



# DEV CHALLENGE 1 2

## Backend Task

Level: **Standard** (частина A) and **Pro** (частина B)

Duration ≈ **4+ hours**

Deadline: **03 June 23:59 EEST**

Коли ваш браузер генерує сторінку, зазвичай він звертається декілька разів до різних CSS файлів. Крім того, для рендерингу цільової сторінки зазвичай йому треба всього лиш 20% з усього CSS коду.

Уявіть собі, як було б зручно, якби сервер віддавав одразу мінімально необхідну кількість стилів для рендерингу в веб браузері. Ви б змогли вставити ці стилі інлайн в тіло HTML документа, тим самим скоротити час очікування відображення сторінки.

Це і є вашим викликом :)

### Завдання:

Написати сервіс, який буде отримувати на вхід список URL адрес, запускати Google Chrome \ Chromium в режимі headless, отримувати з браузера стилі CSS і віддавати у відповіді у форматі JSON об'єкт, де ключі об'єкта являють собою переданий URL, а значення - рядки з мініфікованими CSS відповідно.

### Для Standard [Частина A]:

- Сервіс отримує на вхід список URL адрес
- Обробка URL може проводитися послідовно
- У відповідь у форматі JSON повертається мінімальний CSS для переданих URL

Приблизний формат відповіді:

```
{  
  url1: mincss1,  
  url2: mincss2,  
  ...  
}
```

- Якщо по URL відбувається redirect, то minimal css не повертаєм.

### Формат оформлення завдання Standard:

1. У завданні вказаний мінімально необхідний функціонал, який ви маєте виконати повністю.
2. Наявність архіву з вихідним кодом та інструкцією інсталяції та використанню сервісу у файлі README.

### Для Pro [Частина B]:

- Додати паралельну обробку декількох URL
- Додати до частини A кешування згенерованого мінімального CSS для кожного URL на N секунд
- Ввести перевірку на наявність даних в кеші та віддавати данні звідти
- Додати метод для примусового скидання кешованих даних для конкретного URL
- Додати метод для примусового скидання всіх кешованих даних

**Формат оформлення завдання Pro:**

1. У завданні вказаний мінімально необхідний функціонал, який ви маєте виконати повністю.
2. Наявність архіву з вихідним кодом та інструкцією інсталяції та використанню сервісу у файлі README.

**Критерії оцінювання:**

Критерій для Standard та Pro	Максимальний бал
Якість коду	60
Швидкість обробки запитів з результатами тестування	60
Обробка помилок	50
Стійкість до Memory Leak	40
Наявність тестів	30
Документація	16
$\Sigma$	<b>256 балів</b>

**32 бонусних бали будуть віддані найефективнішим рішенням на думку суддів.**

[Дізнатись інформацію про суддів обраного напрямку.](#)

**Зверніть увагу**

1. Рішення необхідно вивантажити в Особистому кабінеті на сайті <https://devchallenge.it> за обраним напрямком до 03 червня, 23:59 (EEST). Після вичерпання часу можливість вивантажити роботи на сайт буде автоматично заблокована.
2. Питання та уточнення щодо змісту завдання ви можете задати, заповнивши [форму](#). Відповідь на своє запитання шукайте в [документі](#). Перш ніж надсилати запитання, перевірте [документ](#). Можливо, на ваше запитання вже є відповідь :)
3. Судді ігноруватимуть питання, які не стосуються завдання Чемпіонату.
4. Організаційні запитання надсилайте на пошту **team@devchallenge.it**
5. Оголошення результатів відбудеться 15 червня.

## DEV Challenge 12 partners

**facebook**

Strategic partner

CROSS | OVER

General partner

**EMPO**

Strategic partner