

Сервисы

Сервисы для обмена с Диадоком

- [IDiadocService](#)

IDiadocService

Чтобы получить указанный сервис, необходимо реализовать следующий код:

```
var service = Locator.GetServiceNotNull<IDiadocService>();
```

Описание свойств и методов сервиса:

```
namespace ITino.ELMA.Documents.Diadoc.Services
```

```
/// <summary>
/// Сервис работы с Диадок
/// </summary>
public interface IDiadocService
{
    /// <summary>
    /// Сервис инициализирован и авторизован
    /// </summary>
    bool Initialized { get; }

    /// <summary>
    /// Отключить интеграцию
    /// </summary>
    void Shutdown();

    /// <summary>
    /// Ожидание операции по отправке документа в Диадок
    /// </summary>
    bool WaitSendingOps { get; set; }

    /// <summary>
    /// Получить информацию о контрагенте по ИНН
    /// </summary>
    /// <param name="inn"></param>
    /// <param name="kpp"></param>
    /// <returns></returns>
    // ReSharper disable once InconsistentNaming
```

```
IEnumerable<Organization> GetContractorInfoByINN(string inn, string kpp = null);
```

```
/// <summary>
```

```
/// Получить информацию о контрагенте
```

```
/// </summary>
```

```
/// <param name="contractor"></param>
```

```
/// <returns></returns>
```

```
IEnumerable<Organization> GetContractorInfo(IContractor contractor);
```

```
/// <summary>
```

```
/// Получить свою организацию по контрагенту
```

```
/// </summary>
```

```
/// <param name="contractor"></param>
```

```
/// <returns></returns>
```

```
Organization GetOwnContractorInfo(IContractor contractor);
```

```
/// <summary>
```

```
/// Получить контрагента если он есть в активном списке
```

```
/// </summary>
```

```
/// <param name="owner"></param>
```

```
/// <param name="contractor"></param>
```

```
/// <returns></returns>
```

```
Organization GetContractorInList(IContractor owner, IContractor contractor);
```

```
/// <summary>
```

```
/// Получить контрагента по ящику если он есть в списке
```

```
/// </summary>
```

```
/// <param name="owner"></param>
```

```
/// <param name="boxId"></param>
```

```
/// <returns></returns>
```

```
IContractor GetContractorInList(IContractor owner, string boxId);
```

```
/// <summary>
```

```
/// Получить контрагента и статус списка для нашей организации
```

```
/// </summary>
```

```
/// <param name="org"></param>
```

```
/// <param name="contractor"></param>
```

```
/// <returns></returns>
```

```
Counteragent GetContractorInfoForTab(Organization org, IContractor contractor);
```

```
/// <summary>
```

```
/// Получить данные организации по ящику
```

```
/// </summary>
```

```
/// <param name="boxId"></param>
```

```
/// <returns></returns>
```

```
Organization GetOrganizationByBox(string boxId);
```

```
/// <summary>
```

```
/// Можно ли отправить документ
```

```
/// </summary>
```

```
/// <param name="owner"></param>
```

```
/// <param name="contractor"></param>
```

```
/// <returns></returns>
```

```
EODiadocCanSendStatus CanSendDocument(IContractor owner, IContractor contractor);
```

```
/// <summary>
```

```
/// Получить список документов по фильтру
```

```
/// </summary>
```

```
/// <param name="owner"></param>
```

```
/// <param name="filterCategory"></param>
```

```
/// <param name="lastCheck"></param>
```

```
/// <param name="alterIndexKey"></param>
```

```
/// <param name="count"></param>
```

```
/// <returns></returns>
```

```
DocumentList GetDocuments(IContractor owner, string filterCategory, DateTime? lastCheck, string  
alterIndexKey = null, int? count = null);
```

```
/// <summary>
```

```
/// Получить данные о документе в Диадоке
```

```
/// </summary>
```

```
/// <param name="entity"></param>
```

```
/// <returns></returns>
```

```
Document GetDocument(ICODiadocDocument entity);
```

```
/// <summary>
```

```
/// Получить данные о документе в Диадоке
```

```
/// </summary>
```

```
Document GetDocument(string boxId, string messageId, string documentId);
```

```
/// <summary>
```

```
/// Получить строковое содержимое сущности
```

```
/// </summary>
```

```
/// <param name="boxId"></param>
/// <param name="messageId"></param>
/// <param name="entityId"></param>
/// <returns></returns>
string GetEntityContentString(string boxId, string messageId, string entityId);
```

```
/// <summary>
/// Получить содержимое сущности
/// </summary>
/// <param name="boxId"></param>
/// <param name="messageId"></param>
/// <param name="entityId"></param>
/// <returns></returns>
byte[] GetEntityContent(string boxId, string messageId, string entityId);
```

```
/// <summary>
/// Сериализовать данные формализованного документа
/// </summary>
/// <param name="doc"></param>
/// <returns></returns>
object SerializeData(Document doc);
```

```
/// <summary>
/// Получить сообщение по документу
/// </summary>
/// <param name="entity"></param>
/// <returns></returns>
Message GetMessage(ICODiadocDocument entity);
```

```
/// <summary>
/// Генерация печатной формы документа
/// </summary>
/// <param name="entity"></param>
/// <returns></returns>
BinaryFile GeneratePrintForm(ICODiadocDocument entity);
```

```
/// <summary>
/// Сформировать документооборот целиком
/// </summary>
/// <param name="entity"></param>
/// <returns></returns>
```

BinaryFile GenerateDocumentZip(ICODiadocDocument entity);

/// <summary>

/// Получить информацию о подписи

/// </summary>

/// <param name="entity"></param>

/// <param name="own">Наша подпись</param>

/// <returns></returns>

SignatureInfo GetSignatureInfo(ICODiadocDocument entity, bool own = true);

/// <summary>

/// Получить сущность по идентификатору

/// </summary>

/// <param name="entity"></param>

/// <param name="entityId"></param>

/// <returns></returns>

Entity GetEntity(ICODiadocDocument entity, string entityId);

/// <summary>

/// Получить информацию об отказе в подписании

/// </summary>

/// <param name="entity"></param>

/// <returns></returns>

SignatureRejectionInfo ParseSignatureRejectionXml(ICODiadocDocument entity);

/// <summary>

/// Получить информацию об отказе в подписании

/// </summary>

/// <param name="boxId"></param>

/// <param name="messageId"></param>

/// <param name="entityId"></param>

/// <returns></returns>

SignatureRejectionInfo ParseSignatureRejectionXml(string boxId, string messageId, string entityId);

/// <summary>

/// Получить информацию о запросе на аннулирование

/// </summary>

/// <param name="entity"></param>

/// <returns></returns>

RevocationRequestInfo ParseRevocationRequestXml(ICODiadocDocument entity);

```
/// <summary>
/// Получить последнее событие для ящика
/// </summary>
/// <param name="boxId"></param>
/// <returns></returns>
BoxEvent GetLastEvent(string boxId);

/// <summary>
/// Получить новые события для ящика
/// </summary>
/// <param name="boxId"></param>
/// <param name="lastEventId"></param>
/// <returns></returns>
BoxEventList GetNewEvents(string boxId, string lastEventId);

/// <summary>
/// Получить подразделения для ящика
/// </summary>
/// <param name="boxId"></param>
/// <returns></returns>
DepartmentList GetDepartments(string boxId);

/// <summary>
/// Получить сотрудников для подразделения
/// </summary>
/// <param name="boxId"></param>
/// <param name="departmentId"></param>
/// <returns></returns>
IEnumerable<Employee> GetEmployees(string boxId, string departmentId);

/// <summary>
/// Отправить не формализованный документ
/// </summary>
/// <param name="doc">Документ</param>
/// <param name="fromBoxId">Ящик отправителя</param>
/// <param name="toBoxId">Ящик получателя</param>
/// <param name="departmentId">Подразделение</param>
/// <param name="userId">Сотрудник</param>
/// <param name="needRecipientSignature">Требуется подпись контрагента</param>
/// <param name="comment">Комментарий</param>
```

```

/// <returns></returns>

DiadocMessageResult SendNonformalized(string fromBoxId, string toBoxId,
EleWise.ELMA.Documents.Models.IDocument doc, Guid? departmentId = null, Guid? userId = null, bool
needRecipientSignature = true, string comment = null);

/// <summary>
/// Отправить счет
/// </summary>
/// <param name="doc">Документ</param>
/// <param name="fromBoxId">Ящик отправителя</param>
/// <param name="toBoxId">Ящик получателя</param>
/// <param name="docNumber">Номер счета</param>
/// <param name="docDate">Дата счета</param>
/// <param name="total">Сумма</param>
/// <param name="vat">НДС, если есть</param>
/// <param name="departmentId">Подразделение</param>
/// <param name="userId">Сотрудник</param>
/// <param name="comment">Комментарий</param>
DiadocMessageResult SendProformaInvoice(string fromBoxId, string toBoxId,
EleWise.ELMA.Documents.Models.IDocument doc, string docNumber, DateTime docDate, Money total,
Money? vat = null, Guid? departmentId = null, Guid? userId = null, string comment = null);

/// <summary>
/// Отправить договор
/// </summary>
/// <param name="fromBoxId">Ящик отправителя</param>
/// <param name="toBoxId">Ящик получателя</param>
/// <param name="doc">Документ</param>
/// <param name="docNumber">Номер договора</param>
/// <param name="docDate">Дата договора</param>
/// <param name="contractType">Тип договора</param>
/// <param name="contractPrice">Цена договора,если есть</param>
/// <param name="departmentId">Подразделение</param>
/// <param name="userId">Сотрудник</param>
/// <param name="comment">Комментарий</param>
/// <returns></returns>

DiadocMessageResult SendContract(string fromBoxId, string toBoxId,
EleWise.ELMA.Documents.Models.IDocument doc,
string docNumber, DateTime docDate, string contractType,
Money? contractPrice = null, Guid? departmentId = null, Guid? userId = null, string comment = null);

```



```

/// <summary>
/// Отправить дополнительное соглашение
/// </summary>
/// <param name="fromBoxId">Ящик отправителя</param>
/// <param name="toBoxId">Ящик получателя</param>
/// <param name="doc">Документ</param>
/// <param name="docNumber">Номер ДС</param>
/// <param name="docDate">Дата ДС</param>
/// <param name="contractNumber">Номер договора</param>
/// <param name="contractDate">Дата договора</param>
/// <param name="contractType">Тип договора</param>
/// <param name="total">Цена ДС, если есть</param>
/// <param name="departmentId">Подразделение</param>
/// <param name="userId">Сотрудник</param>
/// <param name="comment">Комментарий</param>
/// <returns></returns>

DiadocMessageResult SendSupplementaryAgreement(string fromBoxId, string toBoxId,
EleWise.ELMA.Documents.Models.IDocument doc,
    string docNumber, DateTime docDate,
    string contractNumber, DateTime contractDate, string contractType = "",
    Money? total = null, Guid? departmentId = null, Guid? userId = null, string comment = null);

```

```

/// <summary>
/// Отправить протокол согласования цены
/// </summary>
/// <param name="fromBoxId">Ящик отправителя</param>
/// <param name="toBoxId">Ящик получателя</param>
/// <param name="doc">Документ</param>
/// <param name="docNumber">Номер ДС</param>
/// <param name="docDate">Дата ДС</param>
/// <param name="departmentId">Подразделение</param>
/// <param name="userId">Сотрудник</param>
/// <param name="comment">Комментарий</param>
/// <returns></returns>

DiadocMessageResult SendPriceListAgreement(string fromBoxId, string toBoxId,
EleWise.ELMA.Documents.Models.IDocument doc,
    string docNumber, DateTime docDate,
    Guid? departmentId = null, Guid? userId = null, string comment = null);
}

```