**Структура документа в формате EDoc**

Структуру EDoc документа представляется в виде контейнера, включающего последовательность блочных элементов различного содержания.

## EDoc Контейнер

## 

### XML документ

*<root xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:edoc="URL-ссылка на edoc.xsd">*

*<edoc:document id="document-Guid документа">*

*<edoc:template id="templateGuid" compilerversion="версия компилятора/сборки, которой был создан документ">*

*<edoc:reference uri="Uri репозитория, где находится шаблон для текущего документа">*

*<edoc:digestmethod algorithm="Название алгоритма, которым будет сформирован хэш-образ для xslt-шаблона"></edoc:digestmethod>*

*<edoc:digestvalue><!--хэш образ xslt-шаблона--></edoc:digestvalue>*

*</edoc:reference>*

*</edoc:template>*

*<edoc:references><!--блок ссылок на другие документы-->*

*<edoc:referencedocument id="referenceGuid 1" name="Дополнение 1 к документу №-"></edoc:referencedocument>*

*<edoc:referencedocument id="referenceGuid 2" name="Дополнение 2 к документу №-"></edoc:referencedocument>*

*</edoc:references>*

*<edoc:data> <! -- Данные XML в свободной форме. Запрос или ответ в формате XML --></edoc:data>*

*<edoc:attachments> <!--блок вложений-->*

*<edoc:attachment id="attachmentGuid 1">*

*<edoc:reference uri="Uri файла вложения">*

*<edoc:digestmethod algorithm="Название алгоритма, которым будет сформирован хэш-образ для файла вложения"></edoc:digestmethod>*

*<edoc:digestvalue><!--хэш образ вложения--></edoc:digestvalue>*

*</edoc:reference>*

*</edoc:attachment>*

*</edoc:attachments>*

*</edoc:document>*

*<edoc:templatedata id="templateGuid"><!--base 64 файла шаблона--></edoc:templatedata>*

*<edoc:attachmentdata id="attachmentGuid 1"><!--base 64 файла вложения--></edoc:attachmentdata>*

*<edoc:signatures> <!--блок списка подписей -->*

*<edoc:signature id="signatureGuid">*

*<!--стандартный элемент XMLDSig, который подписывает блок данных, либо блок данных и другие подписи, либо только другую подпись-->*

*</edoc:signature>*

*</edoc:signatures>*

*</root>*

[**Информация о шаблоне визуального представления**](https://ca.agroprombank.com/pki/edoc#block1)

Шаблон визуального представления содержит ссылку на xslt документ. Блок ссылки на XSLT**-шаблон** содержит также дополнительную информацию о шаблоне: хэш шаблона, его идентификатор в формате Guid и версию компилятора, которым трансформируется шаблон.

**XSLT-шаблоны** представляют собой файлы визуального представления документов. Шаблоны построены на основе расширяемого языка преобразований таблиц стилей – XSLT. Они содержат информацию о том, как должен выглядеть документ (форматирование, абзацы, размер листа, поля страницы, таблицы и т.п.). Шаблоны могут быть встроены в файл-контейнер EDoc в виде base64 строки, либо присутствовать только в виде ссылки на репозиторий.

Ссылка на шаблон содержит хэш-образ шаблона, название алгоритма, которым был получен хэш-образ и относительный путь к файлу шаблона. Каждый шаблон подписывается квалифицированной подписью отправителя.

Следующим этапом, после подписи шаблона, является загрузка в репозиторий. В качестве репозитория может выступать локальная папка на диске, или удалённая папка на любом сервере, доступном в сети. Uri ссылка (не путать с Url) на шаблон в удалённом репозитории может иметь вид: **template.repository.edoc: //orgId/templateId/versionId,** где вместо template.repository.edoc:// будет подставлен Url (не путать с Uri) к репозиторию из файла конфигураций, и добавлено orgId/templateId/versionId.

При открытии документа в формате EDoc, приоритетом является проверка наличия вложенного в документ XSLT-шаблона, если таковой отсутствует, то его поиск по очереди ведётся во всех доступных онлайн-репозиториях. Загрузка шаблона производится из первого репозитория, в котором он будет найден.

**Общее описание документа (Document)**

*<root xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:edoc="URL ссылка на edoc.xsd">*

*<edoc:document id="document-9e1f08af-c7f8-47f8-9e28-4ee30d53d99f">*

*<!-- содержимое электронного документа -->*

*</edoc:document>*

*</root>*

Контейнер электронного документа основан на машиночитаемом формате электронной разметки XML. Контейнер имеет стандарт структуры документа – XSD. Все служебные теги стандарта находятся в пространстве имен «edoc:». Корневым элементом документа EDoc является элемент «Document». В корневом элементе документа располагается вся значимая подписываемая информация документа.

Каждый электронный документа формата EDoc имеет уникальный ID документа, который располагается в атрибуте «Id» ноды «Document». Уникальный идентификатор представляет собой запись «document-Guid документа». Например «document-9e1f08af-c7f8-47f8-9e28-4ee30d53d99f». Эта запись говорит о том, что данный документ имеет уникальный идентификатор «9e1f08af-c7f8-47f8-9e28-4ee30d53d99f». Уникальный идентификатор документа позволяет ссылаться на данный документ из других электронных документов формата EDoc.

**Ссылки на другие документы (Reference)**

*<edoc:references><!--блок ссылок на другие документы -->*

*<edoc:referencedocument id="referenceGuid 1" name="Доп. 1 к документу №-">*

*</edoc:referencedocument>*

*<edoc:referencedocument id="referenceGuid 2" name="Доп. 2 к документу №-">*

*</edoc:referencedocument>*

*</edoc:references><edoc:data></edoc:data>*

Формат EDoc позволяет ссылаться на другие документы EDoc, обеспечивая взаимосвязь между документами. Каждый документ может ссылаться на несколько документов, реализуя связь один ко многим. Ссылки на другие документы используются, например, для ответов на запросы при необходимости.

Ссылки на другие документы находятся блоке «References». Каждая ссылка на другой документ описывается в ноде «ReferenceDocument». В атрибутах указываются параметры внешнего документа. Атрибут «Id» содержит уникальный ID документа, атрибут «Name» содержит название документа. Количество ссылок не ограниченно.

**Данные документа (Data)**

Блок данных «Data» описывает суть документа. Содержимое блока данных зависит от вида документа. Каждый вид документа определяет свой набор тегов и значений.

Политика стандарта EDoc однозначно требует машиночитаемости блока данных. Отображение информации в понятном человеку виде, достигается путем XSLT преобразования документа.

Стандарт запрещает использовать несколько блоков данных в одном документе. Пример в части запроса налоговых органов о наличии счетов, об остатках денежных средств на счетах, по операциям на счетах налогоплательщика:

*<edoc:data>*

*<GeneralData>*

*<RequestType>MinFinDocumentType1</RequestType>*

*<BankTitle>\*сокращенное наименование банка, кредитной организации или филиала банка, кредитной организации\*</BankTitle>*

*<RequestDate>\*дата запроса\*</RequestDate>*

*<RequestorFullName>\*Ф.И.О. должностного лица налогового органа\*</RequestorFullName>*

*</GeneralData>*

*<DocumentData>*

*<RequestNumber>\*номерзапроса\*</RequestNumber>*

*<HeadTitle>о наличии счетов/ об остатках денежных средств на счетах/ по операциям на счетах конкретного лица</HeadTitle>*

*<LawReference>\*ссылка на статьи законов\* </LawReference>*

*<RequestedType>\*privateCustomer/corporateCustomer\*</RequestedType>*

*<RequestedPosition>\*privateNotary/iEnterpreneur\*</RequestedPosition>*

*<RequestedFullName>\*наименование юридического лица, Ф.И.О. индивидуального предпринимателя или частного нотариуса\*</RequestedFullName>*

*<RequestedBirthDate>\*дата рождения индивидуального предпринимателя или частного нотариуса (если известна)\*</RequestedBirthDate>*

*<EventType>\*вид мероприятия по контролю\*</EventType>*

*<EventReason>\*основания проведения мероприятия по контролю\*</EventReason>*

*<EventFromDate>\*дата начала проверяемого периода \*</EventFromDate>*

*<EventToDate>\*дата окончания проверяемого периода \*</EventToDate>*

*<GetInfo>\*информация, которую необходимо предоставить\*</GetInfo>*

*<FromDate>\*дата начала периода, за который запрашивается информация\*</FromDate>*

*<ToDate>\*дата окончания периода, за который запрашивается информация\*</ToDate>*

*<EndTitle>\*наименование налогового органа\*</EndTitle>*

*<EndDate>\*дата предоставления информации\*</EndDate>*

*<RequestorContactData>\*контактныйномер\*</RequestorContactData>*

*</DocumentData>*

*</edoc:data>*

**Информация о вложенных файлах (Attachments)**

*<edoc:attachments>*

*<!--блок вложений-->*

*<edoc:attachment id="attachmentGuid 1">*

*<edoc:reference uri="Uri файла вложения">*

*<edoc:digestmethod algorithm="Название алгоритма, которым будет сформирован хэш-образ для файла вложения">*

*<edoc:digestvalue><!--хэш образ вложения--></edoc:digestvalue>*

*</edoc:digestmethod>*

*</edoc:reference>*

*</edoc:attachment>*

*</edoc:attachments>*

Контейнер электронного документа EDoc позволяет работать с вложениями. Вложенным может быть файл любого формата и размера. Стандарт позволяет работать как с вложениями внутри документа, так и находящихся в виде отдельных файлов. Стандарт гарантирует проверку файла на предмет подмены и корректировки. Каждое вложение защищено ХЭШ значением, которое формируется с помощью математических алгоритмов на основании содержимого файла. Значение ХЭШ функции и название алгоритма указываются в параметрах вложения.

Информация о вложенных файлах содержится в ноде **«Attachments».** Информация о конкретном вложенном файле содержится в ноде **«Attachment».** Каждое вложение имеет уникальный идентификатор – GUID, который указывается в атрибуте **«Id».**  Нода **«Reference»** содержит информацию о вложенном файле. Атрибут **«URI»** содержит информацию о местоположении файла согласно формата URI. Значение ХЭШ функции содержатся в ноде **«DigestValue».** Параметры алгоритма формирования ХЭШ значения указываются в ноде **«DigestMethod»** и атрибуте **«Algorithm».** Содержимое файла может быть вложено в документ EDoc. Описание формата вложения содержится в разделе **Вложенные файлы (данные).**

**Вложенный шаблон визуального представления (TemplateData)**

*<edoc:templatedata id="templateGuid">*

*<!--base 64 файла шаблона-->*

*</edoc:templatedata>*

Шаблон визуального представления должен быть вложен внутрь документа. Вложение шаблона визуального представления позволяет работать с документами в оффлайн режиме, программе просмотра электронного документа не потребует поиска шаблона отображения в репозитории шаблонов, но при этом размер самого электронного документа увеличится на размер вложенного шаблона. Если программа просмотра находит вложенный шаблон, то документ отображается только с помощью вложенного шаблона отображения.

Данные вложенного шаблона находятся за пределами блока подписываемых данных в ноде «TemplateData». Значение атрибута «Id» содержит уникальный ID шаблона. Данный уникальный идентификатор должен совпадать со значением атрибута «Id» ноды «Template» в блоке подписываемых данных. Если идентификаторы не будут совпадать, то вложенный шаблон применяться не будет. Шаблон в формате Base64 строки располагается внутри ноды «TemplateData».

**Вложенные файлы (AttachmentData)**

*<edoc:attachmentdata id="attachmentGuid 1"><!--base 64 файла вложения--> </edoc:attachmentdata>*

Стандарт позволяет размещать содержимое вложенных файлов внутри электронного документа. Это позволит автономно использовать документ, в случае использования внешних файлов, они всегда должны поставляться вместе с документом и находится в постоянной доступности. Вложение файлов позволяет этого избежать, но при этом размер электронного документа увеличивается на размер вложенных файлов.

Содержимое файлов размещается в ноде «AttachmentData» в формате Base64 строки. Каждый файл имеет уникальный идентификатор – атрибут «Id». Данный идентификатор должен совпадать со значением атрибута «Id» ноды «Attachment» в блоке подписываемых данных.

Стандарт поддерживает сколь угодно много вложенных файлов. Хэш файла должен совпадать с Хэшем, указанным в блоке подписанных данных.

**Электронные подписи**

*<edoc:signatures>*

*<!--блок списка подписей-->*

*<edoc:signature id="signatureGuid">*

*<!-- стандартный элемент XMLDSig, который подписывает блок данных, либо блок данных и другие подписи, либо только другую подпись -->*

*</edoc:signature>*

*</edoc:signatures>*

EDoc гарантирует неизменность и неотрекаемость электронного документа благодаря электронной подписи. Стандарт позволяет накладывать подписи формата Xades: XADES-BES, XADES-T, а также подписи формата XMLDSIG.

Стандарт поддерживает последовательное, параллельное и смешанное подписание документа. При параллельном подписании подписывается только блок данных. При данной схеме удаление (отмена) одной подписи не ведет к невозможности проверки остальных подписей. Каждая подпись независима.

Последовательное подписание документа применяется в случае, когда подписант ставит свою подпись только при наличии подписей ответственных подписантов. В данном случае подпись подписывает блок данных и предыдущую подпись или подписи. В случае отзыва нижестоящей подписи, все зависимые подписи становятся недействительными. Последовательное подписание исключает отзыв подписи нижестоящего подписанта без ведома остальных.

Смешенное подписание позволяет комбинировать параллельное и последовательное подписание. В таком режиме могут одновременно присутствовать как последовательные, так и параллельные подписи.