

Некоторые API, такие как SOAP или XML-RPC, налагают на разработчиков строгие рамки. Но REST API можно разрабатывать практически на любом языке программирования и поддерживать различные форматы данных. Единственное требование состоит в том, чтобы они соответствовали следующим шести принципам проектирования REST, также известным как архитектурные ограничения:

- единый интерфейс;
- клиент-серверная архитектура;
- отсутствие состояний;
- кэшируемость;
- многоуровневая архитектура;
- код по запросу (необязательно).

В ASP.NETRESTful приложения создаются на базе стандартного шаблона Web-API. API-интерфейсы REST взаимодействуют через HTTP-запросы для выполнения стандартных функций базы данных, таких как создание, чтение, обновление и удаление записей (также известных как CRUD) в ресурсе. Например, REST API будет использовать запрос GET для извлечения записи, запрос POST для ее создания, запрос PUT для обновления записи и запрос DELETE для ее удаления. Все методы HTTP могут использоваться в вызовах API. Хорошо спроектированный REST API похож на веб-сайт, работающий в веб-браузере со встроенными функциями HTTP.

А. В. Сапанович

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

РАЗРАБОТКА ASP.NET-ПРИЛОЖЕНИЯ С ЗАЛОЖЕННЫМИ ОШИБКАМИ

Развитие информационных технологий привело к увеличению важности такого этапа разработки приложений как тестирование. Это очень важный и необходимый инструмент, который можно и нужно использовать для минимизации количества ошибок в приложении.

По этой причине было разработано приложения с заложенными ошибками для отбора кандидатов в тестировщики. Данное приложение призвано облегчить и улучшить отбор потенциальных сотрудников в IT компании путем автоматизации и унификации отборочных заданий.

Приложение представляет собой клиент-серверное web-приложение, представляющее собой систему обратной связи для сотрудников компании, в функционал которого входит проведение опросов сотрудников на различные темы, касающиеся условий труда, создание и обработка заявок на закупку расходных материалов, техники, рассылка уведомлений на электронную почту и мессенджеры и т. д.

Приложение разработано на фреймворке ASP.NET Core на языке C#, и состоит из back-end и front-end частей, отвечающих соответственно за обработку операций на сервере и их визуализацию.

Серверная часть предоставляет REST API для работы с клиентской частью, а также отвечает за работу с базой данных. Можно выделить два основных компонента: слой доступа к данным и слой бизнес-логики. Первый используется для предоставления доступа и обработки данных, и, главным образом, отвечает за подключение к БД и обработки CRUD-операций на низком уровне. Второй, в свою очередь, занят маршрутизацией HTTP-запросов front-end части приложения, валидацией и обработкой полученных данных.

Разделение данных приложения и управляющей логики позволяет модифицировать эти компонента независимо друг от друга, что облегчает разработку, обслуживание и поддержку проекта. В частности, за счет такого разделения увеличивается возможность повторного использования кода.

Д. А. Хвесюк

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

РАЗРАБОТКА ДИАГРАММЫ ПРЕЦЕДЕНТОВ ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АДАПТИВНОГО ПРОЦЕССА ИЗУЧЕНИЯ ТЕХНОЛОГИИ MACHINE LEARNING

Диаграмма вариантов использования в UML – диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Функциональные возможности проекта: изучение терминов, курсов и других учебных материалов, отслеживание прогресса, проверка знаний посредством теоретических и практических заданий.