Робота з API порталу EDIN ID

Колекцію Postman можна скачати в сторінці "Перелік методів API"

Перелік методів API по роботі з порталом EDIN ID

Всі запити нижче перерахованих API методів порталу EDIN ID направляються на адресу: https://id.edin.ua

Для підписання хеш(ів) та/або файлу пароль передається в зашифрованому вигляді.

Отримати і зберегти публічний ключ	GET /api/external/key
Підписати хеш ключем співробітника компанії	POST /api/external/company/sign
Підписати файл ключем співробітника компанії	POST /api/external/company/sign/file

Колекція Postman

Шифрування пароля за допомогою відкритого ключа RSA

Функції для шифрування

```
/**
 * Encrypt data with RSA public key
 * @param publicKey - get from server
 * @param password - user key password
 */
utils = {
    rsaEncrypt: async function(publicKey, password) {
        const cryptoKey = await crypto.subtle.importKey("spki", publicKey, {
            name: "RSA-OAEP",
            hash: "SHA-256"
        }, true, ["encrypt"]);
        const encodedText = new TextEncoder().encode(password);
        const encrypted = await crypto.subtle.encrypt({name: "RSA-OAEP"}, cryptoKey,
encodedText);
        //
        return toBase64(encrypted);
    }
}

function toBase64(buffer) {
    const bytes = new Uint8Array(buffer);
    let binary = '';
    for (let i = 0; i < bytes.byteLength; i++) {
        binary += String.fromCharCode(bytes[i]);
    }
    return btoa(binary);
}
```

Шифрування пароля

```
const encrypted = await utils.rsaEncrypt(Uint8Array.from(publicKey), password));
```

де,

- publicKey - публічний ключ, отриманий методом GET /api/external/key
- password - пароль від ключа

Методи API

Отримати і зберегти публічний ключ GET /api/external/key

REQUEST

URL	
Метод запиту	GET
URL запиту	**/api/external/key**
Authorization	
Auth type	API key
Key / Value	**x-system-id** / токен, отриманий при підключенні
Params	
type	тип відповіді JSON|PEM|XML (якщо параметр не передавати за замовченням буде JSON)

RESPONSE

В тілі **відповіді** повертається ключ у вказаному форматі:

- якщо type = JSON – повертається масив байт
- якщо type = PEM – повертається PEM-файл у вигляді

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQ9QIDAQAB
-----END PUBLIC KEY-----
```

- якщо type = XML – повертається XML-файл у вигляді

```
<?xml version="1.0"?>
<RSAKeyValue>
  <Modulus>wxWy8iReusbmiadsULVLSD36+l5k6cZ0=</Modulus>
  <Exponent>AQAB</Exponent>
</RSAKeyValue>
```

Для type in (PEM, XML) в reponse-header передається параметр `x-key-ttl`, в якому передається термін життя відкритого ключа.

Підписати хеш ключем співробітника компанії POST /api/external/company/sign

REQUEST

URL	
Метод запиту	POST
URL запиту	**/api/external/company/sign**
Authorization	
Auth type	API key
Key / Value	**x-system-id** / токен, отриманий при підключенні
Headers	
Content-Type	aplication/json

Request body	{
	"key": "{{id ключа}}",
	"password": "{{зашифрований пароль від ключа}}",
	"algorithm": "DSTU4145_GOST34311",
	"signType": "тип підпису, приймає значення CADES_BES, CADES_T, CADES_C, CADES_X_LONG, CADES_X_LONG_TRUSTED. Якщо не передано, за замовченням підставляється CADES_BES"
	"hashes": [
	{"hash": "{{хеш документа, що підписується}}", "description": "опис документа, що підписується"}
	]
	}

RESPONSE

В тілі **відповіді** повертається масив підписів.

Підписати файл ключем співробітника компанії POST /api/external/company/sign/file

REQUEST

URL	
Метод запиту	GET
URL запиту	**/api/external/company/sign/file**
Authorization	
Auth type	API key
Key / Value	**x-system-id** / токен, отриманий при підключенні
Headers	
Content-Type	multipart/form-data
Request body	Параметри тіла запиту: • **password** – зашифрований пароль. • **file.pdf** – дані (контент) для підпису у вигляді файлу або base64 (залежно від **inputFormat**)
Params	• **key** – ідентифікатор ключа. • **type –** тип підписання, може приймати значення: ◦ append – додає підпис до переданого файлу. **ВАЖЛИВО!** Якщо файл вже був підписаний, то підпис додається до існуючого, а не підписується разом з існуючим підписом. ◦ sign - підписує контент, використовується за замовчуванням. • **result –** формат підписання (дані й підпис в одному файлі, дані й підпис в окремих файлах), може приймати значення: ◦ enveloped – у відповідь надійде файл з підписом ◦ detached – у відповідь надійде тільки підпис

	• **inputFormat** – формат вхідних даних для підписання, може приймати значення: ◦ file – файл в бінарному вигляді ◦ base64 – файл у вигляді base64 рядка • **outputFormat** – формат вихідних (результуючих) даних після підписання, може приймати значення: ◦ file – файл в бінарному вигляді ◦ base64 – файл у вигляді base64 рядка • **signType** – тип підпису, приймає значення CADES_BES, CADES_T, CADES_C, CADES_X_LONG, CADES_X_LONG_TRUSTED. Якщо не передано, то за замовченням підставляється CADES_BES

RESPONSE

В тілі **відповіді** повертається код 200 та результат підписання відповідно до параметрів **result** та **outputFormat.**

Помилки при роботі з API

Загальні помилки

Код відповіді	Опис
403	Помилка авторизації зовнішньої системи або підключення зовнішньої системи відбувається з недоступних IP-адрес.

Помилки підписання хешів

Код відповіді	Опис
406	Ключ не знайдено або неправильний пароль
413	Більше 100 хешів для підписання
422	Помилка в процесі підписання
426	Неможливо дешифрувати пароль до ключа. Спробувати оновити публічний ключ для шифрування та повторити операцію

Помилки підписання файлів

Код відповіді	Опис
406	Ключ не знайдено або неправильний пароль
412	Атрибути ("key" або "password") або файл неправильно заповнені або відсутні у запиті
413	Розмір файла для підпису більше 1Мб
422	Помилка в процесі підписання
426	Неможливо дешифрувати пароль до ключа. Спробувати оновити публічний ключ для шифрування та повторити операцію