

Разработка веб-приложения для доступа к данным страховых полисов

О. Ф. Ковалёв, Е. И. Миронов

Белорусский государственный университет, Минск, Беларусь,

e-mail: okovaliov@gmail.com, rct.mironov@bsu.by

В статье предложена возможная архитектура веб-приложения, позволяющего получить авторизованный доступ к базам данных страховых полисов.

Ключевые слова: страховой полис; веб-приложение; java; spring; spring secure; авторизация; клиент-серверное приложение; трёхуровневая архитектура; JSON; токен.

Development of a web-application for accessing insurance policy data

A. F. Kavaliou, Y. I. Mironau

Belarusian State University, Minsk, Belarus, e-mail: okovaliov@gmail.com, rct.mironov@bsu.by

This article suggests a possible architecture of the web application which allows to provide an authorized access to the insurance policies' databases.

Keywords: insurance policy; web application; java; spring; spring secure; authorization; client-server application; three-tier architecture; JSON; token.

Введение

В связи с недавними изменениями в законодательстве, касающимися введения единого страхового полиса для владельцев транспортных средств в Союзном государстве, возникает необходимость в разработке программного приложения, позволяющего организовать как безопасное хранение данных полисов страховых компаний, так и предоставить авторизованный доступ к ним для сотрудников ГИБДД Российской Федерации и ГАИ Республики Беларусь, а также неавторизованный доступ для обычных автовладельцев (к примеру, для проверки страхового полиса при составлении европротокола[1, 2]).

Такое веб-приложение было создано на языке программирования Java с использованием фреймворка Spring [5] для доступа к базам данных страховых полисов в режимах авторизованного и неавторизованного доступа.

1. Архитектура приложения

Ввиду наличия в базе данных персональных данных, обеспечение их безопасности принципиально необходимо. Требования безопасности, таким образом, учитывались при разработке бизнес-логики и логики данных приложения, архитектура которого содержит три уровня [3].

В рамках подобной модели существует сервер, на котором находятся файлы или базы данных, и сервер приложений. Любое обращение клиента к базе данных

проходит через промежуточный уровень с бизнес-логикой и логикой данных. Разделение данных и программ для доступа к ним, в свою очередь, позволяет повысить защиту от несанкционированного доступа.

Таким образом, обращаясь к применению данной модели в разработке, на стороне сервера баз данных собраны 3 элемента: российский и белорусский полисы, а также пользователи. Соответственно, созданы 3 репозитория для взаимодействия с базами данных и три сервиса для реализации бизнес-логики.

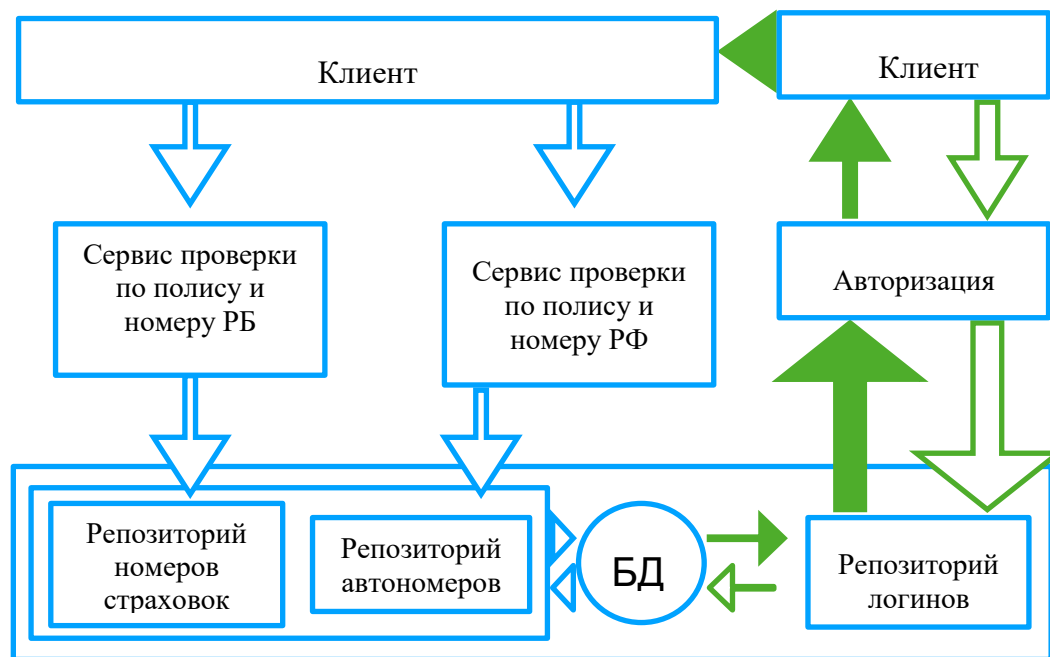


Рис. 1. Схема работы разработанного веб-приложения

2. Работа веб-сервиса

Как уже упоминалось ранее, работа веб-сервиса (рис. 1) предусматривает неавторизованный и авторизованный доступ. При авторизованном доступе пользователь должен пройти аутентификацию: он вводит логин и пароль, которые затем отправляются на сервер, который в свою очередь обращается к базе данных.

Для авторизованного доступа с помощью сервиса Flyway были созданы тестовые таблицы с данными пользователей:

```
«INSERT INTO _user (id,username, password, role) VALUES(1,'user1',
'$2a$11$HtUoIWlm7Tuq46h.xd6XcOdbi2lh83NXG5WrHVqzQTGAXIoUMkrrK',
'USER_ROLE');»
```

Пароль в базе данных храниться в хэш-виде. Сама процедура хэширование производилась с использованием BcryptPasswordEncoder с 5 итерациями хэширования для предотвращения кражи паролей.

Для организации безопасной авторизации пользователей использовался фреймворк Spring Security. При вводе верных логина и пароля, сервер выдаёт клиенту access-токен (рис. 2). Система безопасности устроена так, что при каждом запросе клиента к серверу SecurityFilter перехватывает запрос и проверяет в этом

запросе действительность токена, для которого устанавливается время жизни в 10 минут после авторизации.

При успешной аутентификации пользователь вводит номер автомобиля или номер полиса, происходит проверка и возвращается результат с персональными данными в формате JSON (рис. 2).

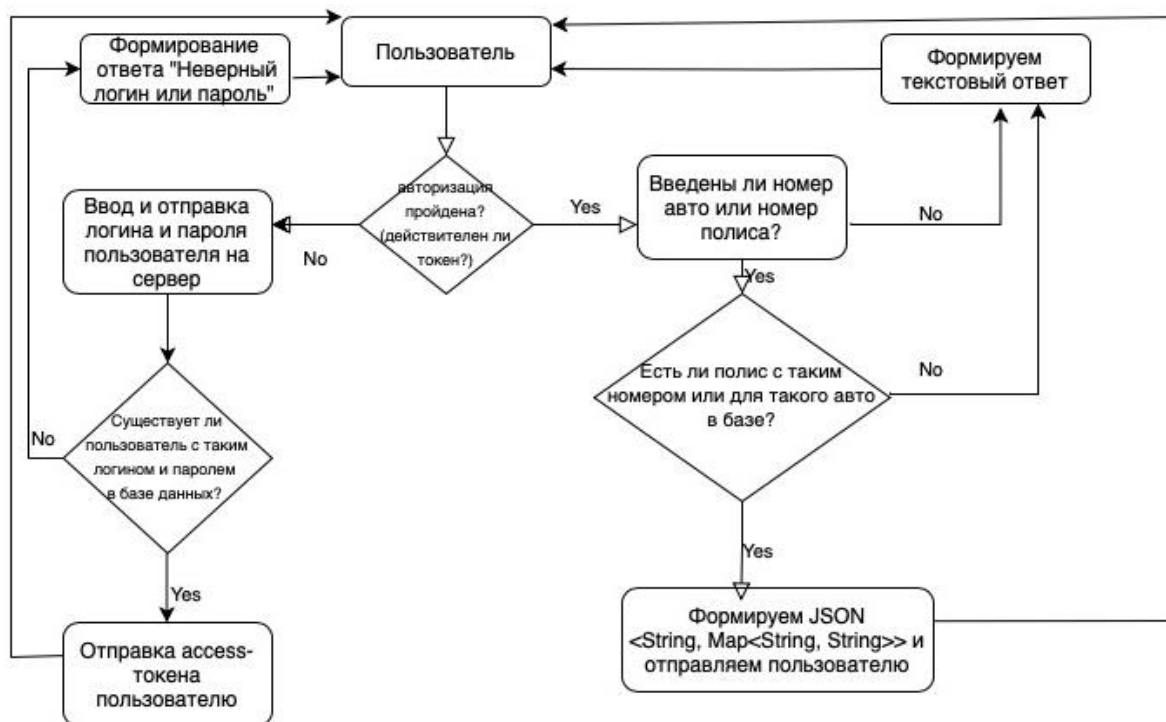


Рис. 2. Алгоритм работы веб-приложения в режиме авторизованного доступа

Следует упомянуть, что при неавторизованном доступе возвращается также JSON, но уже без персональных данных.

Для повышения удобства пользования приложением с помощью React.js был разработан Маршер для преобразования данных. Благодаря ему можно вводить буквы как кириллические, так и латинские, а поиск при этом проходит корректно.

Таким образом, веб-сервис для организации авторизованного доступа страховых полисов, созданный на основе React.js, Spring и Spring Secure, обеспечивает благодаря использованию трёхуровневой архитектуры надёжную защиту данных, отсутствие замедлений в работе при увеличении числа пользователей и гибкость дальнейшей разработки.

Библиографические ссылки

1. Согласование модельного законодательного акта об автостраховании в Союзном государстве завершено [Электронный ресурс]. URL: <https://belrus.ru/info/soglasovanie-modelnogo-zakonodatel'nogo-akta-ob-avtostraxovanii-v-soyuznom-gosudarstve-zavershenno/> (date of access: 28.09.2023).
2. Европротокол (Извещение о ДТП) [Электронный ресурс]. URL: <https://www.bns.by/case/evro-protokol-notice-of-accident/> (date of access: 28.09.2023).
3. Бумфрей Ф. XML. Новые перспективы WWW. Пер. с англ. / Ф. Бумфрей, О. Диренцо, Й. И. Дакетт. - Москва : ДМК Пресс. 688 с.
4. Why Spring? [Electronic resource]. URL: <https://spring.io/why-spring> (date of access 27.09.2023).