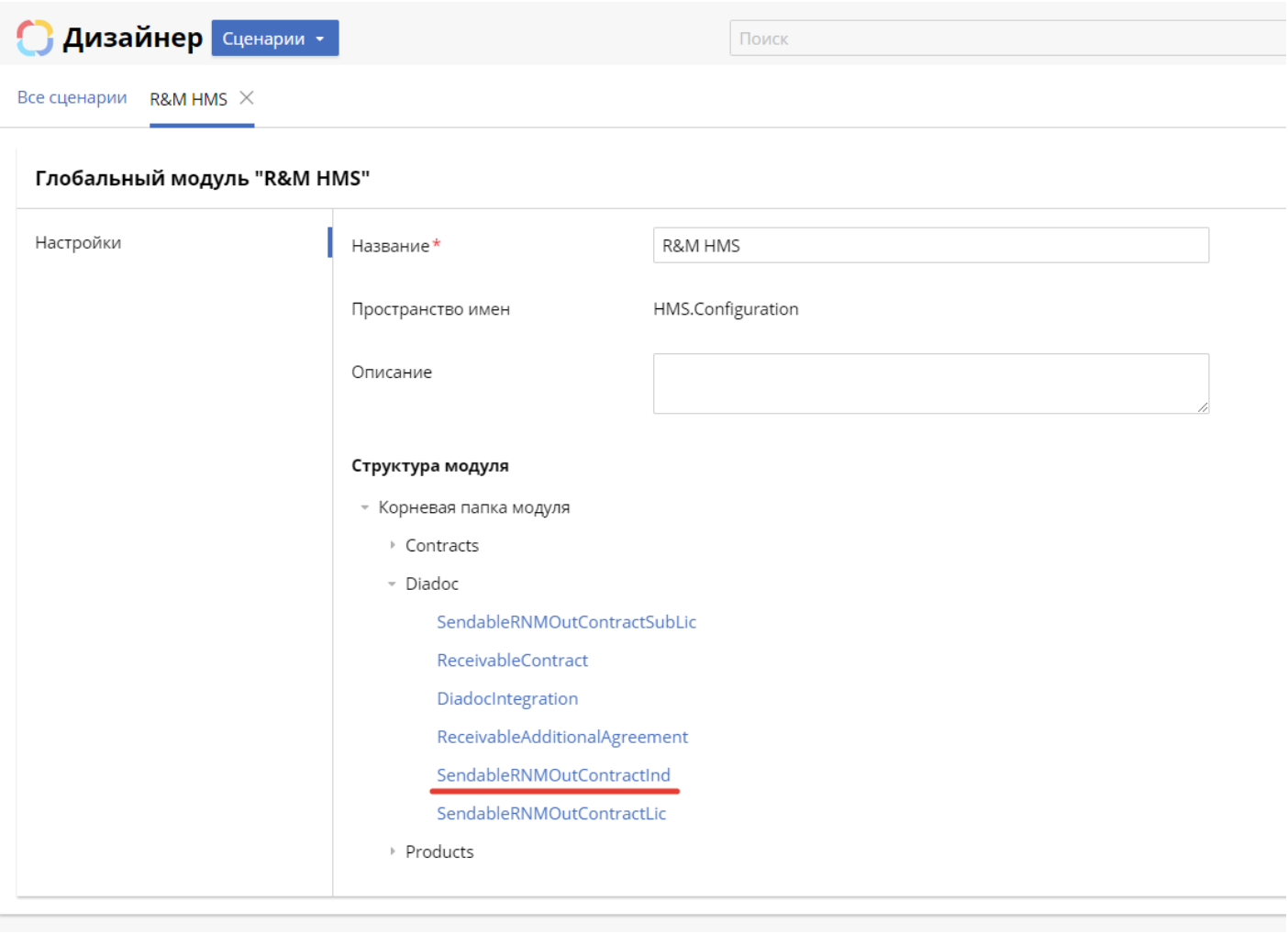
Обработка исходящего Договора с физ. лицом

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



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.Contractor : 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 RNMOutContractInd;
            if (contract != null)
                return service.CanSendDocument(contract.Contractor, contract.Executor);
            return base.CanSendDocument(doc);
        }

        /// <inheritdoc />
        public override ICODiadocDocument Send(IDocument doc, Guid? departmentId, Guid?
userId, string comment = null)
        {
            // Отправить документв Диадок на подписание нами
            var contract = doc as RNMOutContractInd;
            if (contract == null)
                throw new DiadocException("Договор не может быть null");
            var edata = DiadocDocumentManager.Instance.CheckContractors(contract.Contractor,
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.Contractor, contract.Executor, contract.Number, contract.ContractDate);
        }
    }
}
```

> Атрибут [Component] обязателен.

Версия #3
Вербовик Максим Арнольдович создал 2021-09-07 08:58:09 MSK
Вербовик Максим Арнольдович обновил 2021-11-04 09:06:17 MSK