


Обработка исходящего Договора с физ. лицом

В веб дизайнере ELMA4 (в ELMA3 обычном дизайнере) реализуем Глобальный модуль, где определяем нужную точку расширения, например, **SendableRNMOutContractInd**.

 **Дизайнер**

Сценарии ▾

Поиск

Все сценарии R&M HMS ✕

Глобальный модуль "R&M HMS"

Настройки

Название *

R&M HMS

Пространство имен

HMS.Configuration

Описание

Структура модуля

▼ Корневая папка модуля

▸ Contracts

▾ Diadoc

SendableRNMOutContractSubLic

ReceiveableContract

DiadocIntegration

ReceiveableAdditionalAgreement

SendableRNMOutContractInd

SendableRNMOutContractLic

▸ Products

Реализация:

```
using System;
using EleWise.ELMA.ComponentModel;
using EleWise.ELMA.ConfigurationModel;
using EleWise.ELMA.Documents.Models;
using EleWise.ELMA.CRM.Models;
using EleWise.ELMA.Model.Services;
```

```

using EleWise.ELMA.Services;
using ITino.ELMA.Documents.Diadoc.Components;
using ITino.ELMA.Documents.Diadoc.Exceptions;
using ITino.ELMA.Documents.Diadoc.Managers;
using ITino.ELMA.Documents.Diadoc.Models;
using ITino.ELMA.Documents.Diadoc.Services;

namespace HMS.Configuration
{
    /// <inheritdoc />
    [Component]
    public class SendableRNMOutContractInd : SendableDiadocDocument
    {
        /// <inheritdoc />
        public override bool Enabled
        {
            /// Точка расширения включена и обрабатывается
            get { return true; }
        }

        /// <inheritdoc />
        public override DiadocDocumentType DocumentType
        {
            /// Какой тип документа Диадока обрабатываем
            get { return DiadocDocumentType.Contract; }
        }

        /// <inheritdoc />
        public override Guid TypeUid
        {
            /// К какому типу документа в ELMA привязываем
            get { return InterfaceActivator.UID<RNMOutContractInd>(); }
        }

        public override IContractor GetSender(IDocument doc)
        {
            /// Получить отправителя документа
            [var contract = doc as RNMOutContractInd;
            [return contract != null ? contract.Contract : null;
        }
    }
}

```



```
/// <inheritdoc />
public override EODiadocCanSendStatus CanSendDocument(IDocument doc)
{
    [/] Проверить, можно ли отправить документ в Диадок
    if (doc == null)
        return EODiadocCanSendStatus.DocIsNull;
    var service = Locator.GetServiceNotNull<IDiadocService>();
    var contract = doc as RNMOOutContractInd;
    if (contract != null)
        return service.CanSendDocument(contract.ContractId, contract.Executor);
    return base.CanSendDocument(doc);
}

/// <inheritdoc />
public override ICODiadocDocument Send(IDocument doc, Guid? departmentId, Guid? userId, string
comment = null)
{
    [/] Отправить документ в Диадок на подписание нами
    var contract = doc as RNMOOutContractInd;
    if (contract == null)
        throw new DiadocException("Договор не может быть null");
    var edata = DiadocDocumentManager.Instance.CheckContractors(contract.ContractId,
contract.Executor, null, null);
    var result = edata.Service.SendContract(edata.FormBoxId, edata.ToBoxId, doc, contract.Number,
contract.ContractDate.Value, "ГПХ/Физ.лицо/Самозанятый", null, departmentId, userId, comment);
    return DiadocDocumentManager.Instance.SendDocHistory(result, doc, contract.ContractId,
contract.Executor, contract.Number, contract.ContractDate);
}
}
```

Атрибут [Component] обязателен.

Версия #3

Вербовик Максим Арнольдович создал 7 сентября 2021 08:58:09

Вербовик Максим Арнольдович обновил 4 ноября 2021 09:06:17