**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

**ОТЧЁТ**

**по лабораторной работе №4**

**по дисциплине «Программирование в среде .NET»**

**Тема: “Разработка программного интерфейса для взаимодействия с приложением”**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6305 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Буракаев Д. А. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Пешехонов К. А. |

Санкт-Петербург

2020

**Содержание**

[**Цель работы** 3](#_Toc38652795)

[**Задание** 3](#_Toc38652796)

[**Ход работы** 4](#_Toc38652797)

[**Модели** 4](#_Toc38652798)

[**Контроллеры** 5](#_Toc38652799)

[**Файлы разметки веб-страниц** 7](#_Toc38652800)

[**Вывод** 9](#_Toc38652801)

# **Цель работы**

Целью данной лабораторной работы является реализация интерфейса для взаимодействия с разрабатываемым приложением.

# **Задание**

1. Ознакомиться с принципом работы Swagger и AutoMapper.

2. Реализовать интерфейс для взаимодействия с приложением.

# **Ход работы**

## **Модели**

*QuestModel.cs*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace SimpleWebApi2.Ninject.Models

{

public class Quest

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public int PlayersLimit { get; set; }

public double Price { get; set; }

public int Duration { get; set; }

}

}

*QuestListViewModel.cs*

using BLL.DTO;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace SimpleWebApi2.Ninject.Models

{

public class QuestsListViewModel

{

public IEnumerable<QuestDTO> Quests { get; set; }

public string Description { get; set; }

public SelectList Players { get; set; }

public SelectList Duration { get; set; }

public SelectList Price { get; set; }

}

}

*ReservationListModel.cs*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace SimpleWebApi2.Ninject.Models

{

public class ReservationViewModel

{

public int Id { get; set; }

public QuestsListViewModel Quest { get; set; }

public DateTime DateAndTime { get; set; }

}

}

*ReservModel.cs*

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace SimpleWebApi2.Ninject.Models

{

public class ReservModel

{

[Required]

public int QuestId { get; set; }

[Required]

public string Date { get; set; }

[Required]

public string Time { get; set; }

public string Code { get; set; }

}

}

## **Контроллеры**

*QuestController.cs*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

using AutoMapper;

using BLL.DTO;

using BLL.Interfaces;

using SimpleWebApi2.Ninject.Models;

namespace SimpleWebApi2.Ninject.Controllers

{

public class QuestsController : ApiController

{

private readonly IQuestService questService;

private readonly IReservationService reservationService;

public QuestsController(IQuestService questService, IReservationService reservationService)

{

this.questService = questService;

this.reservationService = reservationService;

}

[HttpGet]

// GET api/quests

public IEnumerable<Quest> GetQuests()

{

return Mapper.Map<IEnumerable<QuestDTO>, IEnumerable<Quest>>(questService.GetAll());

}

[HttpGet]

// GET api/quests/5

public Quest Get(int id)

{

return Mapper.Map<QuestDTO, Quest>(questService.Find(id));

}

[HttpPost]

[Route("api/quests/reserv")]

public IHttpActionResult Reserv(ReservModel model)

{

if (!ModelState.IsValid)

return BadRequest("Invalid data");

var quest = questService.GetAll().Where(x => x.Id == model.QuestId).FirstOrDefault();

double cost = quest.Price;

if (!String.IsNullOrEmpty(model.Code))

if (reservationService.VerifyCertificateCode(model.Code))

cost = 0;

else return BadRequest("Invalid code");

reservationService.Add(new ReservDTO { Quest = quest, DateTimeSections = model.Date + " " + model.Time, Cost = cost });

return Ok($"Quest \"{quest.Name}\" reserved for {model.Date} on {model.Time}\nCost: {cost} grn");

}

public IEnumerable<Quest> Get(int? players, int? duration, int? price)

{

IEnumerable<QuestDTO> quests = questService.GetAll();

if (players != null)

{

quests = quests.Where(x => x.PlayersLimit <= players);

}

if (duration != null)

{

quests = quests.Where(x => x.Duration <= duration);

}

if (price != null)

{

quests = quests.Where(x => x.Price <= price);

}

return Mapper.Map<IEnumerable<QuestDTO>, IEnumerable<Quest>>(quests);

}

}

}

*HomeController.cs*

using BLL.Interfaces;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace SimpleWebApi2.Ninject.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

ViewBag.Title = "Home Page";

return View();

}

}

}

## **Файлы разметки веб-страниц**

*index.html*

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width" />

<title>QuestRoom</title>

<link rel="stylesheet" type="text/css" href="/Client/Files/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/Client/Files/Site.css" />

<script type="text/javascript" src="/Client/Files/jquery-1.10.2.min.js"></script>

<script type="text/javascript" src="/Client/Files/bootstrap.js"></script>

<script type="text/javascript" src="/Client/Files/modernizr-2.6.2.js"></script>

<script type="text/javascript" src="/Client/Files/respond.min.js"></script>

<script type="text/javascript" src="/Client/Files/domain.js"></script>

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

</div>

</div>

<div class="container body-content">

<div class="col-md-8 col-md-offset-2" style="margin-top: 20px" id="tableBlock"></div>

<!-- Modal window-->

<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">

</div>

<!--End modal window-->

</div>

<script type="text/javascript">

$(document).ready(function () {

GetAllQuests();

});

</script>

</body>

</html>

*reserv.html*

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Reservation</title>

<link rel="stylesheet" type="text/css" href="/Client/Files/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/Client/Files/Site.css" />

<script type="text/javascript" src="/Client/Files/jquery-1.10.2.min.js"></script>

<script type="text/javascript" src="/Client/Files/bootstrap.js"></script>

<script type="text/javascript" src="/Client/Files/modernizr-2.6.2.js"></script>

<script type="text/javascript" src="/Client/Files/respond.min.js"></script>

<script type="text/javascript" src="/Client/Files/domain.js"></script>

</head>

<body>

<center>

<h3 id="reserv-title"></h3>

<br />

<label>Date</label>

<input type="date" class="form-control" id="inp-date" />

<br />

<label>Time</label>

<input type="time" class="form-control" id="inp-time" />

<br />

<label>Code (optional)</label>

<input type="text" class="form-control" id="inp-code" />

<br />

<button class="btn btn-success" onclick="Reserv()" style="width: 137px">Reserv</button>

<button class="btn btn-default" onclick="location.href = 'index.html'" style="width: 137px">Back</button>

</center>

<script type="text/javascript">

$(document).ready(function () {

LoadReservData();

});

</script>

</body>

</html>

# **Вывод**

В результате выполнения данной лабораторной работы был реализован слой интерфейса для взаимодействия с разрабатываемым приложением.