="surface_rounded" id="scans_graph"></div>
```

```
<div class="surface_rounded" id="updates_graph"></div>
```

```
<div class="surface_rounded" id="live_graph"></div>
```

```
</div>
```

```
<table id="data_table" class="data_table" data-tableid="<%=req.query.id%>">
```

```
<thead>
```

```
<tr>
```

```
<th class="text_align_center" data-identity>№</th>
```

```
<%= const flag_tips = {
```

```
  "IA": "Генерируемый всегда идентификатор",
```

```
  "ID": "Стандартно генерируемый идентификатор",
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



"NN": "Не содержит пропущенных значений",

"D": "Имеет стандартное значение",

"G": "Генерируемое значение",

"U": "Уникальное значение",

"PK": "Ключ"

};

for (const column of columns)

{ %>

<th>

<% const title = Object.entries(column.flags).filter(flag => flag[1]).map(flag => flag\_tips[flag[0].toUpperCase()]).join(' &#10;'); %>

<button class="control\_block text\_size\_regular dropdown\_opener dropdown\_arrow" data-dropdown="data\_table\_head\_<%=column.name%>" title="<%=title%>">

<span><%=column.name%></span>

<span class="text\_minor text\_size\_small">

<%=Object.entries(column.flags).filter(flag => flag[1]).map(flag => `[\$ {flag[0].toUpperCase()}]` ).join(' ')%>

<%=column.type.name%>

</span>

</button>

</th>

<% } %>

</tr>

<tr>

<td></td>

<% for (const column of columns)

{ %>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<td>

    <div id="data_table_head_<%=column.name%>" class="dropdown_inplace
flex_vertical" tabindex="-1" hidden>

        <label> Фильтрация: <%- include("./general/input.ejs", { ...locals, column,
identityType: 'head' }) %> </label>

        <label>

            Сортировка:

            <select class="sortOrder" data-column="<%=column.name%>">

                <option value="asc">По возрастанию</option>

                <option value="desc">По убыванию</option>

                <option value="no" selected>Без сортировки</option>

            </select>

        </label>

    </div>

</td>

<%= } %>

</tr>

</thead>

<%= if (req.authorization.permissions.includes("I")) { %> <tbody
data-type="insert"></tbody> <%= } %>

<tbody data-type="display"></tbody>

<template data-type="row">

    <tr>

        <td>

            <template data-type="insert">

                <div>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        <button class="control_block striped_button" title="Удалить строку"
hidden><span></span></button>

        <span>Новый</span>

    </div>

</template>

<template data-type="display">

    <div>

        <% if (!req.authorization.permissions.includes("D")) { %> <button
class="control_block striped_button" title="Удалить строку" disabled><span></span></button> <% }

        else { %> <button class="control_block striped_button" title="Удалить
строку"><span></span></button> <% } %>

        <span>1</span>

    </div>

</template>

</td>

    <% for (const column of columns) { %> <td> <%- include("../general/input.ejs", {
...locals, column, identityType: 'body' }) %> </td> <% } %>

</tr>

</template>

</table>

</main>

</div>

</body>

</html>

```

#### 4.5. Файл index.js

```

import config from 'common/configs/config.js';

import fs from 'fs';

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

import fastify from 'fastify';
import ajv_formats from 'ajv-formats';

const app = fastify({
  http2: true,
  https:
  {
    allowHTTP1: true,
    key: fs.readFileSync(config.website.ssl_key, 'utf8'),
    cert: fs.readFileSync(config.website.ssl_cert, 'utf8')
  },
  forceCloseConnections: true,
  ignoreTrailingSlash: true,
  ignoreDuplicateSlashes: true,
  logger: config.stage === "testing",
  disableRequestLogging: true,
  ajv:
  {
    plugins: [ [ ajv_formats, { mode: 'full', keywords: true } ] ],
    customOptions: { removeAdditional: true, useDefaults: true, coerceTypes: "array" }
  }
});

```

```

import security from "./plugins/security.js"; app.register(security);
import other from "./plugins/other.js"; app.register(other);
import request from "./plugins/request.js"; app.register(request);
import response from "./plugins/response.js"; app.register(response);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
import utility from "./plugins/utility.js"; app.register/utility);
import oauth from "./plugins/oauth.js"; app.register/oauth);
import routes from "./plugins/routes.js"; app.register/routes);

app.listen({ "port": config.website.https_port, "host": "0.0.0.0"}, (err, address) =>
{
  if (err) throw err;
  console.info(`Server is now listening on ${address}`);
});
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) ГОСТ 19.101-77: Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77: Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77: Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78: Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78: Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78: Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.401-78: Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.602-78: Правила дублирования, учета и хранения программных документов, выполненных печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.603-78: Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) ГОСТ 19.604-78: Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Лист регистрации изменений
----------------------------

[illegible]