Створити чернетку ключа для співробітника POST /api/external/company/employee/pkey/generate/draft

REQUEST

URL	
Метод запиту	POST
URL запиту	**/api/external/company/employee/pkey/generate/draft**
URL параметри	**companyCode** (обов'язково) - код Компанії; **employeeIpn** (обов'язково) - ІПН/РНОКПП співробітника; **store** (обов'язково) - може приймати одне зі значень: • `cloud` - сервер генерує ключ і p10-запити у форматі `PKCS #10`. Потрібен `info.pkPassword`. • `file` - зовнішня система генерує ключ, а сервер приймає p10-запити у форматі `PKCS #10` в part `requests`
Authorization	
Auth type	API key
Key / Value	**x-system-id** - токен, отриманий при підключенні
Headers	
Content-Type	multipart/form-data
REQUEST	
REQUEST Body	**info** (обов'язково) JSON attr - параметри CA user і ключа; **requests** (обов'язково, тільки для `store=file`) JSON attr - p10-запити у форматі `PKCS #10`, створені зовнішньою системою. Всі персональні дані ідентифікації беруться з підпису/сертифіката у file `identification`, а не з JSON `info`.

`info`:

```
{
  "pkName": "Ключ Іваненко",
  "pkType": "UA",
  "pkStoreType": "HSM",
  "pkPassword": "base64 RSA-encrypted password",
  "pkIsStamp": false,
  "emplTitle": "Менеджер",
  "emplOrgUnit": "Відділ продажів",
  "caPassPhrase": "base64 RSA-encrypted pass phrase",
  "certType": "SIGN_AND_ENCRYPT",
  "certValidity": "TWO"
}
```

Опис полів `info`:

Поле	Тип	Обов’язкове	Опис
`pkName`	string	так	Назва ключа.
`pkType`	enum	так	`UA` або `ECDSA`.
`pkStoreType`	enum	так	`HSM` або `FILE`.
`pkPassword`	string	для `store=cloud`	Base64 від RSA-encrypted bytes пароля ключа. Шифрувати актуальним public key з `/api/external/key`.
`pkIsStamp`	boolean	так	`true` — печатка, `false` — особистий ключ співробітника.
`emplTitle`	string/ null	ні	Посада для CA user.
`emplOrgUnit`	string/ null	ні	Підрозділ для CA user.
`caPassPhrase`	string	так	Base64 від RSA-encrypted bytes секретної фрази CA user. Шифрувати актуальним public key з `/api/external/key`.
`certType`	enum	так	`SIGN_ONLY` або `SIGN_AND_ENCRYPT`.
`certValidity`	enum	так	`ONE` або `TWO`.

`requests` для `store=file`:

```
{
  "ecdsa": "base64 PKCS #10 request",
  "signature": "base64 PKCS #10 request",
  "encryption": "base64 PKCS #10 request"
}
```

Опис полів `requests` :

Поле	Тип	Коли потрібне	Опис
`ecdsa`	string	`pkType=ECDSA`	

Поле	Тип	Коли потрібне	Опис
			Base64 `PKCS #10` request для ECDSA сертифіката.
`signature`	string	`pkType=UA`	Base64 `PKCS #10` request для сертифіката підпису.
`encryption`	string	`pkType=UA` і `certType=SIGN_AND_ENCRYPT`	Base64 `PKCS #10` request для сертифіката шифрування.

RESPONSE

В тілі **відповіді** передається статус 200.

JSON приклад відповіді:

```
{
  "pKey": {
    "id": 1001,
    "name": "Ключ Іваненко",
    "uuid": "019ec000-0000-7000-8000-000000000001",
    "status": "COMPANY_GENERATED",
    "storeType": "HSM",
    "keyType": "UA",
    "stamp": false
  },
  "forms": [
    {
      "type": "PK_FORM",
      "pdf": "JVBERi0x...",
      "hash": "HASH_OF_PDF"
    }
  ]
}
```

Для співробітника з роллю `ADMIN` у відповіді додатково буде `PK_APPENDIX`.

Опис полів відповіді:

Поле	Тип	Опис
`pKey`	object	Об’єкт `ESSPrivateKey`, поля описані в розділі 2.3. Використовуйте `pKey.uuid` у методах 9.2 і 9.3.
`forms`	object[]	PDF-документи, які потрібно підписати співробітником.
`forms[].type`	enum	Тип форми.
`forms[].pdf`	string	PDF content у base64.
`forms[].hash`	string/null	

Поле	Тип	Опис
		Hash PDF для контролю цілісності.

CURL

- **для `store=cloud` :**

```
curl -X POST 'https://host/api/external/company/employee/pkey/generate/draft?...'   -H
'x-system-id: 019eb581-307b-7562-8a1f-20227511e898'   -F 'info={
    "pkName":"Ключ Іваненко",
    "pkType":"UA",
    "pkStoreType":"HSM",
    "pkPassword":"BASE64_RSA_ENCRYPTED_PASSWORD",
    "pkIsStamp":false,
    "emplTitle":"Менеджер",
    "emplOrgUnit":"Відділ продажів",
    "caPassPhrase":"BASE64_RSA_ENCRYPTED_CA_PASSPHRASE",
    "certType":"SIGN_AND_ENCRYPT",
    "certValidity":"TWO"
  }'
```

- **для `store=file` :**

```
curl -X POST 'https://host/api/external/company/employee/pkey/generate/draft?...'   -H
'x-system-id: 019eb581-307b-7562-8a1f-20227511e898'   -F 'info={
    "pkName":"Ключ Іваненко",
    "pkType":"UA",
    "pkStoreType":"FILE",
    "pkIsStamp":false,
    "emplTitle":"Менеджер",
    "emplOrgUnit":"Відділ продажів",
    "caPassPhrase":"BASE64_RSA_ENCRYPTED_CA_PASSPHRASE",
    "certType":"SIGN_AND_ENCRYPT",
    "certValidity":"TWO"
  }'   -F 'requests={
    "signature":"BASE64_P10_SIGNATURE",
    "encryption":"BASE64_P10_ENCRYPTION"
  }'
```

Помилки

HTTP	`type`	Опис
400	`invalid_store`	`store` не `cloud` і не `file`.
400	`employee_not_found`	Співробітник не знайдений.
400	`employee_not_active`	Співробітник не активний або не ідентифікований.
400	`company_not_found`	Не знайдено company для співробітника.

HTTP	`type`	Опис
400	`decrypt_error`	Неможливо розшифрувати `caPassPhrase` або `pkPassword`.
400	`employee_identification_not_found`	Не знайдено підпис ідентифікації співробітника.
400	`request_not_found`	Для `store=file` не переданий потрібний p10-запит.
400	`pk_requests_not_found`	Після створення ключа не знайдені p10-запити.
403	`company_access_denied`	Немає доступу до company.
403	`company_wrong_status`	Компанія не `ACTIVE`.

Revision #4
★Created 2026-06-22 08:45:12 UTC by Наталя Сідненко
Updated 2026-06-28 14:09:32 UTC by Тромбола Євген