**Технічні умови**

**подання банками до Національного банку України**

**звітності про активні операції**

**та ведення Кредитного реєстру Національного банку України** (КР2)

Ці технічні умови (далі - ТУ) визначають вимоги щодо подання банками даних про активні операції згідно з Правилами подання звітності про активні операції, що подаються до Національного банку України (затверджені постановою Правління Національного банку України від 22 січня 2024 року №9, та Положенням про Кредитний реєстр Національного банку України (затверджене постановою Правління Національного банку України від 04 травня 2018 року № 50 зі змінами).

1. Загальні положення
	1. Інформаційний обмін між банком (далі – Банк) і інформаційною системою Національного банку України (далі – Система) накопичення звітності про активні операції банків та ведення Кредитного реєстру здійснюється згідно технології REST API.
	2. Передавання інформації між Банком та Системою здійснюється із використанням механізмів захисту інформації – автентифікації, конфіденційності та контролю цілісності.
	3. Для підтвердження автентичності та цілісності інформації використовується електронний підпис (далі – ЕП), вимоги до якого наведено у розділі 2.
	4. Система проводить автентифікацію та авторизацію Банків використовуючи криптографічний протокол TLS (Transport Layer Security), вимоги до налаштування якого наведено у розділі 3.
	5. Усі повідомлення інформаційного обміну мають формуватися з використанням формату JSON та відповідати формалізованому опису, визначеному у JSON – схемі.
	6. JSON - схеми повідомлень наведені у додатках. У всіх повідомленнях використовується кодування символів UTF-8.
2. Вимоги до забезпечення цілісності інформації
	1. Для виконання операцій накладання та перевірки ЕП, Банк може використовувати бібліотеки СКЗІ НБУ, а саме криптопровайдер NbuCSP, (наданий НБУ, з відповідною документацією), або інші бібліотеки, погоджені до використання з управлінням безпеки інформації Департаменту безпеки НБУ.
	2. ЕП накладається відповідно до вимог описаних в стандарті RFC 7515 – JSON Web Signature (JWS), використовується тип серіалізації Flattened JWS JSON Serialization.

Отриманий JWS JSON об’єкт, має містити такі вкладені об’єкти, що подаються у закодованому за допомогою Base64Url кодування (стандарт RFC 4648) вигляді:

* “payload” – об’єкт в якому передаються дані;
* “protected” – об’єкт, що містить захищений заголовок повідомлення;
* “signature” – значення ЕП.

**Приклад JWS, що використовує Flattened JWS JSON Serialization серіалізацію:**

|  |
| --- |
| {“payload” : ”EFycmF5T2ZURVNUVEFCIHhtbG5zOmk9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4bWxucz0iaHR0cDovL3NjaGVtYXMuZGF0YWNvbnRyYWN0Lm9yZy8yMDA0LzA3L0NyZWRSZWVzdHJfREJNb2RlbCI-PFRFU1RUQUIgaTp0eXBlPSJURVNUVEFC”,“protected” : ”NDwvVEFCX0lEPjxUQUJfVEVYVD7QnNCG0JrQntCb0JDQh9CE0pA8L1RBQl9URVhUPjwvVEVTVFRBQj48VEVTVFRBQiBpOnR5cGU9IlRFU1RUQUIiPjxUQUJ”,“signature” : “NCG0JrQntCb0JDQh9CE0pA8L1RBQl9URVhUPjwvVEVT”} |

Опис реквізитів об’єкта “protected” (в якому передається захищений заголовок повідомлень), що використовуються в Системі (усі реквізити є обов’язковими).

|  |  |  |
| --- | --- | --- |
| Найменування реквізиту | Зміст | Значення |
| alg | Криптографічний алгоритм який використано при створенні JWS об'єкту | RS512 |
| typ | Тип серіалізації JWS | JOSE+JSON |
| cty | Тип контенту JWS об'єкту | application/json |
| x5c | Масив строк, що містить один елемент, а саме: закодоване за допомогою Base64 кодування DER значення сертифіката, яке містить відкритий ключ, відповідний використаному для підпису закритому ключу. | Приклад:[“MIIHBDCCBOygAwIBAgIUBr3………IWD6NEdQyzpoxTCAdVXag56zqjLn”] |
| kid | Ідентифікатор ключа користувача, що був використаний для підпису | XRxxPCRnnn |
| dateOper | Дата створення повідомлення | Дата у форматі D(19)[[1]](#footnote-1) |
| appId | Ідентифікатор інформаційної системи | PCR |

1. Вимоги до використання криптографічного протоколу TLS та відповідних сертифікатів відкритих ключів
	1. Система використовує клієнт-серверну технологію. Банки виступають в якості клієнтів, НБУ в якості серверу.
	2. Клієнт та Сервер (далі – Сторони) для встановлення захищеного з’єднання повинні використовувати криптографічний протокол TLS версії 1.2 або вище. Підтримка протоколу SSL будь якої версії або протоколу TLS версій 1.0 або 1.1 повинна бути відключена.
	3. Пріоритет у виборі криптографічних наборів при встановленні з’єднання визначається Сервером, незалежно від пріоритетів, визначених клієнтом.
	4. Клієнт повинен підтримувати один або кілька криптографічних наборів, визначених у пункті 3.9.
	5. Для шифрування інформації повинні використовуватись симетричні криптографічні алгоритми з довжиною ключа не менше 128 біт.
	6. Довжина відкритого ключа RSA повинна бути не меншою 2048 біт.
	7. Заборонено використовувати криптографічні набори які використовують попередньо узгоджений загальний секретний ключ (PSK).
	8. Для узгодження сеансових ключів використовуються протоколи DHE та ECDHE. Довжина відкритого ключа для протоколу DH повинна бути не меншою 2048 біт. Довжина відкритого ключа для протоколу ECDHE повинна бути не меншою 256 біт.
	9. Сторони можуть використовувати виключно наступні криптографічні набори:

|  |  |
| --- | --- |
| **TLS 1.3** |  |
| 1 | TLS\_AES\_256\_GCM\_SHA384 |
| 2 | TLS\_AES\_128\_СCM\_SHA256 |
| 3 | TLS\_AES\_128\_GCM\_SHA256 |
| **TLS 1.2** |  |
| 1 | ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 |
| 2 | ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 |
| 3 | DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 |
| 4 | ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 |
| 5 | ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 |
| 6 | DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 |

* 1. Для роботи з сервісами прийому запитів на пошук та надання інформації має використовуватись персоніфікований сертифікат «Зовнішній користувач ІС – посадова особа учасника ІС». Генерація сертифіката здійснюється за допомогою МГК. Користувач, що здійснює запити до сервісу, повинен мати відповідні повноваження, що надаються власниками неперсоніфікованих SSL сертифікатів в Банку: адміністраторами облікових записів (власники сертифікатів типу ADU) та адміністраторами керування повноваженнями (власники сертифікатів типу ADR). Для надання необхідних повноважень створено окремий ВЕБ сервіс «Керування повноваженнями зовнішніх користувачів інформаційних систем Національного банку України». Назва ролі, що має бути надана користувачу – CM\_CRK\_SKU. Інструкцію для роботи з сервісом надано в Додатку 1.
1. Номенклатура ключів, що використовуються для подання інформації

При взаємодії між Банком та Системою, Банк має використовувати два ключі (які мають різні ідентифікатори), перший для накладання ЕП на повідомлення, що формуються, другий для шифрування SSL-з’єднання між Банком та Системою, наприклад, для учасника з 3-символьним ідентифікатором Rxx ідентифікатори користувача СКЗІ НБУ матимуть вигляд XRxxPCR000 та XRxxPCR001.

Ідентифікатори ключів формуються за маскою:

X**Rxx**PCR0**nn**

де:

X – константа, ознака зовнішнього користувача інформаційної мережі НБУ;

***Rxx*** – унікальний ідентифікатор Учасника в СКЗІ НБУ;

PCR – константа, ідентифікатор інформаційної системи;

***nn*** – порядковий номер робочого місця, що взаємодії з Системою (діапазон значень від 00 до 19).

Якщо Банк використовує більш ніж одне робоче місце, кожне з них використовує власні ідентифікатори ключів, які відрізняються номером робочого місця.

Ідентифікатор ключа, сформований за даними правилами, вміщується в реквізит "Common Name" поля сертифіката Subject та в розширення сертифікату "Subject Alternative Names” з OID 1.3.6.1.4.1.311.20.2.3 (User Principal Name (UPN)).

У разі генерації нового сертифікату відкритого ключа попередній сертифікат з тим самим ідентифікатором скасовується, і таким чином новий сертифікат "заміщує" старий.

1. Порядок забезпечення ключовою системою

Генерація таємного ключа для робочого місця здійснюється засобами модуля генерації ключів (МГК), згідно до п.3 Інструкції з використання програмно-апаратного модулю генерації криптографічних ключів СЕП та інформаційних задач для отримання «SSL-сертифікатів».

Для забезпечення процесу генерації/сертифікації ключів засобами МГК для тестового середовища використовується сервер тестового АЦСК НБУ, для продуктивного – сервер АЦСК НБУ.

Кореневі сертифікати АЦСК НБУ (тестового АЦСК НБУ) розсилаються централізовано або їх можна скачати з відповідних сайтів АЦСК (тестового АЦСК) НБУ.

1. Опис роботи Системи
	1. Система використовує клієнт-серверну технологію. Банки відправляють та отримують інформацію за допомогою HTTP-запитів (POST, GET) до відповідних вебсервісів.
	2. Для тестування і відпрацювання технології інформаційної взаємодії між системами автоматизації банків з Системою розгорнуто тестові вебсервіси.
	3. Взаємодія з вебсервісами відбувається за адресами: **https://app-pcr01** – для продуктивного середовища і **https://app-pcr-stend01**– для тестового (далі по тексту замість зазначених конкретних адрес використовується змінна **cr\_server**).
	4. Для коректної роботи з вебсервісами у файлі «c:\Windows\System32\drivers\etc\hosts» необхідно забезпечити наявність таких рядків:
* «xxx.xxx.xxx.xxx app-pcr01»
* «xxx.xxx.xxx.xxx app-pcr-stend01».
	1. В Системі прийняті наступні домовленості:

- запит типу POST використовується для подання інформації;

- запит типу GET використовується для отримання інформації;

1. Подання інформації до Системи
	1. Банки надають інформацію двома типами повідомлень: загальний та груповий пакет.
	2. На стороні Системи для взаємодії з Банками налаштовано такі вебсервіси: 1) вебсервіс «Пакет» (загальний пакет), 2) вебсервіс «Групи» (груповий пакет), 3) вебсервіс «Перевірки статусу».
	3. Банк формує повідомлення та надсилає його у запиті типу POST на відповідний вебсервіс.
	4. Запит, що надходить до вебсервісу, проходить декілька етапів обробки, на яких він може бути відхилений:

1) автентифікація, авторизація, перевірка цілісності інформації (верифікація електронного підпису), перевірка поданої інформації в запиті на відповідність JSON-схемі;

2) перевірка поданої інформації в запиті на відповідність правилам контролю, викладеним у (посилання на контролі).

* 1. Якщо запит не пройде перевірки першого етапу, то Банк одразу у відповіді на запит отримає повідомлення про помилку і сеанс завершується. Усю інформацію у запиті буде відхилено (див. приклад нижче).

**Приклад відповіді на запит типу POST у разі неуспішної обробки на першому етапі (містить код стану HTTP та короткий опис)**

|  |
| --- |
| **HTTP/**1.1 422 JSON Schema Validation ErrorContent-Type: application/jsonTransactionId: 452304SessionId: 332eb9e3b9f64b7af1570651f8aac8d27f354470612e03d25bba38d48dd678feX-Backside-Transport: FAIL FAILConnection: close{"errMessage":"[JSV0013] Invalid value: the value is not among the permitted enumerated values."} |

Короткий текстовий опис у відповіді може відрізнятись від наведеного в прикладі.

* 1. Якщо запит успішно пройде перевірки першого етапу, то одразу у відповіді на запит (тобто у поточному сеансі) отримає повідомлення з унікальним ідентифікатором пакету {package\_id} і сеанс завершується.
	2. Отримання Банком ідентифікатору пакету свідчить про початок другого етапу перевірки.
	3. Другий етап перевірки відбувається в асинхронному режимі. Подальший результат обробки пакету на другому етапі перевірки Банк може дізнатись, надіславши запит (тип GET) на вебсервіс «Перевірка статусу», використовуючи ідентифікатор пакету (див. приклад нижче).
	4. Від вебсервісу «Перевірка статусу» Банк може отримати такі варіанти відповіді (де змінна {package\_id} є значенням Ідентифікатору пакету):
* **NotFound** – це говорить про те, що повідомлення за ідентифікатором {package\_id} не знайдено або вказаний ідентифікатор є невірним;
* **Passed** – це говорить про те, що повідомлення за ідентифікатором {package\_id} оброблено успішно, інформацію прийнято.
* **Failed** – це говорить про те, що повідомлення за ідентифікатором {package\_id} оброблено з помилками, інформацію не прийнято; (відповідь також буде включати перелік перших 10 помилок).
1. Загальна структура повідомлень, що надходять від Банка до Системи.

**Повідомлення (запит) типу POST.**

**Подання інформації (де xxxx – сервіс main\_pkg або group\_pkg)**

|  |
| --- |
| **POST** **cr\_server/cr2\_banks/api/v1/xxxx** Content-Type: application/json{“payload” : ”EFycmF5T2ZURVNUVEFCIHhtbG5zOmk9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4bWxucz0iaHR0cDovL3NjaGVtYXMuZGF0YWNvbnRyYWN0Lm9yZy8yMDA0LzA3L0NyZWRSZWVzdHJfREJNb2RlbCI-PFRFU1RUQUIgaTp0eXBlPSJURVNUVEFC”,“protected” : ”NDwvVEFCX0lEPjxUQUJfVEVYVD7QnNCG0JrQntCb0JDQh9CE0pA8L1RBQl9URVhUPjwvVEVTVFRBQj48VEVTVFRBQiBpOnR5cGU9IlRFU1RUQUIiPjxUQUJ”,“header” : null,“signature” : “NCG0JrQntCb0JDQh9CE0pA8L1RBQl9URVhUPjwvVEVT”} |

**Повідомлення (запит) типу GET.**

**Перевірка статусу обробки повідомлення (де змінна {package\_id} є значенням ідентифікатору запиту, що надається у відповідь в разі успішного проходження перевірок першого етапу).**

|  |
| --- |
| **GET** **cr\_server/cr2\_status/api/v1/request\_status/{package\_id}** Content-Type: application/json |

**Повідомлення (запит) типу GET. (Реалізація на наступному етапі)**

**Запит ключових реквізитів наборів даних person\_full або person\_short, або liability, або loan, або collateral, або contract, або group (де xxxx – person\_full або person\_short, або liability, або loan, або collateral, або contract, або group; nnnn – ідентифікатор person\_full/person\_short/liability/loan/collateral/contract/group).**

|  |
| --- |
| **GET** **cr\_server/cr2\_banks/api/v1/xxxx/nnnn** Content-Type: application/json |

1. Опис форматів даних, що використовуються у повідомленнях.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип даних** | **Формат даних** | **Опис вимог до подання типу даних** | **Можливі допустимі значення** |
| Символьний **String** | C(10), або С(3:25) | Набуває значення набір (рядок) символів у вигляді а) C(Y) де Y-максимальна кількість символів, або C(X:Y) де X-мінімальна кількість символів,Y-максимальна кількість символів | “asdf” “ddf%#\_ “ “         “ “23456” “1900-02-15” “0” null. |
| Числовий **Number** | N(2) або N(7,4) | Набуває значення «число» у вигляді N(2), де 2-ціле число де кількість цифр у числі не більше 2 або N(7,4) де 7- ціле число де кількість цифр у числі не більше 7 та кількість цифр у числі після крапки не більше 4. | 1 2 324 -526 125.25 0.0000578, додатково може бути розширене значенням null. |
| Логічний **Boolean** | B | Набуває одного з визначеного переліку значень true або false | true, false, додатково може бути розширене значенням null. |
| Дата, ISO 8601 **Date** | D(10) | Набуває значення у форматі “YYYY-MM-DD”, де: YYYY – рік (чотири цифри), MM –місяць (дві цифри), DD – день (дві цифри). Наприклад: 2015-05-19 означає 19 травня 2015 року | “1900-02-15” “2000-08-24”, додатково може бути розширене значенням null. |
|
| Комбіновані дата й час, ISO 8601 **Date** | D(19) | Набуває значення у форматі “yyyy-MM-ddTHH:mm:ss ”, де: yyyy – рік (чотири цифри), MM – місяць (дві цифри), dd – день (дві цифри), HH –годин,mm – хвилин, ss – секунд Наприклад: 2015-05-19 означає 19 травня 2015 року 2015-05-19T15:22:37 | “2015-05-19T15:22:37”, в деяких випадках може містити долі секунд [sssss], додатково може бути розширене значенням null. |

1. Загальний опис інформації та структура повідомлень, що отримує Банк у відповідь на свій запит.
	1. Інформацію про відповіді від вебсервісу, що може отримати клієнт у разі не успішної обробки запиту в залежності від етапу наведено в Додатку 2.
	2. Інформація про відповіді від вебсервісу у разі успішної обробки запиту в залежності від типу запиту:

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип запиту** | **Опис запиту** | **Код стану HTTP** | **Інформація про відповідь** |
| GET **(Реалізація на наступному етапі)** | Запит на отримання інформації про дані надіслані банком | 200 | Відповідь містить код стану HTTP та інформацію про дані надіслані банком у вигляді підписаного JWS об’єкта. |
| GET за конкретним id **(Реалізація на наступному етапі)** | Запит на отримання інформації за конкретним id | 200 | Відповідь містить код стану HTTP та інформацію про конкретний запис у вигляді підписаного JWS об’єкта. |
| POST | Створення нового запису | 201 | Відповідь містить код стану HTTP та квитанцію з результатом виконання операції; |

**Приклад відповіді на запит типу POST від вебсервісу «Пакет» в разі успішної обробки на першому етапі**

|  |
| --- |
| **HTTP/**1.1 201 OKX-Backside-Transport: OK OKConnection: Keep-AliveTransfer-Encoding: chunkedTransactionId: 297312SessionId: f21fb933e1845d028ec776958b67705d7fc5d696434834f6002743814cec1d66X-Global-Transaction-ID: 055a38136548fbdf00048960Content-Type: application/json{"package\_id":"f21fb933e1845d028ec776958b67705d7fc5d696434834f6002743814cec1d66","web\_service\_url":"https://172.22.3.17:8160/cr2\_banks/api/v1/main\_pkg","http\_method":"POST","client\_ip":"10.81.17.60","client\_ssl\_cert\_id":"XY12PCR001","client\_sign\_cert\_id":"XY12PCR002","general\_http\_status\_code":201,"kvi\_date":"2023-11-06T14:44:47.587Z"} |

* 1. Реквізити відповіді за результатами роботи відповідного сервісу (запит типу POST):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №з/п | Зміст реквізиту | Найменування реквізиту  | Тип (макс.дов-жина) | Обов’язковість заповнення | Примітки |
| 1 | Унікальний ідентифікатор пакету, що було надіслано Банком | package\_id | С(64) | + |  |
| 2 | Адреса вебсервіса, яким було оброблено інформацію | web\_service\_url | C(100) | + |  |
| 3 | Http метод запиту | http\_method | C(15) | + |  |
| 4 | IP адреса з якої було здійснено запит | client\_ip | C(15) | + |  |
| 5 | Ідентифікатор сертифіката, який був використаний для встановлення захищеного з’єднання | client\_ssl\_cert\_id | C(10) | + |  |
| 6 | Ідентифікатор ключа користувача, що був використаний для підпису | client\_sign\_cert\_id | C(10) | + |  |
| 7 | Код стану HTTP | general\_http\_status\_code | N(3) | + | Частина першого рядка відповіді сервера при запитах по протоколу HTTP (HttpStatusCode). |
| 9 | Дата та час формування квитанції | kvi\_date | D(19) | + | Дата та час у форматі YYYY-MM-DDThh:mm:ss[.sssss] |

**Приклад відповіді на запит типу GET від вебсервісу «Перевірка статусу» в разі обробки повідомлення з помилками**

|  |
| --- |
| **HTTP/**1.1 424X-Backside-Transport: OK OKConnection: Keep-AliveTransfer-Encoding: chunkedTransactionId: 297222SessionId: f21fb933e1845d028ec776958b67705d7fc5d696434834f6002743814cec1d66X-Global-Transaction-ID: 055a38136548fbdf00048960Content-Type: application/json{  "status": "Failed",  "package\_id": "3ebf12de-9803-418d-88a8-54882d2a9e1c", "response\_timestamp": "2023-11-10T10:00:20.7276661Z",  "control\_errors": [    {      "error\_number": 1,      "error\_id": "00008:01.03",      "error\_code": "00008",      "error\_nesting": [        {          "data\_set\_name": "person\_full",          "data\_set\_index": 1,          "data\_set\_id": "111"        },        {          "data\_set\_name": "address",          "data\_set\_index": 0,          "data\_set\_id": null        }      ]    },    {      "error\_number": 2,      "error\_id": "05112:01.04",      "error\_code": "05112",      "error\_nesting": [        {          "data\_set\_name": "person\_full",          "data\_set\_index": 1,          "data\_set\_id": "111"        },        {          "data\_set\_name": "rating",          "data\_set\_index": 5,          "data\_set\_id": null        }      ]    }  ]} |

**Приклад відповіді на запит типу GET від вебсервісу «Перевірка статусу» в разі, коли повідомлення за ідентифікатором пакету не знайдено**

|  |
| --- |
| **HTTP/**1.1 404X-Backside-Transport: OK OKConnection: Keep-AliveTransfer-Encoding: chunkedTransactionId: 297222SessionId: f21fb933e1845d028ec776958b67705d7fc5d696434834f6002743814cec1d66X-Global-Transaction-ID: 055a38136548fbdf00048960Content-Type: application/json{  "status": "NotFound",  "package\_id": "3ebf12de-9803-418d-88a8-54882d2a9e1c", "response\_timestamp": "2023-11-10T10:00:20.7276661Z"} |

**Приклад відповіді на запит типу GET від вебсервісу «Перевірка статусу» в разі успішної обробки повідомлення (інформацію прийнято)**

|  |
| --- |
| **HTTP/**1.1 200X-Backside-Transport: OK OKConnection: Keep-AliveTransfer-Encoding: chunkedTransactionId: 297222SessionId: f21fb933e1845d028ec776958b67705d7fc5d696434834f6002743814cec1d66X-Global-Transaction-ID: 055a38136548fbdf00048960Content-Type: application/json{  "status": "Passed",  "package\_id": "3ebf12de-9803-418d-88a8-54882d2a9e1c", "response\_timestamp": "2023-11-10T10:00:20.7276661Z"} |

* 1. Реквізити відповіді за результатами роботи вебсервісу «Перевірка статусу»:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №з/п | Зміст реквізиту | Найменування реквізиту  | Тип (макс.дов-жина) | Обов’язковість заповнення | Примітки |
| 1 | Статус обробки пакету | status | С(8) | + | Набуває одне із значень:**NotFound****Passed****Failed** |
| 2 | Унікальний ідентифікатор пакету, що було надіслано Респондентом | package\_id | С(64) | + |  |
| 3 | Дата та час формування відповіді | response\_timestamp | D(19) | + | Дата та час у форматі YYYY-MM-DDThh:mm:ss[.sssss] |
| 4. Перелік помилок. Структура (масив) **control\_errors**. Якщо помилок кілька, то поля з 4.1. по 4.3. повторюються та є обов’язковими. Якщо помилок немає, структура (масив) **control\_errors** не зазначається. |
| 4.1 | Порядковий номер помилки | error\_number | N(2) | + |  |
| 4.2 | Узагальнений ідентифікатор помили відповідно до опису Контролів (посилання на Контролі) | error\_id | С(64) | + |  |
| 4.3 | Код помилки, відповідно до опису Контролів (посилання на Контролі) | error\_code | С(64) | + |  |
| 4.4 Реквізити, в яких допущенні помилки. Структура (масив) **error\_nesting**. Якщо реквізитів, в яких допущенні помилки кілька, то поля з 4.4.1. по 4.4.3 повторюються та є обов’язковими. Якщо помилкові реквізити відсутні, структура (масив) **error\_nesting** не зазначається. |
| 4.4.1 | Назва реквізиту | data\_set\_name | С(64) | + |  |
| 4.4.2 | Порядковий номер елементу в масиві, якщо реквізит, в якому допущено помилку є масивом  | data\_set\_index | N(2) | + | Якщо реквізит, в якому допущено помилку не є масивом, то значення data\_set\_index дорівнює 1. |
| 4.4.3 | Ідентифікатор в наборі даних, що включає реквізит, в якому допущено помилку | data\_set\_id | С(64) | + | Якщо набір даних немає ідентифікатора, то значення data\_set\_id дорівнює null. |

1. Правила формування повідомлень, що надсилаються банками.
	1. Усі вхідні повідомлення мають відповідати ієрархічній структурі (Файл - Пакети для JSON.xlsx).
	2. Існують два види вхідних повідомлень: Packet та GroupPacket. Повідомлення подаються Банком на відповідні вебсервіси «Пакет» (Загальний пакет) та «Групи» (Груповий пакет).
	3. Перед відправкою повідомлень, вони мають бути перевірені та валідовані Банком за JSON-схемою для запобігання надання некоректних даних.
	4. В кожному повідомлені обов’язково присутній реквізит «Звітна дата» (reporting\_date, ID - 0015);
	5. «Звітна дата» – дата, на яку складається звітність. Звітність складається щомісячно станом на 00 годин 00 хвилин першого числа місяця, наступного за звітним;
	6. Повідомлення Packet може складатися із наборів даних верхнього рівня ID01.Особа (розширені відомості) (person\_full), ID02.Особа (скорочені відомості) (person\_short), ID03.Фінансове зобов'язання (liability), ID04.Активна операція (loan), ID05.Забезпечення (collateral),  ID06.Узагальнююча угода (contract).
	7. Повідомлення Груповий пакет складається з наборів даних верхнього рівня ID07.ГСК/ГПК (group) та ID02.Особа (скорочені відомості) (person\_short).
	8. Будь-яке повідомлення Packet має обов'язково містити набір даних ID01.Особа (розширені відомості) (person\_full) та хоча б один з наборів даних ID03.Фінансове зобов'язання (liability) або ID04.Активна операція (loan).
	9. Будь-яке повідомлення GroupPacket має обов’язково містити набори даних ID07.ГСК/ГПК (group) та «ID02.Особа (скорочені відомості) (person\_short).
	10. Для однозначної ідентифікації екземпляру набору даних Банком формуються ідентифікатори, які не можуть бути змінені у часі. Банком має бути забезпечена унікальність ідентифікаторів в межах його облікової системи. Ідентифікатор може набувати символьне значення від 5 символів до 50-ти з можливістю використання цифр, великих латинських літер і дефісу (без пробілів і спеціальних символів). Дублювання ідентифікаторів в рамках одного набору даних вхідного повідомлення не дозволяється.
	11. Набори даних верхнього рівня мають бути пов’язані між собою через ідентифікатори та можуть містити притаманні їм вкладені набори даних. При цьому має виконуватися наступна умова: всі пов'язані ідентифікаторами набори даних мають обов'язково міститися в одному вхідному повідомленні.
	12. Набор даних верхнього рівня ID01.Особа (розширені відомості) (person\_full) має обов'язково містити один з вкладених наборів даних: ID34.Фізична особа, резидент (ind\_person), ID35.Юридична особа, резидент (entity), ID36.Фізична особа, нерезидент (non\_res\_ind\_person), ID37.Юридична особа, нерезидент (non\_res\_entity). Набор даних верхнього рівня ID02.Особа (скорочені відомості) (person\_short) має обов'язково містити один з вкладених наборів даних ID30.Фізична особа (скорочені відомості) (ind\_person\_short) або ID31.Юридична особа (скорочені відомості) (entity\_short). Набор даних верхнього рівня ID05.Забезпечення (collateral) має містити один або більше з наборів даних ID40.Об’єкт рухомого майна (movable), ID41.Об’єкт нерухомого майна (immovable), ID42.Фінансове забезпечення (deposit).
	13. В повідомленнях існують реквізити, значення яких може бути різним за типом даних. Для реалізації можливості вибору варіанту типу значень для таких реквізитів використовуються перемикачі-об’єднувачі. Наприклад, для усіх реквізитів, в яких крім реального значення може бути зазначений код з довідника причин неподання (довідник F170) використовуються перемикачі-об’єднувачі (див. JS\_Main\_Bank.json).
2. JSON схеми повідомлень в Системі
	1. У Системі повідомлення формуються в JSON форматі відповідно до JSON схем.
	2. В Системі використовуються дві головні JSON схеми для загального та групового пакету, а також додаткові JSON схеми для формування значень довідників, на які є посилання з головних схем.
	3. Перевірка повідомлень, що використовуються в Системі, виконується на стороні Системи. У разі отримання від Банка повідомлення, що не пройшло перевірку на відповідність до JSON схеми, Банк отримує відповідь з кодом стану HTTP 422 та коротким описом помилки у форматі JSON.
	4. Для отримання переліку JSON схем, що використовуються Системою, Банк має здійснити HTTP запит типу GET за URL, що вказаний в п.12.7. Перелік JSON схем передається від Системи в тілі відповіді (Response body) у вигляді масиву JSON об’єктів.
	5. Реквізити, які містить окремий JSON об’єкт переліку JSON схем (елемент масиву):

|  |  |  |  |
| --- | --- | --- | --- |
| №з/п | Зміст реквізиту | Найменування реквізиту | Тип (максимальна довжина) |
| 1 | Ім’я файлу JSON схеми | name | C(25) |
| 2 | Розмір файлу JSON схеми в байтах | size | N |
| 3 | Дата та час модифікації файлу JSON схеми | modified | D(19) |
| 4 | Посилання, що може бути використане для отримання файлу JSON схеми | url | C(50) |

* 1. Для отримання конкретної схеми Учасник має здійснити HTTP запит типу GET за URL, який міститься в раніше отриманому переліку схем в значенні реквізиту «url» елемента, що містить інформацію про необхідний файл.
	2. Перелік URL, що використовуються в Системі (де **cr\_server** приймає значення відповідно до п.6.3):

|  |  |  |
| --- | --- | --- |
| Параметр | Опис | Значення |
| URL передачі даних (Packet) до Системи | п.7.3  | **cr\_server**:8160/cr2\_banks/api/v1/main\_pkg |
| URL передачі даних (GroupPacket) до Системи | п.7.3 | **cr\_server**:8160/cr2\_banks/api/v1/groups\_pkg |
| URL для здійснення запиту про статус обробки повідомлення | п.7.8  | **cr\_server**:8168/cr2\_status/api/v1/request\_status/{package\_id} – де змінна {package\_id} є значенням Ідентифікатору пакету. |
| URL для отримання інформації про фізичну особу боржника |  | **cr\_server**:81??/cr2\_banks/api/v1/… |
| URL для отримання інформації про юридичну особу боржника |  | **cr\_server**:81??/cr2\_banks/api/v1/… |
| URL для отримання переліку JSON схем |  |  [**cr\_server:**8162/cr2\_banks/jsonschema](https://172.22.12.34:8034/jsonschema) |

1. Інформація, яка надходить від Системи на запити від банків (GET) щодо отримання даних по своїй установі

Розділ в розробці.

1. Надання інформації Банку з Системи щодо Боржників, які наявні у кредитному реєстрі.

Розділ в розробці.

**Додаток 1**

**ІНСТРУКЦІЯ**

**для роботи з сервісом**

**«Керування повноваженнями зовнішніх користувачів**

**інформаційних систем Національного банку України»**

Ця інструкція визначає порядок роботи з сервісом «Керування повноваженнями зовнішніх користувачів інформаційних систем Національного банку України» ( далі сервіс).

1. Загальні положення
	1. Інформаційний обмін між банком (далі – Банк) і сервісом Національного банку України (далі – НБУ) здійснюється згідно технології RESTful веб-API.
	2. Передавання інформації між Банком та НБУ повинно здійснюватися із використанням механізмів захисту інформації – автентифікації, конфіденційності та контролю цілісності.
	3. НБУ проводить автентифікацію та авторизацію Банків використовуючи криптографічний протокол TLS (Transport Layer Security), вимоги до налаштування якого наведено нижче.
	4. Сервіс дає можливість адміністраторам, працівникам банку самостійно керувати повноваженнями своїх користувачів в інформаційних системах НБУ відповідно до інструкцій інформаційних систем.
	5. Користувачами сервісу є співробітники банку, власники неперсоніфікованих SSL сертифікатів типу ADU – адміністратори облікових записів та типу ADR – адміністратори керування повноваженнями відповідно. Не дозволяється суміщення ролей, тобто, відповідальними за сертифікати мають бути різні люди.

1. Вимоги до використання криптографічного протоколу TLS та відповідних сертифікатів відкритих ключів
	1. НБУ та Банк для встановлення захищеного з’єднання повинні використовувати криптографічний протокол TLS не нижче версії 1.2, а також відповідні особисті ключі та сертифікати відкритих ключів.
	2. У протоколі TLS допускаються різні криптографічні набори.
	3. Криптографічний набір узгоджується між Банком та сервером НБУ (далі – Сервер) під час встановлення з’єднання. Банк передає Серверу список підтримуваних криптографічних наборів, а Сервер вибирає один з них для захисту інформації.
	4. Не повинні використовувати криптографічні набори, які не використовують шифрування або використовують для шифрування алгоритм RC4 (в якості EncryptionAlg встановлено NULL або RC4).
	5. Для шифрування інформації повинні використовуватись симетричні криптографічні алгоритми з довжиною ключа не менше 128 біт.
	6. Довжина відкритого ключа RSA повинна бути не меншою 2048 біт.
	7. Заборонено використовувати криптографічні набори які використовують попередньо узгоджений загальний секретний ключ (PSK).
	8. Для узгодження сеансових ключів використовуються протоколи DHE та ECDHE. Довжина відкритого ключа для протоколу DH повинна бути не меншою 2048 біт. Довжина відкритого ключа для протоколу ECDHE повинна бути не меншою 256 біт.
	9. Рекомендується використовувати такі криптографічні набори:

TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256;

TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256;

TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384;

TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384.

* 1. Генерація сертифікатів таємного ключа адміністраторів здійснюється за допомогою ПМГК, згідно вимогам Департаменту безпеки НБУ. Сервіс використовує неперсоніфіковані SSL сертифікати наступних типів: ADU – адміністратор облікових записів та ADR – адміністратор керування повноваженнями.

1. Налаштування робочого місця для роботи з сервісом
	1. Для коректної роботи сервісу потрібно забезпечити наявність рядку «172.22.200.50 app-certman01» (якщо в мережевій інфраструктурі банку адреса буде відрізнятися від наведеної, то потрібно прописати правильну адресу) в файлі «c:\Windows\System32\drivers\etc\hosts».
	2. Для роботи сервісу з комп’ютера повинен бути доступ до сервера за адресою 172.22.200.50 за портом 443.
2. Опис роботи сервісу
	1. Сервіс використовує клієнт-серверну технологію. Банки відправляють та отримують інформацію за допомогою HTTP-запитів (GET, POST, PUT, DELETE) до відповідних веб-сервісів.
	2. Згідно рекомендацій щодо використання RESTful веб-API прийняті наступні домовленості щодо типів HTTP запитів:
		* GET використовується для отримання інформації;
		* POST- для додавання нового запису;
		* PUT - для редагування запису;
		* DELETE - для видалення інформації.
	3. Документація щодо роботи сервісу в форматі Swagger міститься за наступним посиланням [https://app-certman01/certManager/swagger](https://172.22.0.11/certManager/swagger).
	4. Сервіс дає можливість користувачам, власникам SSL сертифікатів типу ADU та ADR, створювати нові облікові записи, видаляти їх, надавати, або відкликати повноваження співробітникам банку – користувачам інформаційних систем НБУ, що є власниками персоніфікованих сертифікатів, отриманих за допомогою ПМГК. Ім’я облікового запису надається відповідно до значення параметру CN (далі – ідентифікатор персоніфікованого сертифікату користувача) сертифікату співробітника банка. Керування повноваженнями здійснюється шляхом включення раніше створеного облікового запису до необхідної ролі, або виключення з неї.
	5. Для роботи з сервісом можливо використовувати програмне забезпечення, що дає змогу здійснювати HTTP запити за протоколом SSL, на кшталт Postman, curl та інші. Також, для спрощення виконання запитів розроблено ВЕБ сторінки, які надають можливість проведення базових операцій з сервісом. Сторінка для роботи з обліковими записами розміщено за посиланням: [https://app-certman01/certManagerApp/AdmUsers](https://172.22.0.11/certManagerApp/AdmUsers), для здійснення операцій з керування повноваженнями - за посиланням: [https://app-certman01/certManagerApp/AdmRoles](https://172.22.0.11/certManagerApp/AdmRoles). Для роботи з ВЕБ сторінками рекомендовано використовувати такі браузери: Google Chrome, Microsoft Edge абоMicrosoft Internet Explorer.
3. Повноваження адміністраторів
	1. Обидва адміністративних записи мають повноваження на перегляд документації сервісу в форматі Swagger та здійснення запитів на отримання інформації, типу GET (перегляд переліку користувачів та ролей).
	2. Адміністратор облікових записів додатково до повноважень наведених в п. 4.1, може здійснювати операції створення та видалення облікових записів користувачів, запити типу POST та DELETE.
	3. Адміністратор керування повноваженнями додатково до повноважень наведених в п. 4.1, може здійснювати операції на включення певного облікового запису до ролі або виключення з неї, запити типу PUT та DELETE.
4. Робота з обліковими записами користувачів (Адміністратор облікових записів)

Робота з обліковими записами користувачів здійснюється за допомогою наступних типів запитів:

* 1. GET без параметрів – отримання всіх раніше створених облікових записів, url - для здійснення запитів: [https://app-certman01/certManager/api/AdmUsers](https://172.22.0.11/certManager/api/AdmUsers);
	2. GET з параметром, в якості параметра використовується ідентифікатор персоніфікованого сертифіката користувача – отримання інформації про ролі, призначені користувачу, url для здійснення запиту: [https://app-certman01/certManager/api/AdmUsers/{certificateId}](https://172.22.0.11/certManager/api/AdmUsers/%7BcertificateId%7D), де certificateId - ідентифікатор персоніфікованого сертифіката користувача;
	3. POST - запит на створення нового облікового запису відповідно до ідентифікатора персоніфікованого сертифіката користувача, url - для здійснення запиту: [https://app-certman01/certManager/api/AdmUsers/{certificateId}](https://172.22.0.11/certManager/api/AdmUsers/%7BcertificateId%7D), де certificateId - ідентифікатор персоніфікованого сертифіката користувача;
	4. DELETE – запит на видалення інформації про облікових запис користувача відповідно до ідентифікатора персоніфікованого сертифіката користувача, url - для здійснення запиту: [https://app-certman01/certManager/api/AdmUsers/{certificateId}](https://172.22.0.11/certManager/api/AdmUsers/%7BcertificateId%7D), де certificateId - ідентифікатор персоніфікованого сертифіката користувача.
1. Робота з ролями (Адміністратор керування повноваженнями)

Робота з повноваженнями користувачів здійснюється за допомогою наступних типів запитів:

* 1. GET без параметрів – отримання доступних для призначення ролей, url - для здійснення запиту: [https://app-certman01/certManager/api/AdmRoles](https://172.22.0.11/certManager/api/AdmRoles);
	2. GET з параметром, в якості параметра використовується назва ролі – отримання інформації про всіх користувачів банку, яким призначено роль, url - для здійснення запиту: [https://app-certman01/certManager/api/AdmRoles/{roleName}](https://172.22.0.11/certManager/api/AdmRoles/%7BroleName%7D), де roleName  – назва ролі;
	3. PUT – запит на призначення ролі користувачу, використовується з двома параметрами, ідентифікатор персоніфікованого сертифіката користувача та назва ролі, url - для здійснення запиту: [https://app-certman01/certManager/api/AdmRoles/{certificateId}/{roleName}](https://172.22.0.11/certManager/api/AdmRoles/%7BroleName%7D), де certificateId - ідентифікатор персоніфікованого сертифіката користувача, roleName - ідентифікатор персоніфікованого сертифіката користувача;
	4. DELETE – запит на відкликання ролі в облікового запису користувача, що використовується з двома параметрами: ідентифікатор персоніфікованого сертифіката користувача та назва ролі, url - для здійснення запиту: [https://app-certman01/certManager/api/AdmRoles/{certificateId}/{roleName}](https://172.22.0.11/certManager/api/AdmRoles/%7BroleName%7D), де certificateId - ідентифікатор персоніфікованого сертифіката користувача, roleName - ідентифікатор персоніфікованого сертифіката користувача.
1. Робота з обліковими записами користувачів за допомогою відповідної ВЕБ сторінки.

Для роботи з ВЕБ сторінкою має використовуватись сертифікат типу ADU, адміністратор облікових записів. Сторінку розміщено за посиланням: [https://app-certman01/certManagerApp/AdmUsers](https://172.22.0.11/certManagerApp/AdmUsers).



* 1. Перелік раніше створених користувачів банку відображено в списку 5;
	2. Створення нового облікового запису виконується шляхом введення ідентифікатора персоніфікованого сертифіката користувача в поле 1 і натисканням кнопки 3 після цього;
	3. Видалення існуючого облікового запису виконується шляхом вибору ідентифікатора персоніфікованого сертифіката користувача в списку 5 і натисканням кнопки 4 після цього.
1. Робота з ролями за допомогою відповідної ВЕБ сторінки.

Для роботи з ВЕБ сторінкою має використовуватись сертифікат типу ADR, адміністратор керування повноваженнями. Сторінку розміщено за посиланням [https://app-certman01/certManagerApp/AdmRoles](https://172.22.0.11/certManagerApp/AdmRoles).



* 1. Перелік раніше створених користувачів банку відображено в списку 1;
	2. Перелік ролей (повноважень) доступних до призначення відображено в списку 3;
	3. Перелік ролей для обраного облікового запису відображено в списку 2;
	4. Для призначення необхідних повноважень необхідно виконати наступні діїї
* Обрати обліковий запис зі списку 1;
* Обрати необхідну роль зі списку 2;
* Натиснути кнопку 5.
	1. Для відкликання ролі неохідно виконати наступні дії:
* Обрати обліковий запис зі списку 1;
* Обрати роль яку необхідно відкликати зі списку 3;
* Натиснути кнопку 4.

Додаток 2

Інформація про відповіді від веб-сервісу, що може отримати клієнт у разі не успішної обробки запиту в залежності від етапу

|  |  |  |  |
| --- | --- | --- | --- |
| **Етап проходження запиту** | **Опис помилки** | **Код стану HTTP** | **Інформація про відповідь** |
| Автентифікація | Відсутній сертифікат користувача | 403 | Відповідь містить код стану HTTP та короткий опис |
| Автентифікація | Сертифікат виданий невідомим центром сертифікації | 403 | Відповідь містить код стану HTTP та короткий опис |
| Автентифікація | Термін дії сертифікату закінчився  | 403 | Відповідь містить код стану HTTP та короткий опис |
| Автентифікація | Сертифікат відкликано, або неможливо це перевірити | 403 | Відповідь містить код стану HTTP та короткий опис |
| Авторизація | Сертифікат валідний, але призначений для іншої інформаційної системи | 401 | Відповідь містить код стану HTTP та короткий опис |
| Авторизація | Сертифікат валідний, але має не вірний тип | 401 | Відповідь містить код стану HTTP та короткий опис |
| Технологічна перевірка | Не вірна адреса на яку здійснюється запит | 404 | Відповідь містить код стану HTTP та короткий опис |
| Технологічна перевірка | Вхідні дані не є валідним JSON об’єктом | 415 | Відповідь містить код стану HTTP та короткий опис |
| Технологічна перевірка | Вхідний JSON об’єкт не відповідає вимогам, що наведені в JSON схемі | 422 | Відповідь містить код стану HTTP та короткий опис |
| Технологічна перевірка | Підпис або дані пошкоджено, підпис не є дійсним | 412 | Відповідь містить код стану HTTP та короткий опис |
| Помилка під час обробки запиту | Під час обробки запиту виникла помилка | 500 | Короткий опис помилки |
| Сервіс недоступний | Проводяться технічні роботи по оновленню, або обслуговуванню сервісів. | 503 | Короткий опис помилки |

1. Комбіновані дата й час, ISO 8601 Date [↑](#footnote-ref-1)