



## Завдання Онлайн Раунду Backend | DEV Challenge XVIII

### Зміст:

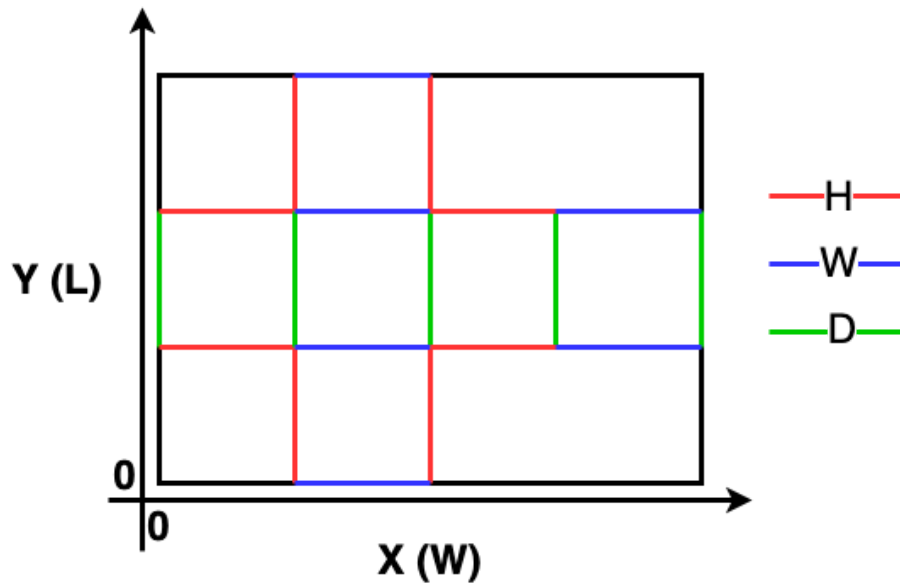
1. [Опис завдання](#)
2. [Формат презентації результатів](#)
3. [Дедлайн подачі рішень](#)
4. [Критерії оцінювання](#)
5. [Контакти](#)

### 1. Опис завдання

Ваш замовник виготовляє картонні коробки з листів картону, щоб запаковувати туди різні речі. Потрібно зробити WebAPI сервіс, що допоможе йому вирізати розкрій з цілого листа — він буде там задавати розмір листа та розмір бажаної коробки, а ваш сервіс видасть йому програму для ЧПУ, що виріже йому максимальну кількість коробок з мінімальною кількістю відходів.

#### Одиниці виміру та координати

- **W** ширина листа у міліметрах, **L** довжина.
- **W** ширина готової коробки, **D** глибина коробки, **H** висота коробки.
- Розкрій проектує ваш сервіс сам. Це коробка у розгорненому вигляді (для спрощення у формі хреста).



ЧПУ має 1 різак, підставку для листів картону та наступну систему команд

- **START** — вставляє новий чистий лист з картону. Різак при цьому встановлюється у координати (0,0) — лівий нижній кут та займає холосте положення.
- **DOWN** — опускає різак з холостого положення у робоче. Якщо він вже опущений — не робить нічого.
- **UP** — підіймає різак з робочого положення в холосте, а якщо він був вже піднятий — не робить нічого.
- **GOTO X, Y** — переходить у точку X,Y. Якщо різак у робочому положенні — утворює розріз (X0, Y0) - (X,Y) де (X0, Y0) — координати попереднього положення різак.
- **STOP** — сигналізує про нормальне завершення програми, піднімає різак і повертає його в 0,0.

Вам потрібні розробити всього лишень один ендпойнт котрий приймає POST запит у форматі JJSON:

```
// POST /api/simple_box
{
  "sheetSize": {
    "w": 800, // ширина листа в мм
    "l": 600 // довжина листа в мм
  },
  "boxSize": {
    "w": 200, // ширина коробки в мм
    "d": 200, // глибина коробки в мм
  }
}
```

```
"h": 200    // висота коробки в мм
}
}
```

Та повертає відповідь 200 у форматі JSON:

```
{
  "success": true, // успіх?
  "amount": 1,     // к-сть коробок що буде вирізано
  "program": [     // послідовний набір команд для різачка
    {
      "command": "START"
    },
    {
      "command": "GOTO", // команда
      "x": 200,          // рух до точки 200мм по осі X
      "y": 0             // рух до точки 0мм по осі Y
    },
    {
      "command": "DOWN"
    },
    {
      "command": "GOTO",
      "x": 200,
      "y": 200
    },
    {
      "command": "GOTO",
      "x": 0,
      "y": 200
    },
    {
      "command": "UP",
    },
    {
      "command": "GOTO",

```

```
    "x": 0,
    "y": 400
  },
  {
    "command": "DOWN",
  },
  {
    "command": "GOTO",
    "x": 200,
    "y": 400
  },
  {
    "command": "GOTO",
    "x": 400,
    "y": 400
  },
  // ... усі інші команди
  {
    "command": "STOP"
  }
]
```

У випадку помилки відповідь сервісу повинна бути 422 у форматі JSON:

```
// Якщо вхідні дані не валідні (не той формат, негативні
значення, тощо)
{
  "success": false,
  "error": "Invalid input format. Please use only positive
integers"
}

//Якщо заданий лист не зможе вирізати хоча б один розмір
коробки
```

```
{  
  "success": false,  
  "error": "Invalid sheet size. Too small for producing at  
least one box"  
}
```

Будьте уважні до помилок, бо це повідомлення буде виведено на екран для оператора і йому повинно бути зрозуміло, що ж пішло не так.

Очікується, що після запиту, час очікування відповіді від сервера не перевищує 10-ти секунд.

## 2. Формат презентації результатів

1. Рішення вивантажувати в кабінеті Учасника на [сайті](#) в ОДНОМУ файлі-архіві з назвою у форматі **Ім'я\_Прізвище.zip**.
2. Рішення має бути надано у вигляді серверної частини. Підніматись в контексті віртуального оточення Docker, але використання Vagrant також допускається. Для старту вашого додатка має бути необхідним запустити єдину команду: **docker-compose up**, або **vagrant up**.
3. Після цього за адресою [http://127.0.0.1:8080/api/simple\\_box](http://127.0.0.1:8080/api/simple_box) повинен бути доступний ендпойнт рішення.
4. README файл, в якому обов'язково вказуйте:
  - Кроки для старту сервісу;
  - Кроки для запуску тестів;
  - Методологію, яку ви обрали, та пояснення, чому на ваш погляд вона найкраще підходить;
  - Речі, на які ви б хотіли звернути увагу або наступні кроки для вдосконалення вашого сервісу.

👉 Зверніть увагу, що назва архіву — єдине місце, де ви вказуєте свої персональні дані. Назви файлів всередині архіву не мають містити вашого ім'я чи прізвища. Розмір архіву з рішенням не має перевищувати 10 MB.

Організатори та Судді залишають за собою право дискваліфікувати роботу Учасника, якщо робота:

- Містить будь-яку вказівку на ім'я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника;
- Виконана у форматі, відмінному від того, що вказаний у завданні;
- Виконана за допомогою сторонніх осіб, а не Учасником особисто.

### 3. Дедлайн подачі рішень

17 листопада 2021 року включно. Після вичерпання часу можливість вивантажити роботи на сайт буде автоматично заблокована.

Оголошення переможців відбудеться 29 листопада 2021 року.

### 4. Критерії оцінювання

Категорія	Критерії	Бали (%)
Lite	Працездатність <code>POST /api/simple_box</code>	14
Lite	Дотримання кодстайлу, структурованість та простота коду	8
Lite	Вибір та реалізація алгоритму	16
Bonus	Наявність та повнота тестів	14
Hard	<code>POST /api/simple_box</code> для декількох коробок на лист	14
Hard	<code>POST /api/simple_box</code> декількох коробок на лист та мінімальної кількості відходів	28
Bonus	Поведінка при підвищеному навантаженні та складності запитів	6

### 5. Контакти

Питання та уточнення щодо змісту завдань:

Канал Slack: [nomination-backend](#).

Судді ігноруватимуть питання, які не стосуються завдань Чемпіонату.

Організаційні питання: [hello@wechallenge.it](mailto:hello@wechallenge.it) чи канал Slack [#02-ask-the-organisers](#).