

УДК 33

DOI: 10.34670/AR.2022.17.39.028

Разработка интеграционного решения АИС «Штрафы» с использованием современных web-технологий

Костиков Юрий Александрович

Кандидат физико-математических наук,
заведующий кафедрой 812,
Московский авиационный институт
(национальный исследовательский университет),
125993, Российская Федерация, Москва, Волоколамское шоссе, 4;
e-mail: jkostikov@mail.ru

Романенков Александр Михайлович

Кандидат технических наук,
доцент кафедры 812,
Московский авиационный институт
(национальный исследовательский университет),
125993, Российская Федерация, Москва, Волоколамское шоссе, 4;
e-mail: romanaleks@gmail.com

Тараненко Даниил Лемович

Магистрант,
Московский авиационный институт
(национальный исследовательский университет),
125993, Российская Федерация, Москва, Волоколамское шоссе, 4;
e-mail: silentiumdz@gmail.com

Аннотация

В данной статье с использованием современных средств разработки web-приложений предложено решение для интеграции модуля АИС «Штрафы» в систему ФССП. Описан алгоритм обмена данными между модулем и основным программным продуктом. На языке TypeScript представлена реализация этого алгоритма. Рассматривается методика создания интеграционного решения программно-аналитического комплекса АИС «Штрафы» на примере городского хозяйства Нижнего Новгорода с использованием NestJs и TypeScript. Основной задачей этого программного комплекса является автоматизация учета и контроля нарушения правил использования парковочных мест персональным автотранспортом. Программное решение АИС «Штрафы» является модульной составляющей комплексного специализированного программного обеспечения, в котором происходит управление настройками оборудования на улицах, поддержка информационного портала и мобильного приложения для граждан, проектирование мест нанесения разметки, логистика периферийного оборудования.

Для цитирования в научных исследованиях

Костиков Ю.А., Романенков А.М., Тараненко Д.Л. Разработка интеграционного решения АИС «Штрафы» с использованием современных web-технологий // Экономика: вчера, сегодня, завтра. 2022. Том 12. № 3А. С. 246-250. DOI: 10.34670/AR.2022.17.39.028

Ключевые слова

NodeJs, NestJs, TypeScript, АИС «Штрафы», интеграционное решение, web-технологии, модуль, программный продукт.

Введение

Необходимость компонентного модуля АИС «Штрафы» обусловлена задачей контроля соблюдения правил по парковке транспортных средств и фиксации нарушений правил дорожного движения, весогабаритного контроля, правил благоустройства.

В качестве основной платформы для реализации бизнес-логики системы выбрано средство разработки Node.js [AWS Well-Architected Framework, November 2018, www], которое является программным средством для выполнения кода на языке JavaScript. Основной особенностью Node.js является то, что вместо традиционного параллелизма на основе потоков предлагается событийно-управляемая неблокирующая модель ввода-вывода [Young et al., 2017], легковесная и с достаточным быстродействием, которая удачно подходит для разработки приложений реального времени, обрабатывающих большие объемы данных и имеющих распределенную архитектуру [Guijarro et al., 2018]. В качестве основного языка программирования для реализации логики внутри системы используется TypeScript, а для упрощения процесса разработки выбран фреймворк NestJS, который позволяет создавать серверные приложения enterprise уровня. Данный фреймворк ориентирован на использование вместе с языком TypeScript и объединяет элементы объектно-ориентированного, функционального и реактивного программирования [Resende, 2018].

Одной из ключевых концепций в NestJS является применение паттерна внедрения зависимостей, что позволяет рассматривать различные сущности приложения как провайдеры, то есть можно внедрять зависимости между этими сущностями [Casciogo, Mammino, 2016]. Это означает, что объекты приложения могут создавать различные отношения друг с другом и функция конструирования и внедрения экземпляров классов может быть в значительной степени делегирована системе внедрения зависимостей NestJS. Такой подход позволяет создавать альтернативные реализации классов, что особенно полезно в юнит-тестировании.

Помимо этого, NestJS позволяет обеспечить автоматическую валидацию и разбор входных данных при помощи использования встроенного механизма, который реализован в таких библиотеках, как class-validator и class-transformer.

Интеграция системы с информационной системой СМЭВ (системой межведомственного электронного взаимодействия) ФССП (Федеральной службы судебных приставов)

Одной из основных задач для системы АИС «Штрафы» является интеграция с СМЭВ, которая необходима для реализации функционала определения наличия задолженностей у нарушителей, а также для получения исполнительных листов по судопроизводству. Интеграция

реализована на основе передачи текстовых данных с помощью обмена xml-схем. Такое решение обусловлено тем, что основной протокол взаимодействия с СМЭВ требует данных в xml-формате.

Для начала взаимодействия необходимо наличие пользователя с действующим сертификатом электронно-цифровой подписи, которая предварительно зарегистрирована в системе. Данный функционал предоставляет конечная точка `api/v1/fssp-request/sign`, через которую обязательно подписывать каждый xml-файл, который в дальнейшем используется при взаимодействии с СМЭВ.

После успешного выполнения данного этапа необходим метод создания запроса xml-документа и отправки запроса `sendFsspRequest`, который используется в конечной точке `api/v1/fssp-request/send`.

В данный метод передается файл электронно-цифровой подписи, внутренний номер документа о нарушении, по которому производится поиск данных о нарушителе. Для обращения было предусмотрено несколько сценариев взаимодействия, а также несколько сценариев ответа в зависимости от вида нарушения или типа данных, которые требовались. Основной проблемой на данном шаге было взаимодействие через длинную систему ответов на запросы и подтверждение запросов, которая могла растянуться на 5–7 шагов и была вынесена в отдельный модуль, который предназначен для ожидания ответа от соответствующего ведомства и отправления подтверждения новой части запроса. Для обработки полученных данных для различных сценариев предназначена конечная точка `api/v1/fssp-request/response` с методом `saveResponse`, который детектирует тип сценария, его результат, а также занимается обработкой ошибок и определяет дальнейшие шаги взаимодействия.

Для отображения результатов интеграции была создана отдельная страница интерфейса с фильтрами под каждый статус и возможность редактирования данных, а также благодаря логированию на всех этапах была создана подробная статистика взаимодействия с круговыми графиками по ошибкам и удачным запросам (рис. 4), тепловой картой запросов и статистикой отправки и получения запросов с возможностью фильтрации. Таким образом был получен модуль для полного взаимодействия с СМЭВ, а также сбора данных для аналитики по данному этапу работы системы.

Заключение

Таким образом, нами была описана последовательность шагов для интеграции модуля АИС «Штрафы» в глобальную многокомпонентную систему ФССП. Рассмотрены используемые технологии и представлены конкретные реализации на языке TypeScript алгоритмических нюансов организации протокола взаимодействия компонент между собой.

Для удобства работы с предлагаемым решением разработан пользовательский web-интерфейс, который позволяет отслеживать статистику нарушений.

Библиография

1. Casciaro M., Mammino L. Node.js Design Patterns. Second Edition: Master best practices to build modular and scalable server-side web applications. 2016.
2. Guijarro D. et al. A Progressive Node.js Framework Kindle Edition. 2018.
3. Kraus D., Obrist T., Hari O. Blockchains, Smart Contracts, Decentralised Autonomous Organisations and the Law. Edward Elgar Publishing, 2019.
4. Munoz-Lopez J. E. Internet Conflict of Laws: A Space of Opportunities for ODR // 14 International Law, Revista

-
- Colombiana de Derecho Internacional. 2009. P. 163-190.
5. Resende D. Hands-On Microservices with Node.js: Build, test, and deploy robust microservices in JavaScript. 2018.
 6. Rohrmann C. A., Rocha F. S., Cunha J. Some legal aspects of cloud computing contracts // Journal of International Commercial Law and Technology. 2015. Vol. 10. No. 1.
 7. Young A. et al. Holowaychuk, Nathan Rajlich. Node.js in Action, Second Edition. 2017.
 8. AWS Well-Architected Framework, November 2018. URL: https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf.
 9. Fox D., Green S. The Law of Cryptocurrencies. – 2019.
 10. Абдуллова М.О. «Смешанная задача для одного уравнения четвертого порядка» Материалы республиканская конференция «Актуальные проблемы дифференциальных уравнений и их приложения» Ташкент 15-17 декабрь 2017 г.

Development of an integration solution AIS "Penalties" with the use of modern web technologies

Yurii A. Kostikov

PhD in Physical and Mathematical Sciences,
Head of the Department 812,
Moscow Aviation Institute (National Research University),
125993, 4 Volokolamskoe highway, Moscow, Russian Federation;
e-mail: jkostikov@mail.ru

Aleksandr M. Romanenkov

PhD in Technical Sciences, Associate Professor,
Department 812,
Moscow Aviation Institute (National Research University),
125993, 4 Volokolamskoe highway, Moscow, Russian Federation;
e-mail: romanaleks@gmail.com

Daniil L. Taranenko

Master student
Moscow Aviation Institute (National Research University),
125993, 4 Volokolamskoe highway, Moscow, Russian Federation;
e-mail: silentiumdz@gmail.com

Abstract

In this article, using modern tools for developing web applications, a solution is proposed for integrating the AIS "Penalties" module into the system of the Federal Bailiffs Service. The algorithm of data exchange between the module and the main software product is described. The TypeScript language provides an implementation of this algorithm. The method of creating an integration solution for the AIS "Penalties" software-analytical complex is considered on the example of the municipal economy of Nizhny Novgorod using NestJs and TypeScript. The main objective of this software package is to automate the accounting and control of violations of the rules for the use of parking spaces by personal vehicles. The AIS "Penalties" software solution is a modular component

of complex specialized software that manages equipment settings on the streets, supports an information portal and a mobile application for citizens, designing marking places, and logistics of peripheral equipment. For the convenience of working with the proposed solution, a web-based user interface has been developed that allows us to track the statistics of violations.

For citation

Kostikov Yu.A., Romanenkov A.M., Taranenko D.L. (2022) Razrabotka integratsionnogo resheniya AIS "Shtrafy" s ispol'zovaniem sovremennykh web-tekhnologii [Development of an integration solution AIS "Penalties" with the use of modern web technologies]. *Ekonomika: vchera, segodnya, zavtra* [Economics: Yesterday, Today and Tomorrow], 12 (3A), pp. 246-250. DOI: 10.34670/AR.2022.17.39.028

Keywords

NodeJs, NestJs, TypeScript, AIS "Penalties", integration solution, web technologies, module, software product.

References

1. AWS *Well-Architected Framework*, November 2018. Available at: https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf [Accessed 14/02/2022].
2. Casciaro M., Mammino L. *Node.js Design Patterns. Second Edition: Master best practices to build modular and scalable server-side web applications*. 2016.
3. Guijarro D. et al. (2018) *A Progressive Node.js Framework Kindle Edition*.
4. Kraus D., Obrist T., Hari O. (2019) *Blockchains, Smart Contracts, Decentralised Autonomous Organisations and the Law*. Edward Elgar Publishing.
5. Munoz-Lopez J.E. (2009) Internet Conflict of Laws: A Space of Opportunities for ODR. In: *14 International Law, Revista Colombiana de Derecho Internacional*, pp. 163-190.
6. Resende D. (2018) *Hands-On Microservices with Node.js: Build, test, and deploy robust microservices in JavaScript*.
7. Rohrmann C.A., Rocha F.S., Cunha J. (2015) Some legal aspects of cloud computing contracts. *Journal of International Commercial Law and Technology*, 10 (1).
8. Young A. et al. (2017) *Holowaychuk, Nathan Rajlich. Node.js in Action*, Second Edition.
9. Fox D., Green S. The law on cryptocurrencies. – 2019.
10. Abdullova M.O. "Mixed problem for one fourth-order equation" Materials of the Republican conference "Actual problems of differential equations and their applications" Tashkent December 15-17, 2017