



Java Task | Online Round

Level: **Standard / Pro**

Duration = **6 hours**

Deadline: **24.03.2019**

У супер секретного поштовика "Роги та Копита" реєстрація акаунтів прив'язана до мобільних телефонів користувачів, які зберігаються в системі. Це зроблено для того, щоб спамери не могли створювати декілька акаунтів. Мобільний телефон є персональною інформацією і користувачі мають можливість видаляти інформацію про номер телефона після підтвердження реєстрації з профайла, що і роблять спамери (і не тільки). Для того, щоб після видалення мобільного телефона спамер не міг створити акаунт, прив'язаний до того ж телефонного номеру, в системі зберігається також хеш телефонного номера. Але для того, щоб відновити пароль до входу, який забув користувач поштовика "Роги та Копита", потрібно ввести номер телефону, за яким було створено акаунт, і на який прийде код підтвердження скидання паролю. Телефонний номер хешується перед відправленням одним з 3 алгоритмів. Тож повинна бути реалізована функція дехешування телефонного номера для відправки смс. Адже "Роги та Копита" не хочуть втрачати користувачів.

1. Завдання:

Написати систему, яка дозволяє хешувати дехешувати (тобто знайти номер телефону за заданим хешем) номери мобільних телефонів абонентів.

$R = \text{SHA1}(\text{phone}, \text{salt})$

Вхідні дані:

- номер телефона абонента для отримання хеш-значення. Формат номера 380XXXXXXXXX;
- хеш для отримання номеру телефону. Формат хеш-значення залежить від алгоритма хешування;
- підтримка basic authentication.

Вимоги:

- RESTful API;
- Тип алгоритму хешування SHA-1, SHA-2, SHA-3. Тип алгоритма має бути заданим в конфізі
- Сіль для хешування має конфігуруватися
- Кількість номерів абонентів в БД не менше 50 000 000
- Час першого запуску додатку не більше 4 хвилин
- Час виконання одинарного запиту не повинно перевищувати 300 ms
- Обробка хеш-колізій

Очікуваний результат:

Серверний API з методами

- Отримати хеш-значення номера абонента
- Отримати номери абонентів по хеш-значенню

2. Формат представлення результатів

1. Рішення оформлюється за вказаними критеріями

- 1.1. Рішення зберігається в окремій папці
- 1.2. Назва папки це назва номінації
- 1.3. Усі папки рішень за Номінаціями архівуються в єдиний архів з назвою у форматі **Ім'я_Прізвище.zip**
- 1.4. Архів завантажується на сайті в особистому кабінеті

2. Обов'язково перевіряйте запуск системи на іншому ПК перед надсиланням.

Проект має запускатися однією командою «docker-compose up» (Docker & Docker Compose).

Після запуску API повинен бути запущений за адресою - <http://localhost:8080/>



DEV CHALLENGE BACKEND

3. У папці також має міститись README файл, в якому обов'язково вкажіть:

- 3.1. Кроки для старту сервісу
- 3.2. Кроки для запуску тестів
- 3.3. Опис API
- 3.4. Речі, на які ви б хотіли звернути увагу або наступні кроки для вдосконалення вашого сервісу

4. Scaleme file

Зверніть увагу, що назва архіву - єдине місце, де ви вказуєте свої персональні дані.

Назви файлів всередині архіву не мають містити вашого імені чи прізвища.

Розмір архіву з рішенням не має перевищувати 10 MB.

5. Організатори та судді залишають за собою право дискваліфікувати роботу учасника, якщо робота:

- 5.1. Містить будь-яку вказівку на ім'я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника
- 5.2. Виконана у іншому форматі, ніж вказано у завданні
- 5.3. Виконана за допомогою сторонніх осіб, а не учасником особисто

3. Обмеження та критерії оцінювання

Категорія	Критерії	Бали
STANDARD	Роботоздатність додатку	100
STANDARD	Відповідність кодстайлу, структурованість, простота коду	30
STANDARD	Вибір архітектури і БД	50
BONUS	UNIT тести	15
BONUS	Наявність SIT тестів	14
BONUS	Наявність документації API DOC, raml	10
BONUS	Розподілена БД	5
BONUS	Наявність логування запитів	10
BONUS	Наявність API для отримання метрик системи	10
PRO	Використання asynchronous and non-blocking API	35
PRO	Час виконання 99% запитів не повинно перевищувати 10 ms при 100 одночасних з'єднаннях*	25
PRO	Підтримка https	16

Відкриті питання - тестовий ізольований стенд (CPU 8 CORE, RAM 16GB, HDD 100 GB) перевіряється JMeter на тестовому стенді.