Точки расширения

Реализация точек расширения для обработки входящих и исходящих документов для Диадока

- ReceivableDiadocDocument
- SendableDiadocDocument

ReceivableDiadocDocument

Данная точка расширения позволяет настроить поведение обработчика входящих из Диадока документов.

> Наследование необходимо делать именно от класса **ReceivableDiadocDocument**, а не от интерфейса **IReceivableDiadocDocument**.

> namespace ITino.ELMA.Documents.Diadoc.Components

```
/// <summary>
/// Точка расширения для документов, которые можно отправить в Диадок
/// </summary>
public class ReceivableDiadocDocument : IReceivableDiadocDocument
{
/// <summary>
    /// Обработчик включен
    /// </summary>
    public virtual bool Enabled { get; }

    /// <summary>
    /// Тип документа Диадок
    /// </summary>
    public virtual DiadocDocumentType DocumentType { get; }

    /// <summary>
    /// Тип документа в системе
    /// </summary>
    public virtual Guid TypeUid { get; }

    /// <summary>
    /// Типы документа в системе
    /// </summary>
    public virtual IEnumerable<Guid> TypeUids { get; }

    /// <summary>
```

```
    /// Обработать кастомные поля (если есть)
    /// </summary>
    /// <param name="doc">Документ Диадока (справочник в ELMA)</param>
    /// <param name="items">Кастомные поля Диадока</param>
    public virtual void ProcessCustomData(ICODiadocDocument doc, IEnumerable<CustomDataItem>
items);
}
```

Пример реализации: Обработка входящего Договора

SendableDiadocDocument

Данная точка расширения позволяет настроить поведение обработчика исходящих в Диадок документов.

> Наследование необходимо делать именно от класса **SendableDiadocDocument**, а не от интерфейса **ISendableDiadocDocument**.

> namespace ITino.ELMA.Documents.Diadoc.Components

```
/// <summary>
/// Точка расширения для документов, которые можно отправить в Диадок
/// </summary>
public class SendableDiadocDocument : ISendableDiadocDocument
{
    /// <summary>
    /// Обработчик включен
    /// </summary>
    public virtual bool Enabled { get; }

    /// <summary>
    /// Тип документа Диадок
    /// </summary>
    public virtual DiadocDocumentType DocumentType { get; }

    /// <summary>
    /// Тип документа в системе
    /// </summary>
    public virtual Guid TypeUid { get; }

    /// <summary>
    /// Типы документа в системе
    /// </summary>
    public virtual IEnumerable<Guid> TypeUids { get; }

    /// <summary>
```

```
    /// Можно ли отправить документ
    /// </summary>
    /// <param name="doc">Документ</param>
    public virtual EODiadocCanSendStatus CanSendDocument(IDocument doc);

    /// <summary>
    /// Получить отправителя (наша организация)
    /// </summary>
    /// <param name="doc">Документ</param>
    public virtual IContractor GetSender(IDocument doc);

    /// <summary>
    /// Отправить документ в Диадок в Головное подразделение
    /// </summary>
    /// <param name="doc">Документ</param>
    /// <param name="comment">Комментарий (опционально)</param>
    public virtual ICODiadocDocument Send(IDocument doc, string comment = null);

    /// <summary>
    /// Отправить документ в Диадок в подразделении любому с правом подписи
    /// </summary>
    /// <param name="doc">Документ</param>
    /// <param name="departmentId">Подразделение в Диадоке</param>
    /// <param name="comment">Комментарий (опционально)</param>
    public virtual ICODiadocDocument Send(IDocument doc, Guid? departmentId, string comment =
null);

    /// <summary>
    /// Отправить документ в Диадок
    /// </summary>
    /// <param name="doc">Документ</param>
    /// <param name="departmentId">Подразделение в Диадоке</param>
    /// <param name="userId">Сотрудник в Диадоке</param>
    /// <param name="comment">Комментарий (опционально)</param>
    public virtual ICODiadocDocument Send(IDocument doc, Guid? departmentId, Guid? userId,
string comment = null);
}
```

Пример реализации: Обработка исходящего Договора с физ. лицом