

УДК 004.457

**ПОДХОДЫ К ПРОЕКТИРОВАНИЮ АРХИТЕКТУРЫ СЕРВЕРНОЙ ЧАСТИ НА ПРИМЕРЕ
ВЕБ-ПРИЛОЖЕНИЯ «ВЕБ-ПОРТАЛ ЛЮБИТЕЛЬСКОГО ГАНДБОЛА РЕСПУБЛИКИ БЕЛАРУСЬ»****В.О. ТРАПЕЗНИКОВ***(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

Рассматриваются технологии, используемые при реализации серверной части веб-приложений. Анализируется фреймворк ADO.NET Entity Framework, подход CodeFirst. Фреймворк ADO.NET Entity Framework позволяет создать базу данных, содержащую всю информацию и функционал, необходимый для комфортной работы веб-приложения; подход CodeFirst предоставляет возможность сгенерировать базу данных с нуля лишь на основе написанных классов моделей.

Веб-приложение состоит из клиентской и серверной частей, тем самым реализуя технологию «клиент – сервер». Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует веб-страницу и отправляет её клиенту по сети с использованием протокола HTTP.

Серверная часть веб-приложения – это программа или скрипт на сервере, обрабатывающая запросы пользователя. При каждом переходе пользователя по ссылке браузер отправляет запрос к серверу. Сервер обрабатывает этот запрос, вызывая некоторый скрипт, который формирует веб-страницу, описанную языком HTML, и отправляет клиенту по сети. Браузер тут же отображает полученный результат в виде очередной веб-страницы.

Также практически ни одно веб-приложение не может обойтись без использования базы данных. База данных (БД, или система управления базами данных, СУБД) – программное обеспечение на сервере, занимающееся хранением данных и их выдачей в нужный момент. В данном случае в базе данных хранится информация о пользователях, игроках, результатах проведения соревнований, командах. База данных располагается на сервере. Серверная часть веб-приложения обращается к базе данных по необходимости, извлекая данные, которые необходимы для формирования страницы, запрошенной пользователем.

Средства решения задачи. Рассмотрим основные средства и технологии реализации серверной части веб-приложения «Веб-портал любительского гандбола Республики Беларусь».

В работе был использован ASP.NET MVC Framework – фреймворк для создания веб-приложений, который реализует шаблон Model-view-controller. Данный фреймворк добавлен Microsoft в ASP.NET. Платформа ASP.NET MVC базируется на взаимодействии трех компонентов: контроллера, модели и представления. Контроллер принимает запросы, обрабатывает пользовательский ввод, взаимодействует с моделью и представлением и возвращает пользователю результат обработки запроса.

Модель представляет слой, описывающий логику организации данных в приложении. Представление получает данные из контроллера и генерирует элементы пользовательского интерфейса для отображения информации. При обработке запросов фреймворк ASP.NET MVC опирается на систему маршрутизации, которая сопоставляет все входящие запросы с определенными в системе маршрутами, которые указывают, какой контроллер и метод должен обработать данный запрос. Встроенный маршрут по умолчанию предполагает трехзвенную структуру: контроллер/действие/параметр [1; 2].

В качестве IoC-контейнера будет использоваться Ninject. Собственно IoC (Inversion of Control (инверсия управления)) – это некий абстрактный принцип, набор рекомендаций для написания слабо связанного кода, суть которого в том, что каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов. Dependency Injection (внедрение зависимостей) – это одна из реализаций этого принципа, а IoC-контейнер – это фреймворк, который позволит упростить и автоматизировать написание кода с использованием данного подхода настолько, насколько это возможно [1; 2].

При реализации архитектуры приложения будет использоваться «луковая» архитектура. Onion-архитектура представляет собой разделение приложения на уровни. Причем есть один независимый уровень, который находится в центре архитектуры. От этого уровня зависит второй уровень, от второго – третий, и так далее. То есть получается, что вокруг первого независимого уровня наслаивается второй – зависимый. Вокруг второго наслаивается третий, который также может зависеть и от первого. Образно это может быть выражено в виде луковицы, в которой также есть сердцевина, вокруг которой наслаиваются все остальные слои, вплоть до шелухи.

Количество уровней может отличаться, но в центре всегда находится модель домена (Domain Model), то есть те классы моделей, которые используются в приложении и объекты которых хранятся в базе дан-

ных. Первый уровень вокруг модели домена образуют интерфейсы, которые управляют работой с моделью домена. Обычно это интерфейсы репозитория, через которые мы взаимодействуем с базой данных.

Внешний уровень представляет такие компоненты, которые часто изменяются. Обычно внешний уровень образуют пользовательский интерфейс, тесты, какие-то вспомогательные классы инфраструктуры приложения. К этому уровню также относятся конкретные реализации интерфейсов, объявленных на нижележащих уровнях. Например, реализация интерфейса репозитория, который объявлен на уровне Domain Services. Вообще все внутренние уровни, которые можно объединить в Application Core, определяют только интерфейсы, а конкретная реализация этих интерфейсов располагается на внешнем уровне.

Также стоит отметить, что все внешние хранилища, как базы данных, файлы, внешние веб-сервисы, от которых мы можем получать данные, – все это является внешним по отношению к архитектуре [1; 2].

Для работы с базой данных будет использоваться фреймворк ADO.NET Entity Framework, подход CodeFirst. ADO.NET Entity Framework (EF) – объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL. Для облегчения построения web-решений используется как ADO.NET Data Services (Astoria), так и связка из Windows Communication Foundation и Windows Presentation Foundation, позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования MVC, MVP или MVVM [1; 2].

При проектировании приложений с подходом CodeFirst, сначала создаются классы модели данных, не обращая никакого внимания на Entity Framework. После того как нам понадобилось работать с базой данных, используем различные инструменты, которые проецируют структуру базы данных из созданной модели классов. После этого можем вернуться к этой модели в коде и, например, изменить её. Эти изменения затем можно будет отразить в базе данных, используя все те же инструменты.

Методы решения задач. В соответствии с выполняемыми функциями, веб-приложение можно разбить на несколько модулей.

1 *Модуль работы с аккаунтом пользователя реализует функции:*

- а) регистрация нового пользователя;
- б) вход в аккаунт;
- в) редактирование персональных данных;
- г) подтверждение email;
- д) восстановление пароля.

2 *Модуль работы с информацией о тренерах реализует функции:*

- а) создание нового тренера;
- б) удаление тренера;
- в) редактирование информации о тренере.

3 *Модуль работы с информацией о комментариях реализует функции:*

- а) создание комментариев;
- б) удаление комментариев.

4 *Модуль работы с информацией об играх реализует функции:*

- а) создание результатов игры;
- б) удаление результатов игры;
- в) редактирование информации об игре.

5 *Модуль работы с информацией о спортивных мероприятиях реализует функции:*

- а) регистрация нового мероприятия;
- б) редактирование мероприятия;
- в) удаление мероприятия;
- г) добавление результатов проведения мероприятия;
- д) просмотр результатов проведения спортивного мероприятия.

6 *Модуль работы с информацией о новостях реализует функции:*

- а) создание новостей;
- б) удаление новостей;
- в) редактирование новостей.

7 *Модуль работы с информацией об игроках реализует функции:*

- а) создание нового игрока;
- б) удаление игрока;
- в) редактирование информации об игроке;
- г) поиск игрока.

8 Модуль работы с информацией о судьях реализует функции:

- а) создание нового судьи;
- б) удаление судьи;
- в) редактирование информации о судье.

9 Модуль работы с информацией о командах реализует функции:

- а) создание новой команды;
- б) удаление команды;
- в) редактирование информации о команде;
- г) просмотр информации о команде.

Основной функционал серверной части содержится в следующих классах:

- 1 AccountController – класс контроллера для работы с пользовательскими аккаунтами.
- 2 CoachController – класс контроллера для работы с информацией о тренерах.
- 3 CommentController – класс контроллера для работы с комментариями.
- 4 EventController – класс контроллера для работы с информацией о спортивных мероприятиях.
- 5 GameController – класс контроллера для работы с информацией о прошедших играх.
- 6 HomeController – класс контроллера для работы с отображением информации на главной странице.
- 7 NewsController – класс контроллера для работы с информацией о новостях.
- 8 PlayerController – класс контроллера для работы с информацией об игроках.
- 9 RefereeController – класс контроллера для работы с информацией о судьях.
- 10 ResultController – класс контроллера для работы с информацией о результатах проведения спортивных мероприятий.
- 11 TeamController – класс контроллера для работы с информацией о командах.

Заключение. Использование модели MVC позволило четко разграничить функционал серверной части, не смешивая его составные части, что привело к простоте понимания и последующего изменения кода.

Фреймворк ADO.NET Entity Framework позволил создать базу данных, содержащую всю информацию и функционал, необходимый для комфортной работы веб-приложения, без вникания в тонкости SQL.

Подход CodeFirst дал возможность сгенерировать базу данных с нуля лишь на основе написанных классов моделей.

Использование IoC-контейнера Ninject сделал веб-приложение гибким для изменений. Надо лишь реализовать интерфейс нужным нам образом на верхнем слое приложения, не углубляясь в нижние слои.

ЛИТЕРАТУРА

- 1. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/>. Дата доступа: 27.09.2016.
- 2. Metanit – сайт о программировании [Электронный ресурс]. – Режим доступа: <http://metanit.com>. – Дата доступа: 27.09.2016.