Научная статья
УДК 004.4
EDN JJRQDS
https://doi.org/10.34216/2587-6147-2026-1-71-40-47
Мария Владимировна Исаева[1]
Людмила Юрьевна Киприна[2]
Никита Михайлович Семенов[3]
[1] Российский технологический университет МИРЭА, г. Москва, Россия
[2,3] Костромской государственный университет, г. Кострома, Россия
[1] mary_is@rambler.ru; https://orcid.org/0000-0002-0714-4424
[2] lskipr@gmail.com; https://orcid.org/0000-0002-0629-7699
[3] semenovknikita@gmail.com; https://orcid.org/0009-0003-5175-876X

ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ В ИНФОРМАЦИОННОЙ СИСТЕМЕ УПРАВЛЕНИЯ ПРОЕКТНО-ОБРАЗОВАТЕЛЬНЫМИ ИНТЕНСИВАМИ СТУДЕНТОВ

Аннотация. *Статья посвящена проектированию и реализации подсистемы идентификации и контроля доступа в информационной системе управления проектно-образовательными интенсивами студентов. В качестве базового механизма аутентификации выбрана технология JSON Web Tokens, обеспечивающая безопасную передачу данных без необходимости хранения состояния сессии на сервере, что соответствует принципам RESTful-архитектуры и повышает масштабируемость системы. В работе детально описана логика работы модуля, включая процесс аутентификации на клиентской стороне с использованием RTK Query. Особое внимание уделено модели управления доступом на основе ролей, реализованной в рамках системы. Введена концепция «текущей роли» пользователя, позволяющая корректно обрабатывать сценарии с множественными ролями, для чего разработан специальный permission-класс в Django REST Framework.*
Ключевые слова: *идентификация, аутентификация, авторизация, JSON Web Tokens (JWT), RBAC, Django REST Framework, RTK Query, управление доступом, образовательные интенсивы*

Для цитирования: Исаева М. В., Киприна Л. Ю., Семенов Н. М. Идентификация пользователей в информационной системе управления проектно-образовательными интенсивами студентов // Технологии и качество. 2026. № 1(71). С. 40–47. https://doi.org/10.34216/2587-6147-2026-1-71-40-47.

Original article
Maria V. Isaeva[1]
Lyudmila Yu. Kiprina[2]
Nikita M. Semenov[3]
[1] Russian Technological University MIREA, Moscow, Russia
[2,3] Kostroma State University, Kostroma, Russia

IDENTIFICATION OF USERSIN THE INFORMATIONMANAGEMENTSYSTEM FOR PROJECT-BASED EDUCATIONALINTENSIVESTUDENTS

Abstract. *The article is devoted to the design and implementation of the subsystem of identification and access control in the information management system of design and educational intensive students. The JSON Web Tokens technology has been chosen as the basic authentication mechanism, which ensures secure data transfer without the need to store session status on the server, which complies with the principles of a RESTful architecture and increases the scalability of the system. The paper describes in detail the logic of the module, including the authentication process on the client side using RTK Query. Special attention is*

paid to the role-based access control model implemented within the system. The concept of the user's "current role" has been introduced, which allows for the correct handling of scenarios with multiple roles, for which a special permission class has been developed in the Django REST Framework.
Keywords: *identification, authentication, authorization, JSON Web Tokens (JWT), RBAC, Django REST Framework, RTK Query, accesscontrol, educationalintensive courses*

For citation: Isaeva M. V., Kiprina L. Yu., Semenov N. M. Identification of usersin the informationmanagementsystem for project-based educationalintensivestudents. Technologies & Quality. 2026. No 1(71). P. 40–47. (In Russ.) https://doi.org/ 10.34216/2587-6147-2026-1-71-40-47.

Цифровая трансформация образовательных учреждений в настоящее время проходит по двум различным траекториям. С одной стороны, наблюдается эффективная цифровизация административных процессов, с другой – недостаточность информатизации образовательного процесса. Внедрение информационных технологий в образовательный процесс зачастую сводится к использованию стандартных решений, таких как системы управления обучением, например LMSMoodle. Такие платформы предлагают базовый функционал, который не всегда возможно адаптировать к специфике учебных планов, методикам преподавания и построению индивидуальных образовательных траекторий.

Параллельно в современном образовании обозначился устойчивый тренд на усиление практико-ориентированной составляющей образовательного процесса. Такой подход погружает обучающихся в решение прикладных практических задач и формирует ключевые профессиональные компетенции [1, 2].

Проектно-образовательные интенсивы – ограниченные по времени форматы, в рамках которых междисциплинарные студенческие команды разрабатывают продукт или решение для реального заказчика. Управление такими интенсивами сопряжено с высокой организационной сложностью: формирование пула тем, регистрация сотен участников, распределение по командам, координация работы с наставниками и экспертами, сбор проектных артефактов, оценка результатов [3]. Существующие цифровые инструменты не в полной мере отвечают специфическим требованиям управления образовательными интенсивами. Поэтому разработка специализированной информационной системы является актуальной научно-практической задачей.

В рамках разработки такой платформы для управления проектными интенсивами одной из задач является проектирование и реализация эффективного модуля управления доступом, обеспечивающего базовый уровень информационной безопасности всей системы. Управление доступом в информационных системах представляет собой важную задачу, направленную на предотвращение несанкционированного доступа к данным и ресурсам. Доступ к информационной системе обычно состоит из процессов аутентификации и авторизации. Под аутентификацией понимается процедура подтверждения подлинности пользователя, тогда как авторизация определяет совокупность предоставленных ему прав доступа.

К традиционным методам аутентификации относится подход, основанный на использовании пары логин – пароль, современные же подходы включают многофакторную аутентификацию, биометрические методы, а также протоколы на основе открытых ключей, которые обеспечивают высокий уровень безопасности. Анализ описанных в литературе решений [4–8] показал, что для систем с ограниченной пользовательской базой предпочтительным является применение токенов JSON Web Token (JWT), что обусловлено их простотой интеграции и достаточным уровнем безопасности. Результаты сравнительного моделирования и исследования пользовательских приложений подтверждают эффективность решений, построенных на основе JWT, для задач управления доступом [5]. При этом значительный объем научной литературы посвящен вопросам безопасности применения JWT, включая анализ потенциальных уязвимостей и способов их устранения [6–8].

На основе проведенного анализа в качестве основы для подсистемы идентификации и контроля доступа в разрабатываемой информационной системе выбрана технология JWT.

JWT представляет собой открытый стандарт (RFC 7519) для создания компактных и самодостаточных токенов доступа, обеспечивающих безопасную передачу информации между сторонами в виде объекта JSON [9]. Данная технология получила широкое распространение в качестве базового метода аутентификации в веб-приложениях.

Логика аутентификации реализованного модуля представлена в виде диаграммы последовательности на рисунке 1.

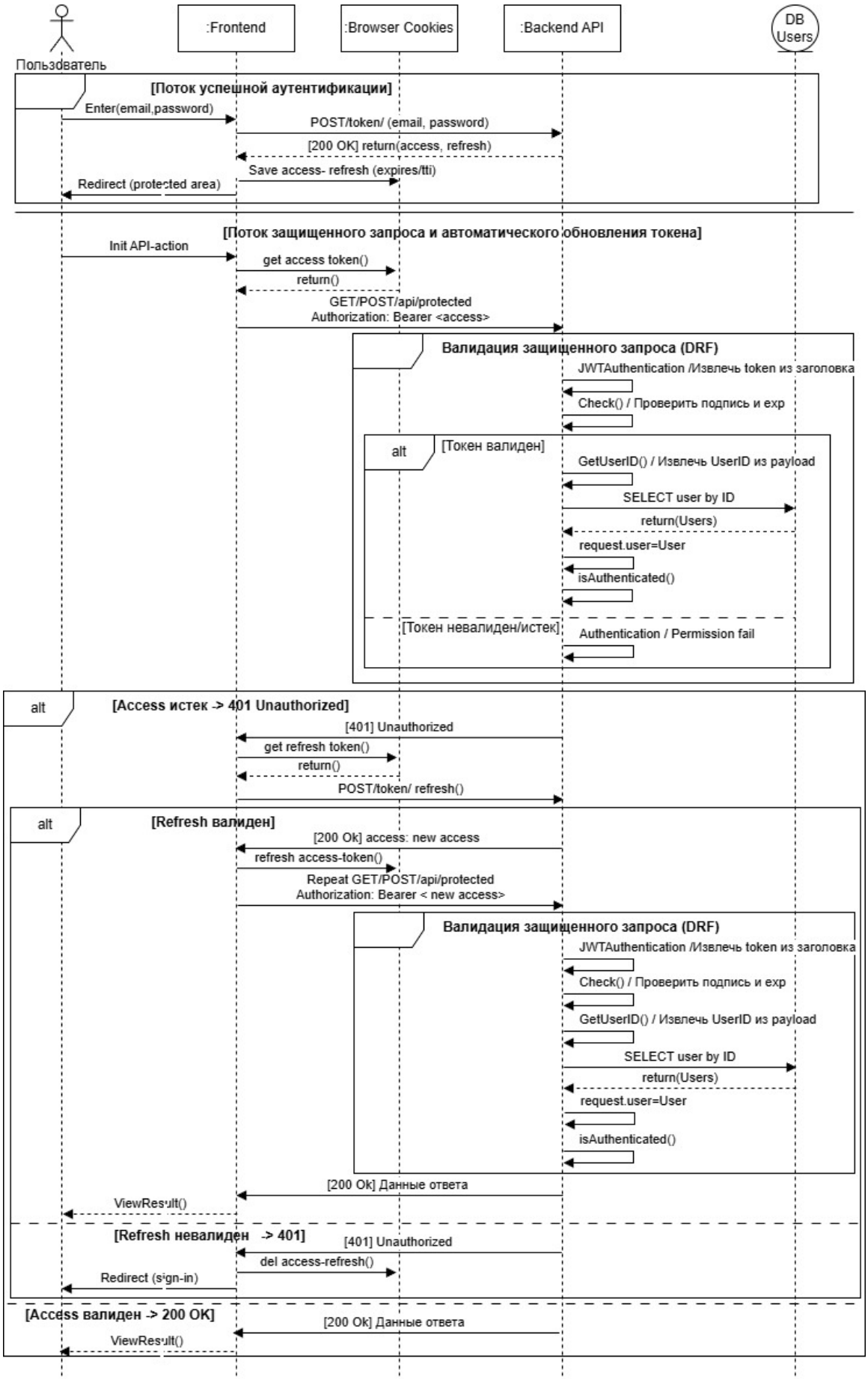

Рис. 1. Алгоритм аутентификации пользователей системы

В общем случае работа модуля аутентификации пользователей состоит из следующих шагов:

1. Процесс аутентификации пользователя инициируется на клиентской стороне через специализированный компонент SingInPage, доступный по маршруту /sign-in/. Интерфейс компонента предоставляет форму для ввода учетных данных (email и пароль). При их отправке вызывается мутация useSignInMutation() из библиотеки RTK Query, которая формирует и отправляет на сервер POST-запрос к эндпойнту /token/. В теле запроса передаются введенные данные. Ответом на запрос является пара JWT-токенов access и refresh, которые, после успешного выполнения запроса и получения ответа в компоненте, сохраняются для последующего использования в cookies браузера. Для этого используется сторонняя библиотека js – cookie. Значениями expires задается срок хранения в днях: access сохраняется на короткие срок, refresh – на более длительный. Сохранение токенов в cookies обеспечивается с помощью js – cookie:

```
Cookies.set('access', responseData.access, {expires: 1});
Cookies.set('refresh', responseData.refresh, {expires: 15});
```

2. Для автоматического добавления актуального access-токена в заголовок Authorization: Bearer<> всех последующих запросов к защищенным API, а также для контроля его срока действия реализована расширенная функция базового запроса baseQueryWithReauth. Данная функция перехватывает ответы с кодом 401 (Unauthorized), что свидетельствует об истечении срока действия access-токена, и автоматически инициирует запрос на его обновление (POST /token/refresh/), используя сохраненный refresh-токен. Если refresh-токен также недействителен, пользователь перенаправляется на страницу авторизации.

3. Защита API и проверка подлинности пользователей на сервере реализована с использованием механизма JWT-аутентификации на базе библиотеки djangorestframework-simplejwt.

В глобальных настройках проекта – settings.py – в качестве стандартного класса аутентификации указан JWTAuthentication:

```
REST_FRAMEWORK = {
'DEFAULT_AUTHENTICATON_CLASSES': (
'rest_framework_simplejwt.authentication.JWTAuthen
tication', ),
}
```

Механизм контроля доступа при обращении пользователя к защищенному эндпойнту функционирует по следующему алгоритму:

– извлекается токен из заголовка Authorization запроса, проверяется в нем цифровая подпись и временные ограничения;
– идентификатор пользователя из полезной нагрузки токена используется для получения информации об объекте User из базы данных. Объект User присваивается атрибуту request.user, делая его доступным для дальнейшей обработки.

Контроль доступа на уровне представлений реализован с использованием встроенного класса IsAuthenticated из Django REST Framework. Данный класс гарантирует, что обработка запроса будет осуществляться только при успешной аутентификации пользователя: при наличии в запросе валидного JWT-токена. Все неаутентифицированные запросы автоматически отклоняются, возвращая HTTP-статус 401 Unauthorized.

Данный алгоритм обеспечивает безопасную аутентификацию пользователей в распределенных системах, построенных с использованием RESRful-архитектуры и контролирует доступ к защищенным ресурсами через встроенные механизмы DjangoRESTFramework.

В рамках системы предусмотрен механизм предоставления доступа по паролю, который генерируется для каждого пользователя администратором после его регистрации в системе и направляется на указанный email-адрес. Данный подход к аутентификации пользователей соответствует стандарту NIST SP 800-63, устанавливающему технические требования и рекомендации по управлению цифровой идентификацией и описывающему уровни аутентификации, а именно соответствует базовому уровню AAL1. Этот уровень предназначен для приложений с низким уровнем риска, где нарушение учетных данных не влечет серьезных последствий. После успешной аутентификации вступает в силу механизм авторизации, определяющий права пользователя.

Для авторизации в системе реализована формализованная ролевая модель контроля доступа (RBAC), определенная в ANSI INCITS 359-2004 – Role-Based Access Control (RBAC) [10]. При ролевой модели доступ к сущностям информационной системы осуществляется на основе вхождения пользователей в роли. Роли могут быть выстроены как на основе отдельных персоналий, организационной структуры, так и на основе функциональных обязанностей или смешанным образом. Данная модель обеспечивает как разграничение прав между различными ролями, так и контроль за действиями пользователей в пределах назначенных им привилегий [11, 12].

Анализ задач проектно-образовательных интенсивов позволил выявить набор функциональных ролей пользователей, участвующих в процессе. К ним относятся: организатор (создатель и куратор интенсива и мероприятий в нем), тьютор (наставник команды), тимлид (лидер студенческой команды), студент (участник команды), жюри (специалист, оценивающий работу студентов).

В разработанной информационной системе данный набор был адаптирован и формализован в виде четырех базовых ролей пользователя: администратор, организатор, преподаватель, студент. Каждая роль имеет определенный набор функциональных возможностей.

Администратор осуществляет управление учебными потоками, группами, учетными записями студентов и преподавателей, а также ведение справочников (корпуса, аудитории и другие системные сущности).

В обязанности организатора входит формирование команд и расписания, обработка образовательных запросов от команд, а также мониторинг оценок участников.

Преподаватель выполняет в рамках интенсива две ключевые функции: тьютора и члена жюри. В роли тьютора, закрепленного за командой, он управляет распределением ролей среди студентов и выставляет оценки ее участникам. В качестве члена жюри он оценивает ответы команд на мероприятиях.

Студент является непосредственным участником интенсива. Его функционал включает: работу с kanban-доской собственной команды, управление ролями участников команды, отправку ответов на мероприятия и формирование образовательных запросов (при наличии прав тимлида).

Любой пользователь системы может иметь несколько ролей, что существенно усложняет реализацию проверки прав доступа, так как права доступа нельзя будет определять через простое сравнение типа роли. В случае множественных ролей такой легкий подход становится неприменимым. Для решения данной проблемы в работе предложена и реализована концепция «текущей роли пользователя», суть которой заключается в том, что при выполнении любого авторизованного запроса на стороне сервера пользователь должен явно передать текущую роль. Такой подход позволяет осуществлять поддержку нескольких ролей пользователя, а также проводить проверку прав доступа только относительно текущей роли.

Для реализации предложенной концепции в Django создано специальное правило доступа IsAuthenticatedWithRole (рис. 2), которое расширяет стандартное правило IsAuthenticated из Django REST Framework. Данное правило проверяет не только факт авторизации пользователя, но и корректность указанной роли, от имени которой выполняется запрос.

В случае если у пользователя в системе только одна роль, она автоматически считается текущей и передавать заголовок необязательно. В случае если у пользователя несколько ролей, система требует явного указания роли в HTTP-заголовке X-Active-Role. При возникновении любой ошибки (отсутствие заголовка, ошибочный заголовок, роль не принадлежит пользователю) система возвращает HTTP-код ошибки 403 (Доступ запрещен) с сообщением «Заголовок X-Active-Role не указан» или «Неправильная роль пользователя» и дальнейшие проверки прав не выполняются.

Для реализации функциональности, требующей идентификации активной роли пользователя в пользовательском интерфейсе, было принято решение использовать механизм локального хранилища браузера (localStorage).

Алгоритм работы реализован следующим образом:

1. После успешной аутентификации и получения с сервера данных о пользователе выполняется проверка количества ролей, ассоциированных с его учетной записью.

2. Если пользователю назначена единственная роль, она автоматически фиксируется как активная и сохраняется в localStorage.

3. В случае если пользователь обладает несколькими ролями, клиентское приложение инициирует интерактивный диалог – модальное окно, в рамках которого пользователь должен выбрать одну из доступных ему ролей для текущей сессии. Выбранная роль затем сохраняется в localStorage и используется для последующих запросов к API.

4. Все последующие HTTP-запросы к API дополняются специальным заголовком X-Active-Role, значением которого является идентификатор активной роли, что обеспечивает контекстную авторизацию на стороне сервера.

ВЫВОДЫ

Таким образом, в данном исследовании разработана и внедрена архитектура модуля аутентификации и авторизации пользователей, интегрированного в информационную систему управления образовательными интенсивами. В качестве базового механизма выбрана технология JSONWebTokens, что позволяет обеспечить безопасность передаваемых данных.

Такой подход соответствует принципам RESTful-архитектуры и позволяет хранить информацию о пользователе в самом токене. Это устраняет необходимость поддержки серверного состояния сессии, что повышает гибкость информационной системы.

Реализация модуля основана на двухуровневой модели обеспечения безопасности доступа. На клиентском уровне с использованием библиотеки RTK Query реализован автоматический процесс обновления JWT-токена, который обеспечивает непрерывность сессии пользователя. На серверном уровне с помощью библиотеки JWT Authentication в каждый токен добавляется криптографическая подпись. Это способствует строгой верификации запросов пользователей и служит надежной защитой от несанкционированного доступа к данным.

```python
class IsAuthenticatedWithRole(permissions.BasePermission):
    def has_permission(self, request, view):
        if not permissions.IsAuthenticated().has_permission(request, view):
            return False  # Если не аутентифицирован — отказываем

        user = request.user
        user_roles = user.roles.values_list("name", flat=True)

        # Если роль у пользователя одна, сохраняем ее (можно не передавать заголовок)
        if user_roles.count() == 1:
            user.active_role = user_roles.first()
            return True

        active_role = request.headers.get("X-Active-Role")

        # Если ролей несколько проверяем заголовок на однозначное определение роли в запросе
        if active_role:
            active_role = urllib.parse.unquote(active_role)

            if active_role in user_roles:
                # Добавляем активную роль если передали правильно
                user.active_role = active_role
                return True
            else:
                # Если передали неправильную роль
                logger.warning("Неправильная роль пользователя")
                user.active_role = None
                raise PermissionDenied(
                    "Необходимо указать корректную роль пользователя в заголовке X-Active-Role"
                )
        else:
            # Роль не передана (нет заголовка)
            logger.warning("Заголовка X-Active-Role нет")
            user.active_role = None
            raise PermissionDenied(
                "Необходимо указать текущую роль пользователя в заголовке X-Active-Role"
            )
```

Рис. 2. Описание класса IsAuthenticatedWithRole, реализующего процедуру доступа в систему

СПИСОК ИСТОЧНИКОВ

1. Пузанова Ж. В., Кострикин Е. Г. Проектный подход в обучении: практика в вузах // Вестник Российского университета дружбы народов. Серия: Социология. 2025. Т. 25, № 3. С. 652–664.
2. Донская Е. Ю. Применение проектного обучения в высшей школе // Мир науки. Педагогика и психология. 2023. Т. 11, № 3. URL: https://mir-nauki.com/PDF/13PDMN323.pdf (дата обращения: 12.01.2026).
3. Абрамова Е. А. Применение проектного подхода при реализации образовательного курса в вузе // Современные наукоемкие технологии. Региональное приложение. 2022. № 2(70). С. 39–46. URL: https://snt-isuct.ru/article/view/4573. (дата обращения: 12.01.2026).
4. Феоктистов И. В. Сравнительное исследование методов аутентификации в информационных системах // Инновации и инвестиции. 2023. № 7. С. 193–198.

5. Девицына С. Н., Пилькевич П. В., Удод Е. В. Способы улучшения защищённости сервисов, использующих JWT-токены // Экономика. Информатика. 2023. № 50(1). С. 144–151.
6. Монахов М. Ю., Уймин А. Г. Инфраструктура JsonWebToken. Инфраструктура защиты // Информатика, вычислительная техника и управление. Серия: Естественные и технические науки. 2023. № 1. С. 136–141.
7. О некоторых особенностях JWT аутентификации в веб-приложениях / А. Б. Бетелин, И. Б. Егорычев, А. А. Прилипко, Г. А. Прилипко, С. Г. Романюк, Д. В. Самборский // Труды НИИСИ РАН. 2021. Т. 11, № 1. С. 4–10.
8. Большаков А. С., Добряков А. С., Туктаров Р. Р. О реализации информационной защищенности системы распределенного хранения данных малого бизнеса // Инженерный вестник Дона. 2025. № 2. С. 374–397. URL: http://www.ivdon.ru/uploads/article/pdf/IVD_54N12y24_Bolshakov.pdf_16c0147d0c.pdf (дата обращения: 12.01.2026).
9. RFC 7519. JSON Web Token (JWT). URL: https://datatracker.ietf.org/doc/html/rfc7519 (дата обращения: 12.01.2026).
10. Role Based Access Control. URL: https://csrc.nist.gov/projects/role-based-access-control (дата обращения: 15.01.2026).
11. Давыдов Д. Стандарты ролевого подхода к управлению доступом // Cleverics.ru. Digital Enterprise : портал № 1 по управлению цифровыми и информационными технологиями. URL: https://cleverics.ru/digital/2015/10/incits-rbac-standards (дата обращения: 15.01.2026).
12. Корниенко С. В., Протасов М. С. Особенности использования ролевой модели без опасности (модель взаимоисключающих ролей) в вузе // Интеллектуальные технологии на транспорте. 2025. № 4(44). С. 5–16.

REFERENCES

1. Puzanova Zh. V., Kostrikin E. G. Project-based approach in teaching: University practice. *Vestnik Rossijskogo universiteta druzhby narodov. Seriya: Sociologiya* [Bulletin of the Peoples' Friendship University of Russia. Series: Sociology]. 2025;25(3):652–664. (In Russ.)
2. Donskaya E.Yu. Application of project-based learning in higher education. *Mir nauki.* Pedagogika *i psihologiya* [World of Science. Pedagogy and psychology]. 2023;11(3). URL: https://mir-nauki.com/PDF/13PDMN323.pdf. (accessed 12.01.2026). (In Russ.)
3. Abramova E. A. Application of the project approach in the implementation of the educational at the university. *Sovremennye naukoyomkie tekhnologii. Regional'noe prilozhenie* [Modern high-techtechnologies. Regional application]. 2022;2(70):39–46. URL: https://snt-isuct.ru/article/view/4573 (accessed 12.01.2026). (In Russ.)
4. Feoktistov I. V. Comparative study of authentication methods in information systems. *Innovacii i investicii* [Innovations and Investments]. 2023;7:193–198. (In Russ.)
5. Devitsyna S., Pilkevich P., Udod E. Ways to improve the security of servicesusing JWT-tokens. *Ekonomika. Informatika* [Economics. Computer Science]. 2023;50(1):144–151. (In Russ.)
6. Monakhov M., Uymin A. JSON Web Token infrastructure. *Informatika, vychislitel'naya tekhnika i upravlenie. Seriya: Estestvennye i tekhnicheskie nauki* [Security infrastructure. Series: Natural and Technical Sciences]. 2023;1:136–141. (In Russ.)
7. Betelin A. B., Egorychev I. B., Prilipko A. A., Prilipko G. A., Romanyuk S. G., Samborskiy D. V. Some features of JWT authentication in web applications. *Trudy NIISI* [Proceedings of the Scientific Research Institute of System Analisis of the Russian Academy of Sciences]. 2021;11,1:4–10. (In Russ.)
8. Bolshakov A. S., Dobryakov A. S., Tuktarov R. R. On the implementation of information security in distributed data storage system for small businesses. *Inzhenernyj vestnik Dona* [Engineering Bulletin of the Don]. 2025;2(122):374–397. URL: http://www.ivdon.ru/uploads/article/pdf/IVD_54N12y24_Bolshakov.pdf_16c0147d0c.pdf (accessed 12.01.2026). (In Russ.)
9. RFC 7519. JSON Web Token (JWT). URL: https://datatracker.ietf.org/doc/html/rfc7519 (accessed 12.01.2026).
10. Role Based Access Control. URL: https://csrc.nist.gov/projects/role-based-access-control (accessed 15.01.2026).
11. Davydov D. Standards of a role-based approachtoaccessmanagement. Cleverics.ru. Digital Enterprise, web site. URL:https://cleverics.ru/digital/2015/10/incits-rbac-standards (accessed 15.01.2026)* (In Russ.)

* Перевод названия источника выполнен авторами статьи / Translated by author’s of the article.

12. Kornienko S. V., Protasov M. S. Implementation of the role-based access control model (mutually exclusive roles model) at higher education institution. *Intellektual'nye tekhnologii na transporte* [Intelligent technologies in transport]. 2025;4(44):5–16. (In Russ.)

Статья поступила в редакцию 12.01.2026
Принята к публикации 09.02.2026

ИНФОРМАЦИЯ ОБ АВТОРАХ:
М. В. Исаева, кандидат технических наук, доцент
Л. Ю. Киприна, кандидат технических наук, доцент
Н. М. Семенов, магистрант