

第07期-使用ABP实战CRUD增删改查

2020年7月19日 8:48

创建 Razor 页面

```
@page
@model BookStore.Web.Pages.Books.IndexModel
@{
    ViewData["Title"] = "Index";
}
```

```
<h1>Book</h1>
```

创建菜单

打开 Menus 文件夹中的 BookStoreMenuContributor 类，在ConfigureMainMenuAsync方法的底部添加如下代码：

```
context.Menu.AddItem(new ApplicationMenuItem("BookStore", 1["Menu:BookStore"]))
    .AddItem(new ApplicationMenuItem("BookStore.Book", 1["Menu:Book"], url:
"/Books"));
```

本地化菜单

本地化文本位于BookStore.Domain.Shared项目的Localization/BookStore文件夹下：

```
{
    "culture": "en",
    "texts": {
        "Menu:Home": "Home",
        "Welcome": "Welcome",
        "LongWelcomeMessage": "Welcome to the application. This is a startup
project based on the ABP framework. For more information, visit abp.io.",

        "BookStore": "Zero Book Store",
        "Books": "Book List",
        "Menu:BookStore": "Book Store",
        "Menu:Book": "Book",
        "Actions": "Actions",
        "Edit": "Edit",
        "PublishDate": "Publish date",
        "NewBook": "New book",
        "Name": "Name",
        "Type": "Type",
```

```

        "Price": "Price",
        "CreationTime": "Creation time",
        "AreYouSureToDelete": "Are you sure you want to delete this item?",
        "BookDeletionConfirmationMessage": "Are you sure to delete the book {0}?",
        "SuccessfullyDeleted": "Successfully deleted"
    }
}

{
    "culture": "zh-Hans",
    "texts": {
        "Menu:Home": "首页",
        "Welcome": "欢迎",
        "LongWelcomeMessage": "欢迎来到该应用程序。这是一个基于ABP框架的启动项目。有关更多信息，请访问 abp.io。",

        "BookStore": "零度书城",
        "Books": "图书列表",
        "Menu:BookStore": "书店",
        "Menu:Book": "图书",
        "Actions": "操作",
        "Edit": "编辑",
        "PublishDate": "出版日期",
        "NewBook": "创建图书",
        "Name": "名称",
        "Type": "分类",
        "Price": "价格",
        "CreationTime": "创建时间",
        "AreYouSureToDelete": "你确定要删除这个项目吗？",
        "BookDeletionConfirmationMessage": "您确定要删除图书 {0}？",
        "SuccessfullyDeleted": "删除成功！"
    }
}

```

实现列表展示

我们将使用 datatables.net JQuery 插件来显示页面上的表格列表。

```

@page
@model BookStore.Web.Pages.Books.IndexModel
@using Microsoft.AspNetCore.Mvc.Localization
@using BookStore.Localization
@inject IHtmlLocalizer<BookStoreResource> L
@section scripts
{
    <abp-script src="/Pages/Books/index.js" />
}
<abp-card>
    <abp-card-header>

```

```

        <h2>@L["Book"]</h2>
    </abp-card-header>
    <abp-card-body>
        <abp-table striped-rows="true" id="BookTable">
            <thead>
                <tr>
                    <th>@L["Name"]</th>
                    <th>@L["Type"]</th>
                    <th>@L["PublishDate"]</th>
                    <th>@L["Price"]</th>
                    <th>@L["CreationTime"]</th>
                </tr>
            </thead>
        </abp-table>
    </abp-card-body>
</abp-card>

$(function () {
    var dataTable =
$( '#BookTable' ).DataTable( abp.libs.datatables.normalizeConfiguration({
    ajax:
abp.libs.datatables.createAjax( bookstore.books.book.getList ),
    columnDefs: [
        { data: "name" },
        { data: "type" },
        { data: "publishDate" },
        { data: "price" },
        { data: "creationTime" }
    ]
    }
    ));
});

```

实现创建功能

前端页面代码

```

@page
@using Volo.Abp.AspNetCore.Mvc.UI.Bootstrap.TagHelpers.Modal
@using Microsoft.AspNetCore.Mvc.Localization
@using BookStore.Localization
@inject IHtmlLocalizer<BookStoreResource> L
@model BookStore.Web.Pages.Books.CreateModalModel
@{
    Layout = null;
}
<abp-dynamic-form abp-model="Book" data-ajaxForm="true" asp-
page="/Books/CreateModal">
    <abp-modal>
        <abp-modal-header title="@L["NewBook"].Value"></abp-modal-header>
        <abp-modal-body>
            <abp-form-content />

```

```

        </abp-modal-body>
        <abp-modal-footer
buttons="@ (AbpModalButtons.Cancel | AbpModalButtons.Save)"></abp-modal-footer>
    </abp-modal>
</abp-dynamic-form>

```

页面后台代码

```

public class CreateModalModel : BookStorePageModel
{
    [BindProperty]
    public CreateUpdateBookDto Book { get; set; }

    private readonly IBookAppService _bookAppService;

    public CreateModalModel(IBookAppService bookAppService)
    {
        _bookAppService = bookAppService;
    }

    public async Task<IActionResult> OnPostAsync()
    {
        await _bookAppService.CreateAsync(Book);
        return NoContent();
    }
}

```

在首页添加「创建」按钮

```

<abp-card-header>
    <abp-row>
        <abp-column size-md="_6">
            <h2>@L["Books"]</h2>
        </abp-column>
        <abp-column size-md="_6" class="text-right">
            <abp-button id="NewBookButton"
                text="@L["NewBook"].Value"
                icon="plus"
                button-type="Primary" />
        </abp-column>
    </abp-row>
</abp-card-header>

```

添加弹框脚本到 index.js 脚本文件

```

var createModal = new abp.ModalManager(abp.appPath + 'Books/CreateModal');

createModal.onResult(function () {

```

```

        dataTable.ajax.reload();
    });

$('#NewBookButton').click(function (e) {
    e.preventDefault();
    createModal.open();
});

```

实现编辑功能

前端页面代码

```

@page
@using BookStore.Web.Pages.Books
@using Volo.Abp.AspNetCore.Mvc.UI.Bootstrap.TagHelpers.Modal
@using Microsoft.AspNetCore.Mvc.Localization
@using BookStore.Localization
@inject IHtmlLocalizer<BookStoreResource> L
@model EditModalModel
@{
    Layout = null;
}
<abp-dynamic-form abp-model="Book" data-ajaxForm="true" asp-
page="/Books/EditModal">
    <abp-modal>
        <abp-modal-header title="@L["Update"].Value"></abp-modal-header>
        <abp-modal-body>
            <abp-input asp-for="Id" />
            <abp-form-content />
        </abp-modal-body>
        <abp-modal-footer
buttons="@(<AbpModalButtons.Cancel | AbpModalButtons.Save>)"></abp-modal-footer>
    </abp-modal>
</abp-dynamic-form>

```

页面后台代码

```

public class EditModalModel: BookStorePageModel
{
    [HiddenInput]
    [BindProperty(SupportsGet = true)]
    public Guid Id { get; set; }

    [BindProperty]
    public CreateUpdateBookDto Book { get; set; }

    private readonly IBookAppService _bookAppService;

```

```

public EditModalModel(IBookAppService bookAppService)
{
    _bookAppService = bookAppService;
}

public async Task OnGetAsync()
{
    var bookDto = await _bookAppService.GetAsync(Id);
    Book = ObjectMapper.Map<BookDto, CreateUpdateBookDto>(bookDto);
}

public async Task<IActionResult> OnPostAsync()
{
    await _bookAppService.UpdateAsync(Id, Book);
    return NoContent();
}
}

```

BookDto 到 CreateUpdateBookDto 对象映射

```

public class BookStoreWebAutoMapperProfile : Profile
{
    public BookStoreWebAutoMapperProfile()
    {
        //Define your AutoMapper configuration here for the Web project.

        CreateMap<BookDto, CreateUpdateBookDto>();
    }
}

```

为表格添加「操作」下拉菜单

```

<abp-table striped-rows="true" id="BookTable">
    <thead>
        <tr>
            <th>@L["Actions"]</th>
            <th>@L["Name"]</th>
            <th>@L["Type"]</th>
            <th>@L["PublishDate"]</th>
            <th>@L["Price"]</th>
            <th>@L["CreationTime"]</th>
        </tr>
    </thead>
</abp-table>

```

添加脚本文件

```

$(function () {

```

```

var l = abp.localization.getResource('BookStore');

var createModal = new abp.ModalManager(abp.appPath + 'Books/CreateModal');
var editModal = new abp.ModalManager(abp.appPath + 'Books/EditModal');

var dataTable =
$('#BookTable').DataTable(abp.libs.datatables.normalizeConfiguration({
    processing: true,
    serverSide: true,
    paging: true,
    searching: false,
    autoWidth: false,
    scrollCollapse: true,
    order: [[1, "asc"]],
    ajax:
abp.libs.datatables.createAjax(bookStore.books.book.getList),
    columnDefs: [
        {
            rowAction: {
                items: [
                    {
                        text: l('Edit'),
                        action: function
(data) {
                            editModal.
open({ id: data.record.id });
                        }
                    }
                ]
            },
            { data: "name" },
            { data: "type" },
            { data: "publishDate" },
            { data: "price" },
            { data: "creationTime" }
        ]
    })
));

createModal.onResult(function () {
    dataTable.ajax.reload();
});

editModal.onResult(function () {
    dataTable.ajax.reload();
});

$('#NewBookButton').click(function (e) {
    e.preventDefault();
    createModal.open();
});

```

```
});
```

实现删除功能

添加脚本文件

打开 Pages/BOOKS/index.js 文件,在 rowAction 的 items 下新增一项:

```
{
    text: l('Delete'),
    confirmMessage: function (data) {
        return l('BookDeletionConfirmationMessage', data.record.name);
    },
    action: function (data) {
        bookStore.books.book
            .delete(data.record.id)
            .then(function () {
                abp.notify.info(l('SuccessfullyDeleted'));
                dataTable.ajax.reload();
            });
    }
}
```

添加删除提示多语言支持

```
"BookDeletionConfirmationMessage": "Are you sure to delete the book {0}?",
"SuccessfullyDeleted": "Successfully deleted"
```

修改网站 LOGO 品牌

```
[Dependency(ReplaceServices = true)]
public class BookStoreBrandingProvider : DefaultBrandingProvider
{
    private readonly IStringLocalizer<BookStoreResource> _localizer;

    public BookStoreBrandingProvider(IStringLocalizer<BookStoreResource>
localizer)
    {
        _localizer = localizer;
    }

    public override string AppName => _localizer.GetString("BookStore");
}
```