

Общий метод управления доступом на основе ролей RBAC для всех маршрутов

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описан подход проверки прав доступа пользователей к ресурсу на основе ролей. Метод проверки реализован в одной функции для всех предоставляемых маршрутов. Функция относится к типу middleware, в которую передаются параметры необходимые для настройки и определения возможности доступа. В работе представлен программный код middleware-компонента и разобран практический пример его внедрения для гибкого управления доступом к конечным точкам приложения. Описан последовательный подход доступа клиента к ресурсам системы.

Ключевые слова: RBAC, авторизация, маршрутизация, права доступа.

A common RBAC role-based access control method for all routes

Vikhlyayev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the process of creating diagrams using the PlantUML language. A detailed description of the syntax and key aspects related to the definition of classes, relationships between classes and use cases is provided. A local and online editor are involved in creating diagrams.

Keywords: RBAC, authorization, routing, access rights.

1 Введение

1.1 Актуальность

RBAC (Role-Based Access Control) – это метод управления доступом, основанный на ролях. Применение RBAC обеспечивает централизованное управление доступом, повышая его безопасность и упрощая администрирование, особенно в системах с большим количеством пользователей. Использование данной модели позволяет сократить риски, связанные с некорректным или избыточным предоставлением прав доступа, а также улучшить контроль и аудит доступа в системе. Возможность настройки гибких политик доступа, которые можно привязать как к отдельным маршрутам, так и к группам маршрутов. Это достигается за счёт использования конфигурации, которая определяет связи между маршрутами,

действиями и ролями. Например, роли "Администратор", "Менеджер" и "Пользователь" могут иметь доступ к различным наборам операций в системе, что позволяет чётко разграничить их полномочия.

1.2 Обзор исследований

В.Л.Самсонов описал концепцию управления доступом пользователей образовательного портала РГУПС на основе модели "RBAC" [1]. Т.Редмонд представил управление доступом в EXCHANGE на основе ролей и RBAC MANAGER [2]. И.Б.Саенко, И.В.Котенко, С.Н.Круглов использовали подход к решению "проблемы извлечения ролей" при формировании модели RBAC на основе генетических алгоритмов [3]. Д.Д.Кононов применил расширенную ролевую модель безопасности на основе RBAC для защиты веб-сервисов [4]. С.Н.Городко, Н.В.Марковская добавили модуль RBAC на сайт УИИ2 [5]. Н.А.Семенова провела анализ безопасности взаимодействия ролевой модели разграничения доступа и модели распределенного администрирования [6]. В исследовании Н.А.Семёнова приведены примеры реализации ролевой модели разграничения доступа с использованием методов логики линейного времени [7]. В работе О.В.Акимин расписана технология управления доступом на основе ролей [8].

1.3 Цель исследования

Цель исследования – описать систему проверки прав доступа основанной на ролевой модели управления доступом (RBAC).

2 Материалы и методы

В материалах приводится описание промежуточной функции, проверки прав доступа пользователя к запрашиваемым ресурсам на основе полученных данных метода HTTP-доступа, информации о роли пользователя и значения маршрута.

3 Результаты и обсуждения

Ролевая модель управления доступом (RBAC) представляет собой метод управления доступом на основе ролей, назначаемых пользователям, что позволяет централизованно определять и управлять доступом к ресурсам системы. Основными компонентами RBAC являются пользователи, роли, разрешения и ресурсы. Пользователи представляют собой субъекты, которым назначаются роли, определяющие их права в системе. Роли, в свою очередь, представляют собой логические объединения разрешений, упрощающие управление доступом путем группировки прав доступа. Разрешения характеризуют конкретные действия, которые пользователь может выполнять с определенными ресурсами, такими как API-эндпоинты, базы данных или файлы (рис.1).

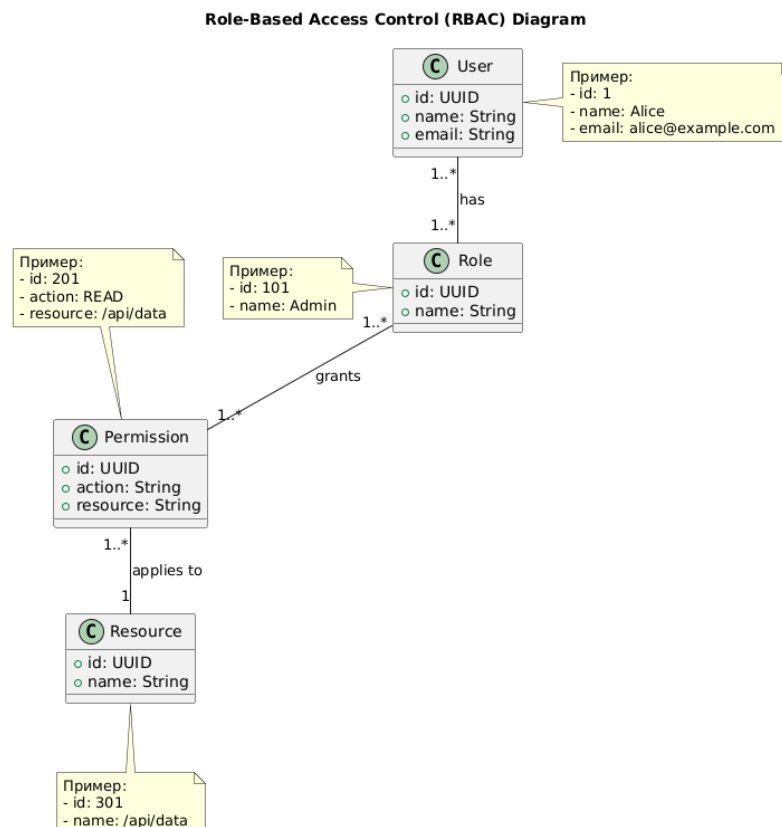


Рис. 1. Доступ пользователя к ресурсам через RBAC

Реализация валидации на основе RBAC включает несколько ключевых этапов. На первом этапе происходит идентификация пользователя и сопоставление ему одной или нескольких ролей. На втором этапе система проверяет наличие у назначенной роли соответствующих разрешений для выполнения запрашиваемого действия. Данная проверка осуществляется путем сопоставления набора разрешений, закрепленных за ролью, с требованиями запрашиваемого ресурса. Если роль пользователя включает в себя необходимое разрешение, доступ к ресурсу предоставляется; в противном случае доступ отклоняется (рис.2).

```

const rbac = {
  Admin: {
    '/users': ['GET', 'POST', 'PUT', 'DELETE'],
    '/stations': ['GET', 'POST', 'PUT', 'DELETE'],
    '/stations/:id': ['GET', 'POST', 'PUT', 'DELETE'],
  },
  develop: {
    '/stations': ['GET'],
    '/stations/:id': ['GET', 'POST', 'PUT', 'DELETE'],
  },
  guest: {
    '/stations': ['GET'],
    '/stations/:id': ['GET'],
  },
};
  
```

Рис. 2. Описание доступа ролей по маршруту и методу

В современных веб-приложениях для реализации проверки доступа пользователей на основе ролей (RBAC, Role-Based Access Control) широко применяется архитектурный подход, включающий использование единой промежуточной обработки (middleware) для всех маршрутов. Данный метод позволяет централизованно управлять проверкой прав доступа, улучшая читаемость и поддерживаемость кода.

Клиентская сторона отправляет HTTP-запрос на определённый маршрут веб-приложения. Запрос может содержать такие параметры, как заголовки авторизации, токен доступа и маршрут, определяющий желаемую операцию (например, чтение данных пользователя, создание записи и т. д.).

На этапе маршрутизации входящий запрос направляется к определённому маршруту в соответствии с конфигурацией маршрутов приложения. Каждый маршрут связан с определённой функциональностью, реализуемой в соответствующем контроллере.

Прежде чем запрос достигает контроллера, он обрабатывается RBAC middleware. Middleware выполняет центральную роль в обеспечении безопасности, проверяя наличие у пользователя прав доступа к запрашиваемому ресурсу. На первом этапе работы middleware осуществляется проверка наличия и валидности токена авторизации. Если токен отсутствует или недействителен, middleware немедленно возвращает ответ с кодом состояния HTTP 401 Unauthorized. При успешной аутентификации middleware извлекает роль пользователя из токена или базы данных и проверяет её соответствие необходимым разрешениям, которые заданы для текущего маршрута. Разрешения могут быть определены в виде статической конфигурации или динамически получены из внешнего источника, такого как система управления доступом. Если проверка прав доступа завершилась успешно, middleware передаёт управление в соответствующий контроллер маршрута. На этом этапе запрос достигает логики приложения, где выполняется обработка бизнес-правил, взаимодействие с базой данных или внешними API.

Контроллер выполняет основное действие, связанное с маршрутом, например, получение данных из базы, создание новой записи или удаление существующей. Результаты обработки возвращаются в виде HTTP-ответа с соответствующим кодом состояния и данными. После выполнения запроса в контроллере результирующий HTTP-ответ передаётся клиенту. Если в процессе выполнения запроса возникли ошибки или нарушения прав доступа, клиент получает сообщение об ошибке с кодами состояния 403 Forbidden или 401 Unauthorized (рис.3).

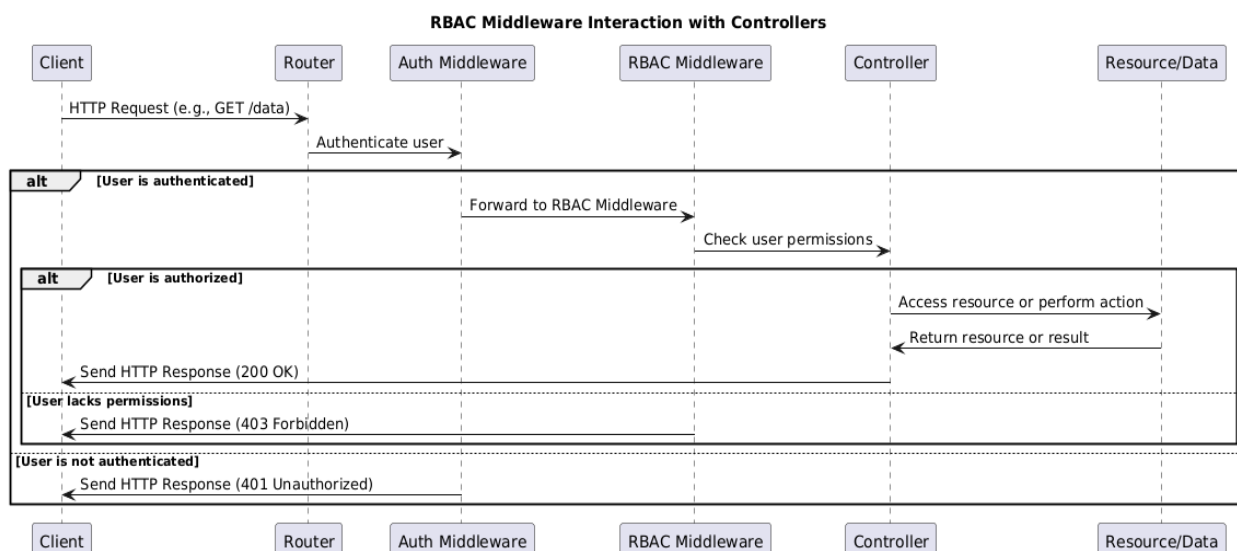


Рис. 3. Последовательность этапов исполнения

Использование единого middleware для всех маршрутов обеспечивает централизованную проверку доступа и упрощает управление безопасностью приложения (рис. 4).

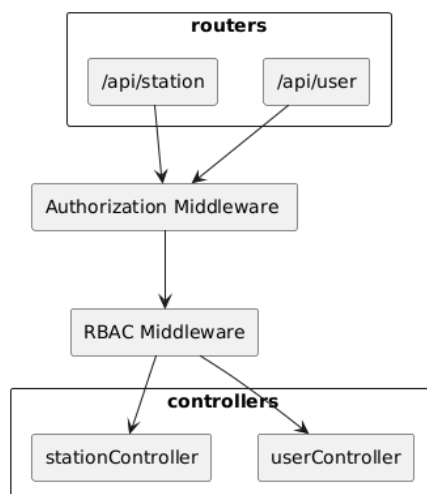


Рис. 4. Использование одной промежуточной функции для всех маршрутов запроса

Этот подход также минимизирует дублирование кода, так как логика проверки доступа сосредоточена в одном месте, а не распределена по контроллерам. Внедрение RBAC middleware способствует соблюдению принципов модульности и повышения уровня безопасности веб-приложений

4 Вывод

Рассматриваемый метод управления доступом на основе ролей (RBAC) использует единую централизованную функцию проверки прав для всех маршрутов. В качестве модификаторов доступа выступают методы HTTP-запроса. Права пользователя на доступ к ресурсам определяются его ролью. Этот подход обеспечивает логическую группировку клиентов системы и

создаёт удобный механизм для управления их перемещением и правами доступа.

Библиографический список

1. Самсонов В.Л. Концепция управления доступом пользователей образовательного портала РГУПС на основе модели "RBAC" // В сборнике: Труды Международной научно-практической конференции "Транспорт-2014". В 4-х частях. 2014. С. 107-109.
2. Редмонд Т. Управление доступом в EXCHANGE на основе ролей и RBAC MANAGER // Windows 2000 Magazine/Re. 2015. № 6. С. 48.
3. Саенко И.Б., Котенко И.В., Круглов С.Н. Подход к решению "проблемы извлечения ролей" при формировании модели RBAC на основе генетических алгоритмов // Методы и технические средства обеспечения безопасности информации. 2016. № 25. С. 113.
4. Кононов Д.Д. Расширенная ролевая модель безопасности на основе RBAC для защиты веб-сервисов // В сборнике: Решетневские чтения. Материалы XXIII Международной научно-практической конференции, посвященной памяти генерального конструктора ракетно-космических систем академика М.Ф. Решетнева. В 2-х частях. Под общей редакцией Ю.Ю. Логинова. 2019. С. 432-433.
5. Городко С.Н., Марковская Н.В. Модуль на YII2: добавление RBAC (роли, разрешения, правила) для сайта "центр планирования и развития карьеры" на базе Гродненского государственного университета имени Янки Купалы // Интернаука. 2018. № 16-1 (50). С. 5-7.
6. Семенова Н.А. Анализ безопасности взаимодействия ролевой модели разграничения доступа и модели распределенного администрирования // Безопасность информационных технологий. 2011. Т. 18. № 3. С. 50-56.
7. Семёнова Н.А. Верификация ролевой модели разграничения доступа с использованием методов логики линейного времени // Проблемы информационной безопасности. Компьютерные системы. 2010. № 4. С. 7-13.
8. Акимин О.В. Управление доступом на основе ролей (RBAC) // E-Scio. 2023. № 11 (86). С. 8-12.