

помощью разработчики получают возможность точно идентифицировать текущее положение пользователей, чтобы адаптировать контент и функционал приложений под их географические координаты. Это открывает путь к персонализированному взаимодействию, повышая удобство.

В приложении «Маршруты по Беларуси» геоданные будут служить основой для построения маршрутов, отображения ключевых точек на карте и предоставления возможности просматривать уже созданные маршрутные схемы. Для получения информации о различных точках на маршруте применяется Radar Core API, который обеспечивает визуализацию карт и доступ к необходимым инструментам работы с геоданными. Адреса преобразуются в координаты с помощью Radar Geocoding API. Отслеживание местоположения пользователя в реальном времени будет достигаться при помощи инструментов платформы .NET MAUI.

Оптимизация работы с геоданными будет достигаться за счёт локального хранения пользовательских маршрутов и применения алгоритмов для фильтрации и обработки данных, что повысит быстродействие приложения и снизит нагрузку на сервер. Дополнительно планируется использовать современные методы кеширования данных и офлайн-режим, что позволяет сохранять работоспособность сервиса даже при слабом интернет-соединении или при его отсутствии.

Доступ к геоданным будет запрашиваться только при необходимости, а пользователи смогут самостоятельно управлять предоставлением данных через настройки приложения. Вся обработка местоположения будет соответствовать актуальным требованиям GDPR и другим региональным стандартам, обеспечивая защиту персональной информации.

Р. Д. Савенко, Е. В. Рафалова
(ГГУ имени Ф. Скорины, Гомель)

ПРИМЕНЕНИЕ БИБЛИОТЕКИ FLUENTVALIDATION ДЛЯ ВАЛИДАЦИИ WEB API НА БАЗЕ ФРЕЙМВОРКА ASP.NET CORE

В современных веб-приложениях, особенно в контексте разработки API, валидация данных является неотъемлемой частью про-

цесса обработки запросов. Она обеспечивает корректность и целостность информации, поступающей от клиентов, а также защищает приложение от потенциальных ошибок и уязвимостей, таких как некорректный формат данных, SQL-инъекции или другие виды атак. Валидация помогает гарантировать, что только допустимые данные попадают в систему, что особенно важно для поддержания стабильности и безопасности приложения.

В ASP.NET Core валидация может быть реализована с использованием атрибутов валидации, или, точнее, атрибутов `DataAnnotations`, которые позволяют указать правила, которым должна соответствовать модель привязки. Этот подход прост в использовании, но имеет ограничения в гибкости, особенно, когда требуется реализовать сложные сценарии. Альтернативой этому подходу является ручная валидация в коде, однако это увеличивает сложность поддержки и может привести к дублированию.

Для решения описанных проблем была разработана библиотека `FluentValidation`, которая предлагает множество преимуществ. Она позволяет создавать сложные и кастомные правила проверки данных, которые сложно реализовать с помощью стандартных атрибутов `DataAnnotations`, обеспечивая высокую гибкость настройки. Благодаря использованию текучего синтаксиса (`fluent-интерфейса`), код валидации становится более понятным и легко читаемым, что значительно упрощает его поддержку. Правила валидации можно выносить в отдельные классы, что способствует их повторному использованию и упрощает процесс тестирования. Библиотека легко интегрируется с ASP.NET Core, автоматически проверяя входящие данные в контроллерах, что сокращает объем ручной работы и ускоряет разработку.

`FluentValidation` также поддерживает многоязычные сообщения об ошибках «из коробки», что упрощает адаптацию приложения для международной аудитории. Кроме того, библиотека оптимизирована для работы с большими объемами данных и поддерживает кэширование валидаторов, что повышает производительность приложения.

Таким образом, применение библиотеки `FluentValidation` в ASP.NET Core позволяет эффективно решать задачи валидации данных, обеспечивая гибкость, читаемость кода и удобство интеграции, что делает ее отличным выбором для разработки современных веб-API.