



Документация по QR/Кросс подписанию в приложении eGov Mobile

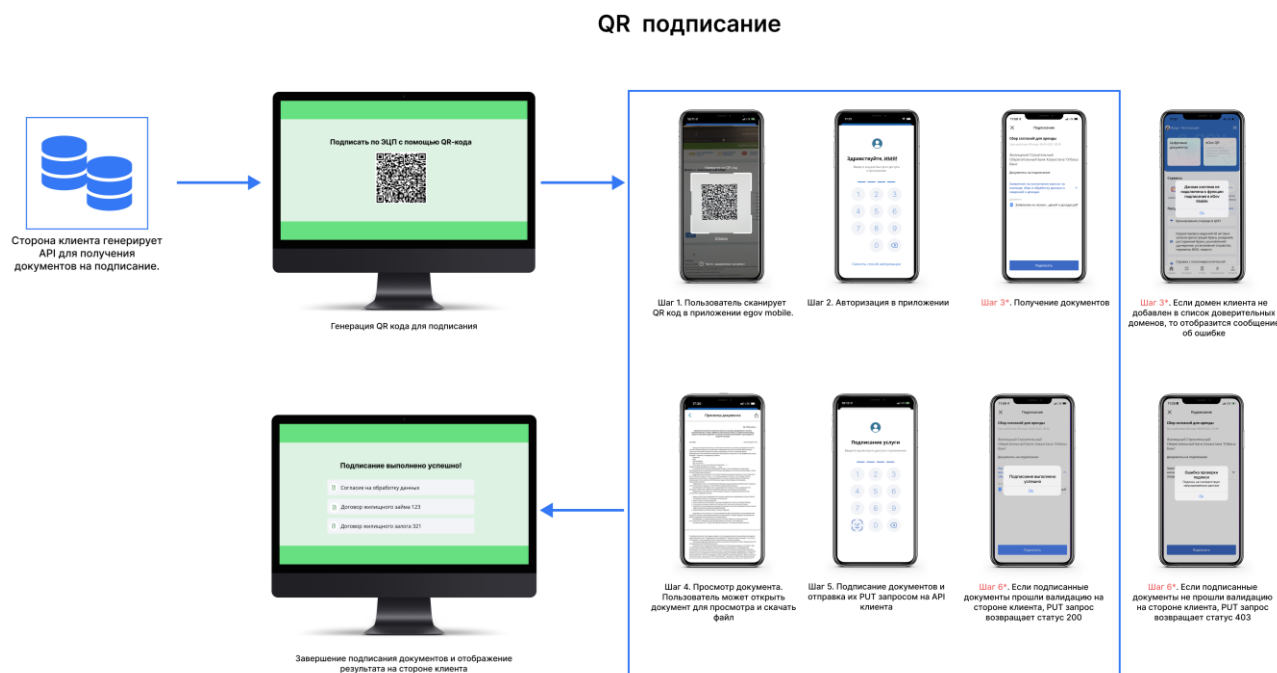
Данная документация детально описывает процесс QR и Кросс подписания документов для внешних, сторонних сервисов, интегрированных с приложением eGov Mobile. В данном контексте используются два API запроса: API №1 и API №2.

1. QR подписание

Это метод, при котором подпись осуществляется через сканирование QR-кода. QR-код содержит API №1 с префиксом «**mobileSign:**». Пример содержания QR кода (пробелы исключаются):

mobileSign:<https://some.domen.kz>

Рис. 1 - Процесс QR подписания



2. Кросс подписание

Это метод, при котором клиентское приложение инициирует процесс подписания, используя динамическую ссылку для перехода непосредственно в приложение eGov Mobile.

Формат Динамической Ссылки:

iOS:

https://mgovsign.page.link/?link={API_№1}&isi=1476128386&ibi=kz.egov.mobile

Android:

https://mgovsign.page.link/?link={API_№1}&apn=kz.mobile.mgov

Компоненты Ссылки:

- URL Префикс:
 - **https://mgovsign.page.link** - Это фиксированный URL префикс, который используется для идентификации динамических ссылок eGov Mobile.
- URL Параметр **?link=**:
 - Значение этого параметра является API №1, который используется для получения данных на подписание.
 - Пример: **?link={API_№1}** - где **{API_№1}** будет заменено на актуальный URL API.
- Параметры для перехода в приложение:
 - **&apn=kz.mobile.mgov** - Название пакета Android приложения. Используется для перенаправления на приложение в Google Play Store, если оно не установлено на устройстве.
 - **&isi=1476128386** - ID приложения для перехода в App Store (iOS). Этот параметр позволяет перенаправить пользователя в App Store, если приложение eGov Mobile не установлено на устройстве.
 - **&ibi=kz.egov.mobile** - Bundle ID iOS приложения. Используется для идентификации и запуска приложения на устройствах Apple.

Меняется только значение параметра **link**, остальные параметры ссылки должны оставаться статичными

Пример генерации и использования динамической ссылки

Шаг 1: Генерация API №1 Клиентом

Сторона клиента инициирует процесс, сначала генерируя API №1 для получения данных, которые необходимо подписать. Например:

`https://some.domen.kz/mgovSign?parameter1=value1¶meter2=value2`

Шаг 2: Кодирование URL Параметров

Для предотвращения конфликтов с параметрами динамической ссылки, **важно** кодировать URL параметры API клиента.

Примеры кодирования:

- `&` кодируется как `%26`
- `=` кодируется как `%3D`
- `?` кодируется как `%3F`

Шаг 3: Вставка API №1 в Динамическую Ссылку

После генерации и кодирования, API №1 вставляется в параметр `?link=` динамической ссылки.

Пример Готовой Динамической Ссылки:

Исходный url:

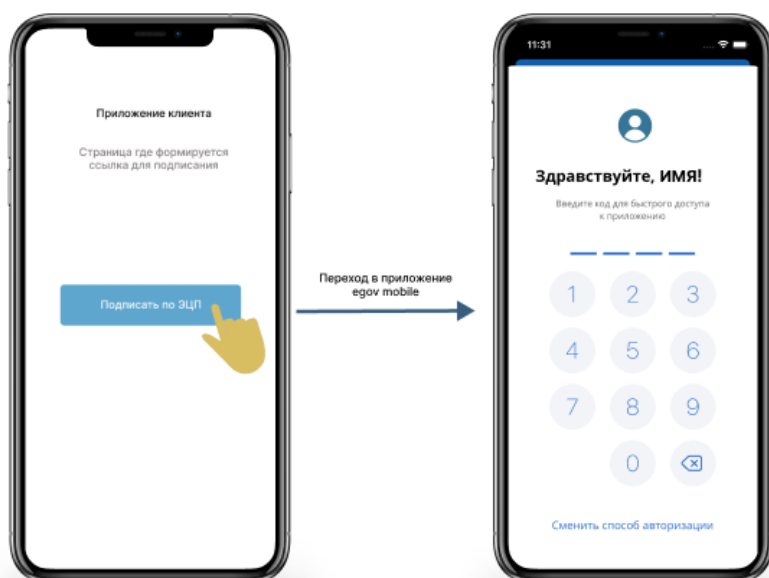
`https://some.domen.kz/mgovSign?parameter1=value1¶meter2=value2`

iOS:

`https://mgovsign.page.link/?link=https://some.domen.kz/mgovSign%3Fparameter1%3Dvalue1%26parameter2%3Dvalue2&isi=1476128386&ibi=kz.egov.mobile`

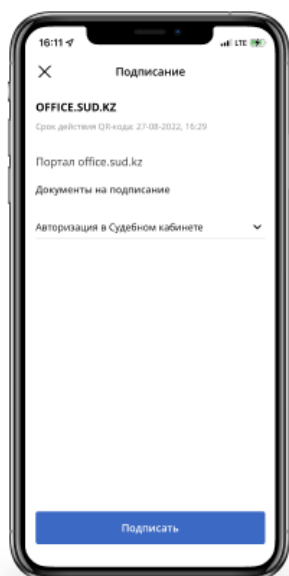
Android:

`https://mgovsign.page.link/?link=https://some.domen.kz/mgovSign%3Fparameter1%3Dvalue1%26parameter2%3Dvalue2&apn=kz.mobile.mgov`

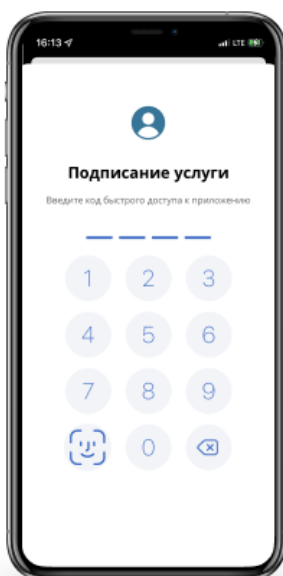


Шаг 1. Сторона клиента генерирует API для получения документов на подписание. Затем генерирует deeplink для перехода в приложение egov mobile

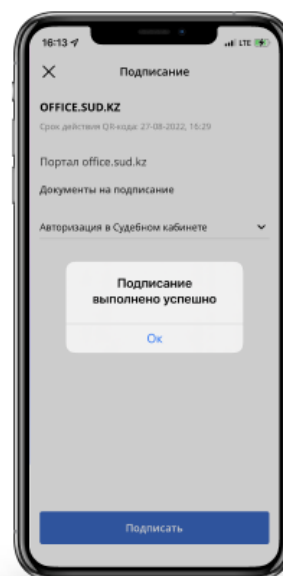
Шаг 2. После перехода в приложение egov mobile, пользователь проходит авторизацию



Шаг 3. По полученному API из deeplink, приложение получает список документов



Шаг 4. Пользователь подписывает полученные документы



Шаг 5. Приложение отправляет POST запрос с подписанными документами на API клиента

Рис. 2 - Процесс Кросс подписания

3. Методы подписания документов:

В приложении реализованы следующие методы подписания документов:

- **CMS_WITH_DATA:**
 - Этот метод включает в себя подпись документа, который передается закодированный в формате base64.
 - При подписании в структуру CMS добавляется штамп времени, подписанный НУЦ РК.
 - При использовании этого метода подпись и данные объединяются в один пакет, тем самым документ можно извлечь и после подписания. Подписанная структура CMS с массивом байтов исходного файла возвращается обратно в закодированном формате base64.
- **CMS_SIGN_ONLY:**
 - Данный метод так же включает в себя подпись документа, который передается закодированный в формате base64.
 - При подписании в структуру CMS добавляется штамп времени, подписанный НУЦ РК.
 - При этом методе создается только хэш подписи, без включения исходных данных. Подписанная структура CMS возвращается обратно в закодированном формате base64.
- **XML:**
 - Подпись осуществляется непосредственно в XML-документе, что позволяет интегрировать электронную подпись в структуру XML. Формат кодирования - UTF8.
- **SIGN_BYTES_ARRAY:**
 - При этом методе для подписания входные данные преобразуются в массив байтов.
- **MIX_SIGN:**
 - Это комбинированный метод, который позволяет использовать различные типы подписей в зависимости от контекста и требований к документу.
 - MIX_SIGN может применяться в случаях, когда необходимо сочетать несколько методов подписания для одного набора документов, обеспечивая гибкость и расширенные возможности подписи.
 - **ВАЖНО:** Данный метод подписания поддерживается начиная с версии **2.4.1** для приложений на iOS и с версии **1.6.77** для приложений на Android.

* Каждый из этих методов подписания предназначен для определенных сценариев использования, и выбор конкретного метода зависит от требований безопасности, удобства обработки и передачи данных, а также от специфики интеграции с другими системами и сервисами. Важно также соблюдать требования к подписи, которые могут варьироваться в зависимости от законодательства и внутренних политик компании.

4. Описание API №1

Запрос API №1 представляет собой метод **GET** и содержит общие данные для подписания. В ответе на запрос API №1, клиент получает всю необходимую информацию для начала процесса подписания.

В JSON-объекте:

Поле	Тип данных	Описание
description	String	описывает тип или содержание документа
expiry_date	String □ Дата в формате ISO 8601: "yyyy-MM-dd'T'HH:mm:ss.SSSz"	указывает на дату истечения срока действия подписания документа
organisation □	JSON object	содержит информацию об организации: наименование на разных языках и БИН
□ nameRu	String	
□ nameKz	String	
□ nameEn	String	
□ bin	String	
document □	JSON object	указаны данные для аутентификации и получения документов для подписания
□ uri	String	содержит API №2 для получения документов
□ auth_type	String	тип аутентификации: <i>Token</i> , <i>Eds</i> или <i>None</i>
□ auth_token	String (опциональное поле)	содержит токен аутентификации если <i>auth_type</i> = <i>Token</i>

В объекте "**document**" поле "**auth_type**" определяет тип аутентификации в системе клиента, т.е *клиент сам определяет* какой тип аутентификации необходим для вызова API №2:

- "**Token**" - При указании этого типа так же необходимо передавать токен в поле **auth_token**. В этом случае GET запрос на получение документов для подписания и PUT запрос для отправления подписанных документов обратно на сервер на API №2 будут вызываться с указанием токена авторизации в Headers:

```
{ "Authorization" : "Bearer значение_auth_token" }
```

- **"Eds"** - При указании этого типа запрос для получения документов на API №2 будет вызываться методом POST. А в теле запроса будет содержаться подписанный XML: { "xml": "signed xml" }. XML для подписания генерируется на стороне eGov mobile и будет содержать API №2 и текущее время, а для подписания используется ключи ЭЦП для авторизации (AUTH_*.p12):

```

<?xml version="1.0" encoding="UTF-8"?>
<login>
  <url>\(API #2)</url>
  <timeStamp>\(timestamp)</timeStamp>
</login>

```

- **"None"** - Аутентификация не требуется. При указании этого типа запрос на API №2 будет вызываться методом GET.

Пример JSON-объекта, получаемого в ответе на запрос по API №1:

```

{
  "description": "Подписание документов",
  "expiry_date": "2024-01-01T04:52:23.626Z",
  "organisation": {
    "nameRu": "АО \"НИТ\"",
    "nameKz": "\"ҰАТ\" АҚ",
    "nameEn": "NITEC",
    "bin": "000740000728"
  },
  "document": {
    "uri": "some API #2 here",
    "auth_type": "Token",
    "auth_token": "2dffe45dsvg"
  }
}

```

5. Описание API №2

API №2 представляет собой запрос для получения документов на подписание и отправки подписанных документов.

Данный API поддерживает следующие методы, в зависимости от типа аутентификации (**auth_type**), выбранного в API №1:

- **GET:**
 - Используется, когда **auth_type** равен **"None"** или **"Token"**.
 - В случае "Token", необходимо включить токен аутентификации в заголовок запроса.
- **POST:**
 - Применяется, когда **auth_type** равен **"Eds"**.
- **PUT:**
 - Используется после подписания документов для отправки данных обратно на сервер. В случае "Token", необходимо включить токен аутентификации в заголовок запроса.

В JSON-объекте из ответа API №2:

Поле	Тип	Описание
signMethod	String	Метод подписания: • XML, • CMS_WITH_DATA • CMS_SIGN_ONLY • SIGN_BYTES_ARRAY • MIX_SIGN
version	Int	По умолчанию указывать значение 1.
documentsToSign ☐	Array of JSON objects	Массив документов для подписания
☐ id	Int	Идентификатор документа
☐ signMethod	String	Метод подписания: XML, CMS_WITH_DATA, CMS_SIGN_ONLY, SIGN_BYTES_ARRAY ☐ Данное поле указывается только для метода подписания MIX_SIGN, в остальных случаях это поле не используется.
☐ nameRu	String	Наименование документа на русском языке
☐ nameKz	String	Наименование документа на казахском языке
☐ nameEn	String	Наименование документа на английском языке
☐ meta ☐	Array of JSON objects	Массив метаданных
☐ ☐ name	String	Название метаданных
☐ ☐ value	String	Значение метаданных
☐ documentXml	String	XML-документ для подписания (только для метода XML)
☐ document ☐	JSON object	Информация о типе документа и самом документе (только для методов CMS и SIGN_BYTES_ARRAY)
☐ ☐ file ☐	JSON object	Информация о файле
☐ ☐ ☐ mime	String	MIME-тип файла
☐ ☐ ☐ data	String	Данные файла или подписанная структура CMS в кодировке base64

Примечание:

- **documentXml** (String) - Данное поле применимо только для подписания методом XML, для остальных методов подписания это поле не указывается.
- **document** (JSON object): - Данный JSON объект применим только для подписания методами CMS и SIGN_BYTES_ARRAY, для XML подписания этот объект и его содержимое не указывается
- **mime** - тип подписываемого документа, необходим для отображения пользователю подписываемого документа. Возможные значения:
 - "text/plain" - пользователю отображается текст.
 - "application/pdf" - пользователю отображается PDF-документ с возможностью загрузить и отправить его.
 - "application/xml" - пользователю отображается XML (применимо только для SIGN_BYTES_ARRAY).
 - Данные остальных MIME-типов не отображаются пользователю.
- Только для метода подписания **SIGN_BYTES_ARRAY**: После подписания, значение **document.file.data** преобразуется в JSON в виде строки:

```
{
  "signature": "SIGNATURE BASE64" //подпись в формате BASE64
  "certificate": "PEM BASE64"
}
```

Пример JSON для метода подписания XML:

```
{
  "signMethod": "XML",
  "version": 1,
  "documentsToSign": [
    {
      "id": 12345,
      "nameRu": "Документ для подписания",
      "nameKz": "Қол қою үшін құжат",
      "nameEn": "Document for Signing",
      "meta": [
        {
          "name": "Дата создания",
          "value": "2024-01-01"
        }
      ],
      "documentXml": "<xml>Тут XML документ для подписания</xml>"
    }
  ]
}
```

Пример JSON для метода подписания CMS_SIGN_ONLY:

```
{
  "signMethod": "CMS_SIGN_ONLY",
  "version": 1,
  "documentsToSign": [
    {
      "id": 67890,
      "nameRu": "Соглашение о неразглашении",
      "nameKz": "Құпиялылық туралы келісім",
      "nameEn": "Non-Disclosure Agreement",
      "meta": [
        {
          "name": "Дата создания",
          "value": "2024-01-02"
        }
      ],
      "document": {
        "file": {
          "mime": "application/pdf",
          "data": "base64 encoded PDF data"
        }
      }
    }
  ]
}
```

Пример JSON для метода подписания CMS_WITH_DATA:

```
{
  "signMethod": "CMS_WITH_DATA",
  "version": 1,
  "documentsToSign": [
    {
      "id": 54321,
      "nameRu": "Договор аренды",
      "nameKz": "Жалға алу шарты",
      "nameEn": "Lease Agreement",
      "meta": [
        {
          "name": "Номер договора",
          "value": "123/ABC"
        }
      ],
      "document": {
        "file": {
          "mime": "application/pdf",
          "data": "base64 encoded PDF data"
        }
      }
    }
  ]
}
```

Пример JSON для подписания документов для метода подписания SIGN_BYTES_ARRAY:

```
{
  "documentsToSign": [
    {
      "id": 1,
      "nameRu": "TEXT",
      "nameKz": "TEXT",
      "nameEn": "TEXT",
      "document": {
        "file": {
          "mime": "text/plain",
          "data": "123"
        }
      }
    },
    {
      "id": 2,
      "nameRu": "PDF",
      "nameKz": "PDF",
      "nameEn": "PDF",
      "meta": [
        {
          "name": "ИИН",
          "value": "87654321"
        }
      ],
      "document": {
        "file": {
          "mime": "application/pdf",
          "data": "JVBERi0xLjcK..."
        }
      }
    },
    {
      "id": 3,
      "nameRu": "XML",
      "nameKz": "XML",
      "nameEn": "XML",
      "document": {
        "file": {
          "mime": "application/xml",
          "data": "<?xml version='1.0' encoding='UTF-8'?><invoice>...</invoice>"
        }
      }
    }
  ],
  "version": 1,
  "signMethod": "SIGN_BYTES_ARRAY"
}
```

Пример подписанного JSON для отправки на сервер для метода подписания SIGN_BYTES_ARRAY:

```
{
  "documentsToSign": [
    {
      "id": 1,
      "nameRu": "TEXT",
      "nameKz": "TEXT",
      "nameEn": "TEXT",
      "document": {
        "file": {
          "mime": "text/plain",
          "data":
            "{ \"signature\": \"BASE64_SIGNATURE_STRING\", \"certificate\": \"BASE64_CERTIFICATE_STRING\" }"
        }
      }
    },
    {
      "id": 2,
      "nameRu": "PDF",
      "nameKz": "PDF",
      "nameEn": "PDF",
      "meta": [
        {
          "name": "ИИН",
          "value": "87654321"
        }
      ],
      "document": {
        "file": {
          "mime": "application/pdf",
          "data":
            "{ \"signature\": \"BASE64_SIGNATURE_STRING\", \"certificate\": \"BASE64_CERTIFICATE_STRING\" }"
        }
      }
    },
    {
      "id": 3,
      "nameRu": "XML",
      "nameKz": "XML",
      "nameEn": "XML",
      "document": {
        "file": {
          "mime": "application/xml",
          "data":
            "{ \"signature\": \"BASE64_SIGNATURE_STRING\", \"certificate\": \"BASE64_CERTIFICATE_STRING\" }"
        }
      }
    }
  ],
  "version": 1,
  "signMethod": "SIGN_BYTES_ARRAY"
}
```

Пример JSON для метода подписания MIX_SIGN:

```

{
  "signMethod": "MIX_SIGN",
  "version": 1,
  "documentsToSign": [
    {
      "id": 11223,
      "signMethod": "CMS_SIGN_ONLY",
      "nameRu": "Протокол собрания",
      "nameKz": "Отырыс хаттамасы",
      "nameEn": "Meeting Minutes",
      "document": {
        "file": {
          "mime": "application/pdf",
          "data": "base64 encoded PDF data"
        }
      }
    },
    {
      "id": 44556,
      "signMethod": "XML",
      "nameRu": "Поручение",
      "nameKz": "Тапсырма",
      "nameEn": "Assignment",
      "documentXml": "<xml>Тут XML документ для подписания</xml>"
    }
  ]
}

```

Процедура подписания

- Если все поля и значения верны, приложение отобразит входные данные. Пользователь имеет возможность просмотра документов для подписания.
- В результате подписания, полученный по API №2 JSON-объект "мутирует", то есть подписанные документы замещают исходные данные (поле "documentXml" или "data"). Остальные поля остаются в исходном виде.
- Полученный JSON-объект с уже подписанными данными отправляется обратно на API №2 методом **PUT**.

Возвращаемые статусы

- Подписанные документы успешно прошли валидацию подписи: Статус 200 и сообщение "success".
- Подписанные документы не прошли валидацию подписи: Статус 403.

Примечания:

- * Все поля и статические значения в ответах запросов должны строго соответствовать указанным регистрам, как описано в предыдущих разделах документации.
- * Валидация подписей проходит на стороне клиента согласно [Правил проверки подлинности электронной цифровой подписи](#).
- * С версиями приложения 2.4.1 для iOS и 1.6.77 для Android, во всех запросах к API теперь указывается предпочтительный язык интерфейса в заголовке **Accept-Language**: "ru/kk/en".
- * Эти обновления также включают отображение ошибок сервера непосредственно в интерфейсе приложения, что упрощает пользователям понимание проблем. Сообщения об ошибках от сервера должны предоставляться в формате: { "message": "Локализованный текст сообщения" }

Отображение данных JSON из API №1 и API №2

В

интерфейсе
приложения:

	9:41	Подписание
API №1	Договор на оказание услуг, Соглашение о неразглашении Срок действия QR-кода: 30-09-2023, 15:00 ТОО "Белоснежка и семь гномов"	description expiry_date organisation->name(-Kz/-Ru/-En)
	Документы на подписание	
	Договор на оказание безвозмездных услуг повара Соглашение о неразглашении конфиденциальных данных лесного содружества ИИН 123456780345 Тип запроса Кредитная история Номер телефона +77771500034 Документ fileName.pdf	documentsToSign->name(-Kz/-Ru/-En) documentsToSign->name(-Kz/-Ru/-En) documentsToSign->meta documentsToSign->document->file->data
Подписать Отказать		

9:41

Подписание

Договор на оказание услуг, Соглашение о неразглашении
 Срок действия QR-кода: 30-09-2023, 15:00
 ТОО "Белоснежка и семь гномов"

Документы на подписание

Договор на оказание безвозмездных услуг повара
 Соглашение о неразглашении конфиденциальных данных лесного содружества
 ИИН 123456780345
 Тип запроса Кредитная история
 Номер телефона +77771500034
 Документ fileName.pdf

Подписать

Отказать