

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
Филиал федерального государственного автономного образовательного учреждения
высшего образования «Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)
Отделение ИКС

Технология создания и функционирования базы данных «Книги»

Текст программы

(версия 1.1.0 от 05.12.2018)

Приложение разработано студентом группы ИВТ2-Б15 Туголуковым Н. П.

E-mail: tugolukov40@yandex.ru

Содержание

Содержание	2
1. Добавление, чтение, изменение, удаление информации (CRUD)	3
2. Описание моделей для CRUD	3
Модель книги	3
Модель персоны	3
Модель издательства	3
Модель сферы применения	3
3. Сервисы для работы с данными в БД	3
4. Дополнительный функционал	3
5. Описание моделей в БД	4
6. Razor-представления	4
7. Библиотеки JS	4
8. JS-скрипты	4
9. CSS	4
10. Файл конфигурации	4
11. Файл контекста БД	5
12. Миграции	5
13. Файл StartUp.cs	5
14. Файл Program.cs	5
Лист изменений	6

1. Добавление, чтение, изменение, удаление информации (CRUD)

Книги – `elibrary/Library.API/Controllers/BooksController.cs`

Персоны – `elibrary/Library.API/Controllers/PersonsController.cs`

Издательства – `elibrary/Library.API/Controllers/PublishingsController.cs`

Серия применения – `elibrary/Library.API/Controllers/TechnologiesController.cs`

2. Описание моделей для CRUD

Модель книги

Чтение книги – `elibrary/Library.Domain/Models/Book/BookModel.cs`

Создание книги – `elibrary/Library.Domain/Models/Book/CreateBookModel.cs`

Изменение книги – `elibrary/Library.Domain/Models/Book/UpdateBookModel.cs`

Модель персоны

Создание автора – `elibrary/Library.Domain/Models/Person/AuthorModel.cs`

Создание читателя – `elibrary/Library.Domain/Models/Person/ReaderModel.cs`

Создание персоны – `elibrary/Library.Domain/Models/Person/CreatePersonModel.cs`

Изменение персоны – `elibrary/Library.Domain/Models/Person/UpdatePersonModel.cs`

Модель издательства

Создание издательства –

`elibrary/Library.Domain/Models/Publishing/CreatePublishingModel.cs`

Изменение издательства –

`elibrary/Library.Domain/Models/Publishing/UpdatePublishingModel.cs`

Модель сферы применения

Создание издательства –

`elibrary/Library.Domain/Models/Technology/CreateTechnologyModel.cs`

Изменение издательства –

`elibrary/Library.Domain/Models/Technology/UpdateTechnologyModel.cs`

3. Сервисы для работы с данными в БД

Работа с таблицей «Books» – `elibrary/Library.Domain/Services/BooksService.cs`

Работа с таблицей «Persons» – `elibrary/Library.Domain/Services/PersonsService.cs`

Работа с таблицей «Publishings» – `elibrary/Library.Domain/Services/PublishingsService.cs`

Работа с таблицей «Technologies» – `elibrary/Library.Domain/Services/TechnologiesService.cs`

4. Дополнительный функционал

Работа с ролями персон – `elibrary/Library.Domain/Services/RolesService.cs`

Поиск книг – `elibrary/Library.Domain/Services/SearchService.cs`

5. Описание моделей в БД

Книга – `elibrary/Library.Database/Models/Book.cs`

Персона – `elibrary/Library.Database/Models/Person.cs`

Издательство – `elibrary/Library.Database/Models/Publishing.cs`

Серия применения – `elibrary/Library.Database/Models/Technology.cs`

Связная таблица Книга-Читатель – `elibrary/Library.Database/Models/BookReader.cs`

Вспомогательная таблица Роль – `elibrary/Library.Database/Models/Options/Role.cs`

Вспомогательная таблица Типы обложки –
`elibrary/Library.Database/Models/Options/Cover.cs`

Вспомогательная таблица Формат – `elibrary/Library.Database/Models/Options/Format.cs`

Вспомогательная таблица Языки программирования –
`elibrary/Library.Database/Models/Options/Language.cs`

6. Razor-представления

Razor-представления содержатся в папке `elibrary/Library.API/Views`. Они именованы согласно названиям методов в контроллерах и напрямую обращаются к ним.

7. Библиотеки JS

jQuery (легкий доступ к любому элементу, обращаться к атрибутам и содержимому) - `jquery.js`

8. JS-скрипты

Скрипты для вывода модальных окон находятся в Razor-представлениях, в которых модальные окна вызываются.

9. CSS

`elibrary/Library.API/wwwroot/lib/bootstrap` – скомпилированные файлы CSS-фреймворка Bootstrap.

`elibrary/Library.API/wwwroot/css` – файлы стилей, которые задаются разработчиком.

10. Файл конфигурации

`elibrary/Library.API/appsettings.json` – файл, содержащий параметры конфигурации сборки приложения, параметры подключения к БД.

11. Файл контекста БД

elibrary/Library.Database/DatabaseContext.cs – файл, содержащий в себе методы для создания миграций, содержащий в себе структуру БД.

12. Миграции

elibrary/Library.API/Migrations/ – директория, содержащая в себе файлы миграции, которые используются для развертывания базы данных. При разработке приложения на платформе .NET Core используется паттерн проектирования CodeFirst, подразумевающий создание БД на основе написанных моделей в решении .NET Core. Таким образом разработчики получают возможность администрировать базу данных напрямую из кода приложения.

13. Файл StartUp.cs

elibrary/Library.API/Startup.cs – конфигурирование сервисов для запуска приложения.

14. Файл Program.cs

elibrary/Library.API/Program.cs – запуск приложения.

Документация

[Общее описание технологии](#)

[Руководство пользователя](#)

[Руководство администратора](#)

[Руководство оператора](#)

[Описание программы](#)

[Описание организации информационной базы](#)

[Текст программы](#)

Лист изменений

№ п/п	Дата внесения изменений	Номер раздела	Содержание внесенного изменения	Версия документа после изменений
1	05.12.2018	Все разделы	Обновление	1.1.0