

## **СИСТЕМА «КЛАРИС»**

Описание технической архитектуры программного обеспечения

## СОДЕРЖАНИЕ

<b>1. АННОТАЦИЯ.....</b>	<b>3</b>
<b>2. СПИСОК СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ .....</b>	<b>4</b>
<b>3. АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....</b>	<b>5</b>
3.1 Клиентская часть.....	5
3.2 Серверная часть .....	5
3.3 Слой данных.....	6
3.4 Схема потока данных .....	6
<b>4. СТОРОНЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.....</b>	<b>7</b>

## **1. АННОТАЦИЯ**

Настоящий документ содержит краткую описание технической архитектуры программного обеспечения «Кларис», далее ПО.

Документ содержит описание внутренней структуры ПО, а также описание основных компонентов ПО и механизмов их взаимодействия.

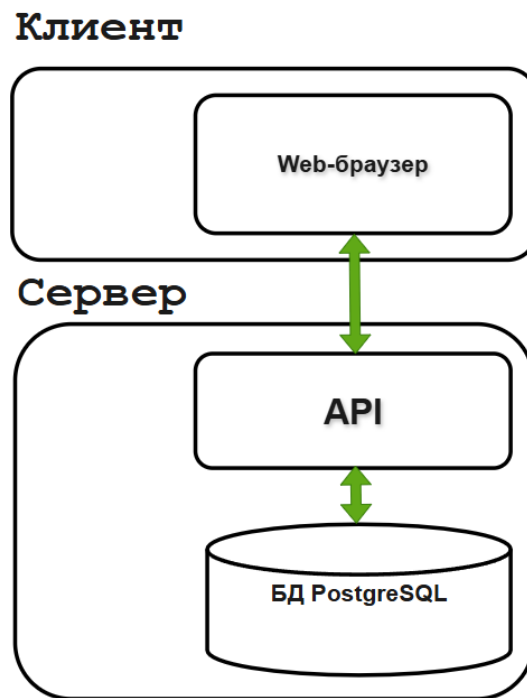
## 2. СПИСОК СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ

- **API** - Application Programming Interface, программный интерфейс приложения;
- **CSS** - язык, используемый для определения представления и макета веб-страницы;
- **HTTP** - протокол, используемый для передачи данных;
- **JavaScript** - язык программирования;
- **ПИ** - пользовательский интерфейс;
- **СУБД** - система управления базами данных;
- **ORM** - Object-Relational Mapping, библиотека для работы с базами данных;

### 3. АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ПО построено на базе классического Client-Server взаимодействия с использованием современного стека технологий.

Рис.1 Схема архитектуры ПО



#### 3.1. Клиентская часть.

- стек: React + TypeScript.
- Взаимодействие: Общение с сервером происходит через REST API (используя axios).
- Особенности: TypeScript обеспечивает надежность кода и минимизирует ошибки при работе с данными от API.

#### 3.2. Серверная часть

- стек: Node.js + Express.
- Роль: Обработка HTTP-запросов, реализация бизнес-логики, маршрутизация (Routing) и аутентификация.

- Особенности: Использование промежуточного ПО для валидации данных, логирования и обработки ошибок.

### **3.3. Слой данных**

- База данных: PostgreSQL (реляционная СУБД).
- ORM: Sequelize.
- Функции: Sequelize выступает прослойкой между кодом и БД, позволяя описывать модели данных на TypeScript. Он управляет связями (один-ко-многим, многие-ко-многим), выполняет миграции и обеспечивает безопасные запросы к базе без написания чистого SQL.

### **3.4. Схема потоков данных**

- Пользователь посредством web-браузера взаимодействует с интерфейсом React.
- React отправляет асинхронный запрос на web-сервер Express.
- Express обрабатывает запрос и обращается к моделям Sequelize.
- Sequelize выполняет запрос к PostgreSQL и возвращает объектный результат.
- Сервер отправляет JSON-ответ обратно в web-браузер.

#### 4. СТОРОНЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

При создании ПО «Кларис» использовалось программное обеспечение (пакеты) сторонних разработчиков. Перечень пакетов приведен в таблице 1.

Таблица 1.

№ п/п	Имя пакета	Лицензия	Правообладатель (Copyright holder)
1	axios	MIT	Matt Zabriskie / Axios Authors
2	bcrypt	MIT	Kelektiv (Daniel Cousins)
3	cors	MIT	Expressjs Authors
4	dotenv	BSD-2-Clause	Scott Motte
5	express	MIT	TJ Holowaychuk / StrongLoop / Expressjs Authors
6	express-fileupload	MIT	Richard Girges
7	http-server	MIT	IndexZero (Charlie Robbins)
8	jsonwebtoken	MIT	Auth0 (Okta)
9	pg	MIT	Brian Carlson
10	pg-hstore	MIT	Mick Hansen
11	query-string	MIT	Sindre Sorhus
12	react	MIT	Meta Platforms, Inc. (Facebook)
13	react-contenteditable	MIT	Bryce Kalow
14	react-dom	MIT	Meta Platforms, Inc. (Facebook)
15	react-redux	MIT	Dan Abramov / Redux Team
16	react-router	MIT	Remix Software / Shopify
17	react-router-dom	MIT	Remix Software / Shopify
18	sequelize	MIT	Sequelize Authors
19	uuid	MIT	Robert Kieffer / uuid Authors
20	yamljs	MIT	Jeremy Faivre

Лицензии MIT License и BSD 2-Clause – разрешительные открытые лицензии, не препятствующие получению исключительных прав на «Кларис».