

*Д.Д. Морозов, К.О. Маханов, И.Р. Хамитов, Т. И. Паюсова,
М.Б. Атманских*

Тюменский государственный университет, г. Тюмень

УДК 004.056

СОЗДАНИЕ БЕЗОПАСНОГО МОДУЛЯ АНАЛИЗА И ОБРАБОТКИ ДАНЫХ ПЛАТФОРМЫ «GREEN NETWORKING» НА ЯЗЫКЕ JAVASCRIPT

Аннотация. В данной статье обсуждаются инструменты и технологии, необходимые для создания безопасного модуля анализа базы данных платформы «Green Networking», приведена программная реализация процесса создания модуля. Приведена нормативно-правовая база, регулирующая сбор и анализ данных, а также обеспечение безопасности модуля.

Ключевые слова: JavaScript, Node.js, MongoDB, D3.js, Mongoose, федеральные законы, обезличивание.

Введение

Платформа «Green Networking» – единая информационно-коммуникационная платформа для выстраивания диалога между экобизнесом, экологически активными представителями общества и представителями власти. Платформа является своеобразным аналогом биржи фриланса для экологических деятелей и организаций с возможностью получения самых важных новостей экологической сферы.

Проблемы экологии вызывают все большую озабоченность различных сообществ, а мировой опыт показывает, что одним из основных способов поддержки экологических проектов выступают специальные интернет-сервисы (краудфандинг, сайты и т.п.). Однако на сегодняшний день в Тюменской области не существует сервисов для экоактивистов, которые бы содержали модуль сбора статистических данных для научных исследований.

Результаты работы позволят интегрировать модуль для сбора и анализа статистических данных об экологических проектах на платформу-агрегатор для

представителей экологических сообществ и экобизнеса, что поможет организовать изучение экологических практик Тюменского региона в рамках научно-исследовательских проектов.

Модуль анализа данных поможет выявить потребности и характеристики проектов, которые будут использованы для поиска партнеров и волонтеров. Выстраивание диалога с партнерами на платформе «Green Networking» будет способствовать масштабированию экологических практик Тюменского региона.

Данные проектов представляют коммерческую ценность и являются интеллектуальной собственностью. Потенциальные злоумышленники могут пытаться вести промышленный шпионаж, нанося тем самым финансовый и репутационный ущерб. Поэтому должна быть обеспечена безопасность разрабатываемого модуля.

Атрибуты

В соответствии с функционалом платформы, на ней могут регистрироваться как физические лица, так и организации.

У физических лиц берутся на обработку такие данные, как:

- статус самозанятости;
- информация о себе;
- адрес электронной почты;
- телефон;
- район;
- город;
- улица;
- место работы;
- перечень проектов, в которых физическое лицо принимало участие.

У организаций берется следующий перечень данных:

- наименование;
- организационная форма (ООО, ИП, НКО, другое и т. д.);
- описание;
- логотип или аватар;

- отметка о подтверждении учетной записи;
- юридический адрес организации;
- адрес, по которому ведется фактическая деятельность (район, город, район города, улица, дом, офис);
- телефон;
- адрес электронной почты;
- сайт организации (если имеется);
- ссылки на упоминание в СМИ;
- ОКВЭД и сфера деятельности.

Для анализа будут собираться такие данные о проекте как:

- название;
- описание;
- логотип или аватар;
- владелец (физическое или юридическое лицо);
- участники (физическое или юридическое лицо);
- дата публикации;
- сроки реализации;
- статус (реализованный, активный, планируемый);
- место реализации (район, город, район города, улица).
- бюджет.

Нормативно-правовая база для обеспечения безопасности

При обработке данных мы должны соблюдать нормативно-правовую базу для обеспечения безопасности, в которую входят: Федеральный закон № 152 [1] и приказ Роскомнадзора № 996 [2].

Использование персональных данных (далее – ПДн) разрешено только с согласия владельцев ПДн на их обработку. В нашем проекте осуществляется обработка ПДн, поэтому в соответствии со ст. 9 Федерального закона от 27 июля 2006 г. № 152-ФЗ «О персональных данных» физическим и юридическим лицам, которые будут являться пользователями нашей платформы, требуется предоставить согласие на обработку ПДн.

Функционал платформы предполагает обработку и анализ ПДн, а также предоставление пользователям сайта результатов статистической обработки. Согласно приказу Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор) от 5 сентября 2013 г. N 996 г. Москва "Об утверждении требований и методов по обезличиванию персональных данных", такие персональные данные подлежат обезличиванию. Для обезличивания были выбраны методы перемешивания и изменения состава.

Инструменты разработки

Для создания платформы, защищенных чатов и личных кабинетов пользователей использовался MERN-стек инструментов. Это стек для языка JavaScript, разработанный для упрощения процесса разработки, включает в себя четыре компонента с открытым исходным кодом: MongoDB [3], Express, React и Node.js [4].

Для удобного внедрения модуля анализа и обработки данных на платформу, в качестве его основы были взяты язык JavaScript, Node.js, среда выполнения JavaScript, и база данных MongoDB. Для качественной работы с MongoDB и Node.js необходима библиотека ODM Mongoose. Node.js после выполнения запроса, будет передавать информацию в модуль отрисовки графиков.

Библиотека D3.js [5] предоставляет обширный инструментарий для качественного графического представления данных, которые будут передаваться посредством запросов к базе данных и для их первичной обработки, так как при ответе от mongoDB, мы получаем коллекцию, с которой в JavaScript удобно взаимодействовать.

Программная реализация

В первую очередь модуль анализа и статистической обработки данных должен быть подключен к базе данных платформы. На рис. 1 представлен программный код, обеспечивающий соединение с базой данных.

```

async function main(){

  const uri = "mongodb+srv://<username>:<password>@<your-cluster-url>/test?retryWrites=true&w=majority";

  const client = new MongoClient(uri);

  try {
    // Соединение с кластеру MongoDB
    await client.connect();

    // делаем подходящий вызов к БД
    await listDatabases(client);
  } catch (e) {
    console.error(e);
  } finally {
    await client.close();
  }
}

main().catch(console.error);|

```

Рис. 1. Процесс соединения с базой данных с помощью Node.js

После успешного соединения с базой данных, требуется подключить библиотеку D3.js, которая позволит работать с графиками. Ее подключение показано на рисунке 2.

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/d3/6.7.0/d3.min.js"></script>

const d3 = require('d3'),
      jsdom = require('jsdom');

const document = jsdom.jsdom(),
      svg = d3.select(document.body).append( name: 'svg');|

```

Рис. 2. Подключение D3.js к модулю и добавление базовых переменных

В качестве входных данных D3.js получает массив объектов, который был взят из базы данных платформы. На рис. 3 представлена программная реализация выгрузки данных из базы данных для их дальнейшей обработки.

```

export default function define(runtime, observer) {
  const main = runtime.module();
  const fileAttachments = new Map( entries: [{"fizlica.csv", import.meta.url}]);
  main.builtin("FileAttachment", runtime.fileAttachments(name => fileAttachments.get(name)));
  main.variable(observer()).define(["md"], function(md){return(
md)

```

Рис. 3. Модуль раскрытия csv файла и его запуск в обработку

На рисунке 4 продемонстрированы алгоритмы добавления в массив выгруженных данных, а также подсчет повторяющихся значений в данном массиве.

```

main.variable(observer("data")).define("data", ["d3", "FileAttachment"], async function(d3, FileAttachment){return(
Object.assign(d3.csvParse(await FileAttachment("alphabet.csv").text(),
  ({letter, frequency}) =>
    ({name: letter, value: +frequency})).sort((a, b) =>
      d3.descending(a.value, b.value)), source: {format: "%"}
));
main.variable(observer("format")).define("format", ["x", "data"], function(x, data){return(
x.tickFormat(20, data.format)
));
main.variable(observer("x")).define("x", ["d3", "data", "margin", "width"], function(d3, data, margin, width){return(
d3.scaleLinear()
  .domain([0, d3.max(data, d => d.value)])
  .range([margin.left, width - margin.right]
));

```

Рис. 4. Реализация добавления в массив выгруженных данных и подсчет повторяющихся значений в данном массиве

На рисунке 5 продемонстрирован процесс создания макета для дальнейшей визуализации на нем результатов проведенного анализа и статистической обработки.

```

main.variable(observer("y")).define("y", ["d3", "data", "margin", "height"], function(d3, data, margin, height){return(
d3.scaleBand()
  .domain(d3.range(data.length))
  .rangeRound([margin.top, height - margin.bottom])
  .padding(0.1)
));
main.variable(observer("xAxis")).define("xAxis", ["margin", "d3", "x", "width", "data"],
function(margin, d3, x, width, data){return(
g => g
  .attr("transform", `translate(0, ${margin.top})`)
  .call(d3.axisTop(x).ticks(width / 80, data.format))
  .call(g => g.select(".domain").remove()
));
main.variable(observer("yAxis")).define("yAxis", ["margin", "d3", "y", "data"], function(margin, d3, y, data){return(
g => g
  .attr("transform", `translate(${margin.left}, 0)`)
  .call(d3.axisLeft(y).tickFormat(i => data[i].name).tickSizeOuter(0)
));
main.variable(observer("barHeight")).define("barHeight", function(){return(
25
));

```

Рис. 5. Программная реализация макета для визуализации результатов анализа

На рисунке 6 приведена программная реализация столбчатой диаграммы, на которой приведен процент самозанятых физических лиц по отношению к общему числу физических лиц, зарегистрированных на платформе.

```
main.variable(observer("height")).define("height", ["data", "barHeight", "margin"],
function(data, barHeight, margin){return(
Math.ceil( x: (data.length + 0.1) * barHeight) + margin.top + margin.bottom
)});
main.variable(observer("margin")).define("margin", function(){return(
{top: 30, right: 0, bottom: 10, left: 30}
)});
main.variable(observer("d3")).define("d3", ["require"], function(require){return(
require("d3@6")
)});
return main;
}
```

Рис. 6. Программная реализация столбчатой диаграммы с процентом самозанятых

Для наглядного отображения результатов анализа были подобраны параметры визуализации графиков. Программная реализация приведена на рисунке 7.

```
let margin = {top: 20, right: 40, bottom: 30, left: 20},
width = 960 - margin.left - margin.right,
height = 500 - margin.top - margin.bottom,
barWidth = Math.floor( x: width / 19 ) - 1 ;
let x = d3.scale.linear()
.range([barWidth / 2, width - barWidth / 2]);

let y = d3.scale.linear()
.range([height, 0]);
let yAxis = d3.svg.axis()
.scale(y)
.orient("right")
.tickSize(-width)
.tickFormat(function (d) {
return Math.round( x: d / 1e6) + "M"
});

let svg = d3.select("body").append( name: "svg")
.attr("width", width + margin.left + margin.right)
.attr("height", height + margin.top + margin.bottom)
.append( name: "g")
.attr("transform", "translate(" + margin.left + "." + margin.top + ")")
```

Рис. 7. Подбор параметров отображения графика

Результатом работы данного алгоритма является столбчатая диаграмма, приведенная на рисунке 8:

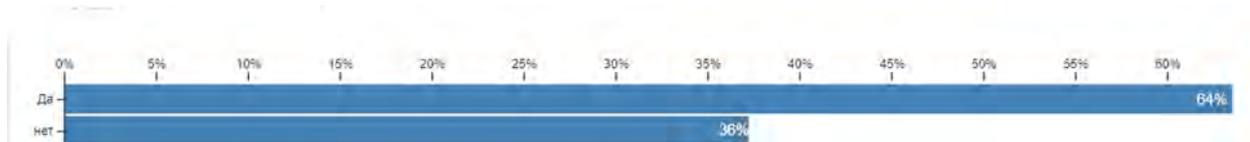


Рис. 8. Столбчатая диаграмма с процентом самозанятых физических лиц

Заключение

В ходе проделанной работы был реализован работоспособный модуль анализа базы данных платформы «Green Networking» на Node.js и MongoDB, приведена программная реализация процесса создания модуля и работы с библиотекой D3.js для визуализации графиков. Представлена нормативно-правовая база, регулирующая сбор и анализ данных, а так же обеспечение безопасности модуля.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон "О персональных данных" от 27.07.2006 № 152-ФЗ. – URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 17.05.2021).
2. Приказ Роскомнадзора № 996. – URL: http://www.consultant.ru/document/cons_doc_LAW_346719/ (дата обращения: 17.05.2021).
3. MongoDB + Node.js [Электронный ресурс]. – URL: <https://docs.mongodb.com/drivers/node/current/> (дата обращения: 17.05.2021).
4. Node.js v16.1.0 documentation [Электронный ресурс]. – URL: <https://nodejs.org/api/crypto.html> (дата обращения: 17.05.2021).
5. D3.js documentation [Электронный ресурс]. – URL: <https://github.com/d3/d3/wiki> (дата обращения: 17.05.2021).