

# Обработка исходящего Договора с физ. лицом

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

The screenshot shows the 'Дизайнер' (Designer) interface with a search bar and a dropdown menu for 'Сценарии' (Scenarios). Below the search bar, there are tabs for 'Все сценарии' (All scenarios) and 'R&M HMS' (selected). The main content area is titled 'Глобальный модуль "R&M HMS"' (Global module "R&M HMS"). On the left, there is a 'Настройки' (Settings) sidebar. The main area displays the following configuration:

- Название\*** (Name): R&M HMS
- Пространство имен** (Namespace): HMS.Configuration
- Описание** (Description): [Empty text area]
- Структура модуля** (Module structure):
  - Корневая папка модуля (Root folder):
    - Contracts
    - Diadoc
      - SendableRNMOContractSubLic
      - ReceivableContract
      - DiadocIntegration
      - ReceivableAdditionalAgreement
      - SendableRNMOContractInd (highlighted with a red underline)
      - SendableRNMOContractLic
    - 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 2021-09-07 08:58:09 MSK

Вербовик Максим Арнольдович обновил 2021-11-04 09:06:17 MSK