

# SOFTWARE ENGINEERING SCHOOL 3.0

## Кейсове завдання



### Правила

Дедлайн виконання кейсового завдання: **28 травня о 20:59.**

### Як здати роботу?

- Виконане завдання необхідно **завантажити на GitHub** (публічний репозиторій).
- Сабмітнути виконане завдання в [цю гугл-форму](#).
- Перевірити чи твій профіль на GitHub відкритий для перегляду.

В GitHub, окрім проєкту, ти можеш додавати коментарі чи опис логіки виконання роботи в README.md документі. Правильна логіка може стати перевагою при оцінюванні, якщо ти не повністю виконаєш завдання.

Мови виконання завдання: **PHP або Go.**

Використання зазначених мов говорить про готовність працювати з подальшими практичними завданнями на школі.

Можна користуватися всією доступною інформацією, але просимо виконувати завдання самостійно.

## Опис завдання

Потрібно реалізувати сервіс з API, який дозволить:

- дізнатись поточний курс біткоїну (BTC) у гривні (UAH);
- підписати емейл на отримання інформації по зміні курсу;
- запит, який відправить всім підписаним користувачам актуальний курс.

## Додаткові вимоги:

1. Сервіс має відповідати описаному API. Сам API описаний [тут](#) у вигляді swagger документації. Для зручного перегляду можна скористатися сервісом <https://editor.swagger.io/>.
2. Всі дані, для роботи додатку повинні зберігатися в файлах (підключати базу даних не потрібно). Тобто, потрібно реалізувати збереження та роботу з даними (наприклад, електронними адресами) через файлову систему.
3. В репозиторії повинен бути Dockerfile, який дозволяє запустити систему в Docker. З матеріалом по Docker вам необхідно ознайомитись самостійно.
4. Документацію потрібно дотримуватись повною мірою, тому змінювати контракти не можна.
5. Можна використовувати релевантні фреймворки.
6. Також ти можеш додавати коментарі чи опис логіки виконання роботи в README.md документі. Правильна логіка може стати перевагою при оцінюванні, якщо ти не повністю виконаєш завдання.

Ти можеш користуватися всією доступною інформацією, але виконуй завдання самостійно.

Успіхів!

Бонус! Матеріали для ознайомлення з Docker`ом: [Dockerizing a Go App — and Optimising Its Image Size](#)