

УДК 664.012
РЕГИСТРАЦИЯ ПРИЛОЖЕНИЙ В МНОГОПОЛЬЗОВАТЕЛЬСКОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЕ

И. В. АКИНШЕВА
Белорусско-Российский университет

В. А. ПАНТЮХОВ
Могилевский государственный университет продовольствия
Могилев, Беларусь

Целью исследования является разработка части web-приложения «community-app», предназначенного для регистрации множества web-приложений в многопользовательской системе. Назначение проекта состоит в получении информационной платформы, которая объединяла бы все созданные web-приложения внутри IT-компании на основе информационно-обучающей конкурентной среды.

Данное web-приложение предоставляет всем приложениям, зарегистрированным в нем, общую систему API, общую модель регистрации и авторизации, общую статистику по каждому приложению. В рамках «community-app» велась разработка единой системы авторизации, системы смены пароля и системы сбора статистических данных с зарегистрированных web-приложений.

Разработанное web-приложение относится к Single Page Application (SPA) приложениям. SPA – это web-приложение или web-сайт, использующий единственный HTML-документ в качестве оболочки для всех web-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, посредством технологии AJAX. SPA исполняются в рамках браузера, а не в собственном процессе операционной системы.

В разработанной многопользовательской системе регистрации и авторизации реализованы следующие этапы: разработка React компонентов-форм для реализации пользовательского ввода; создание механизма валидации пользовательского ввода со стороны «клиента»; создание механизма отправки запроса с данными пользователя на сервер и отображение состояния ожидания результата посредством добавления системы уведомлений пользователя; создание контроллера обработки запросов, направленных на регистрацию и авторизацию пользователя; создание таблицы пользователей с уникальными идентификаторами; создание механизма валидации на сервере; зашифровка пароля при регистрации и сохранение пользователя в базе данных, при подтверждении уникальности e-mail-а, и возвращение вновь созданного токена (авторизации) пользователю, который затем будет использован для обращения к защищенным API; расшифровка пароля пользователя из базы данных и возвращение

пользователю токена для обращения к защищенным API; сохранение токена пользователя в глобальном хранилище состояния web-приложения (здесь таковым является «redux-store»).

Для организации функции смены пароля, при условии, что пользователь знает свой предыдущий пароль, были реализованы следующие этапы: разработаны на основе React компоненты форм для реализации пользовательского ввода; создан механизм валидации пользовательского ввода со стороны «клиента»; создан механизм отправки запроса с данными пользователя на сервер и отображение состояния ожидания результата, посредством добавления системы уведомлений пользователя; создан контроллер обработки запроса для смены пароля пользователя; организована расшифровка пароля текущего пользователя и его сопоставление с паролем из базы данных; реализован механизм замены пароля из таблицы новым, введенным пользователем, паролем с предварительной зашифровкой и вывод уведомления об успешном изменении пароля.

При реализации механизма передачи статистических данных от зарегистрированного приложения к «community-app» необходимо было унифицировать данные, которые каждое зарегистрированное приложение будет передавать обратно в «community-app». Для этих целей была создана и опубликована библиотека под названием «community-layer», содержащая логику передачи статистических данных, представленных в единой форме.

Последняя задача состояла в том, чтобы организовать сбор статистических данных с зарегистрированного приложения. Алгоритм решения данной задачи был разделен на следующие этапы: подключение и настройка «community-layer» в зарегистрированном приложении; сохранение статистических данных в оперативной памяти на каждой итерации жизненного цикла стороннего приложения; реализация post-запроса при завершении жизненного цикла стороннего приложения с выводом результатов через «community-layer»; создание таблицы «statistic» в «community-app»; обработка полученных статистических данных и их запись в таблицу; создание таблицы компонентов со стороны «клиента»; реализация клиент-серверного запроса статистических данных с последующим уведомлением пользователя; выполнение функции сортировки по убыванию от пользователя с наибольшим количеством очков к пользователю с наименьшим количеством очков; визуальное отображение статистических данных текущего пользователя с помощью d3.js в виде трехмерного графика: где OX – дни недели, OY – процент выполнения, OZ – среднее количество очков.

Все описанные задачи были выполнены, оптимизированы и успешно протестированы, данное приложение было опубликовано в сети Интернет по адресу: <https://github.com/js-machine/community-app>. В данный момент работа над проектом ведется со стороны ИООО «ЭПАМ Системз».