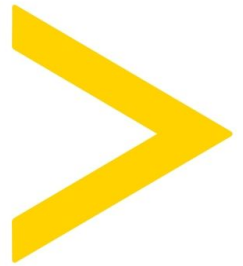


Final task  
DEV Challenge 11  
Fullstack developer



## 1. Task

People love to travel. But with so many airlines and flights these days, it can be confusing to figure out how to get from city A to city B, efficiently. Moreover, everyone has its own definition of what efficiency means:

- A. **time**: optimize for the shortest duration of the trip (including layovers).
- B. **price**: select the itinerary which has the lowest total price.
- C. **connections**: optimize for the shortest number of flights (minimum transfers).

Your goal is to help these customers figure out their travel plans. You will deliver an end-to-end solution, which includes a backend component (made available through an API) and a website that serves customer needs; the website will interact with the backend component through its API.

### **Backend**

The backend component will be a REST service (which receives requests through the HTTP protocol and sends replies using the JSON-based format). The following end-points need to be defined:

#### **/flights**

- this can be used to add, delete or update flights in the system. An important restriction consists in the fact that all the flights re-occur on a daily basis. Therefore, in order to define a flight, you only need to specify its ID, its start time, its duration (in minutes), the origin and destination cities, and the cost that people must pay for it (in USD):

#### **Request**

```
PUT /flights?id=1&start=10:30&duration=60&origin=Kyiv&destination=Bucharest&cost=200
```

#### **Response**

*http response status code: 200 OK*

```
{}
```

If you want to update, at a later stage, the cost of the flight, you would make another call:

#### **Request**

```
POST /flights?id=1&cost=100
```

#### **Response**

*http response status code: 200 OK*

```
{}
```

You can also query for the flights matching a specific condition:

### Request

GET /flights?id=1

### Response

*http response status code: 200 OK*

```
[
  {
    "id": "1",
    "start": "10:30",
    "duration": "60",
    "origin": "Kyiv",
    "destination": "Bucharest",
    "cost": "100"
  }
]
```

or delete a flight from the system:

### Request

DELETE /flights?id=1

### Response

*http response status code: 200 OK*

```
{}
```

### /trips

This will be the other end-point, which will be used in order to get the optimal itinerary for the specified origin and destination cities, using the specified efficiency criterion: time, price or connections.

PUT /flights?id=1&start=10:30&duration=60&origin=Kyiv&destination=Bucharest&cost=200

PUT /flights?id=2&start=12:30&duration=60&origin=Bucharest&destination=Iasi&cost=200

GET /trips?origin=Kyiv&destination=Iasi&efficiency=time&results=1

Response:

```
[
  {
    "flights": ["1", "2"],
    "time": "180",
    "price": "400",
    "connections": 2
  }
]
```

In case multiple results are requested, they must be returned in sorted order based on the efficiency criteria specified.

P.S.: the maximum duration for a single direct flight is 1440 minutes. Customers need at least 30 minutes to transfer to another flight in the same airport.

## **Frontend**

The primary audience for the frontend consists in potential customers that want to find information about their best vacation itinerary. You should create an UI that allows them to input an origin and destination city, as well as their preferred efficiency criterion. The UI should display back the results found for their specified query, in sorted order.

A secondary audience for the frontend is made by travel agency operators, that should have authorization to add, update or delete flights from the system.

## **2. Format of presentation of results**

Unload the decision in a personal account on the site in ONE file archive with the name in the format Name\_Sign.zip, which should contain the following:

- 1.1. a README.txt file containing on its first line a single command required to launch the webserver. The README.txt should also contain the root URL of the backend API and the root URL of the frontend's homepage;
- 1.2. the source code of the solution;
- 1.3. a Makefile to build the binaries from source.

Please note that the name of the archive is the only place where you specify your personal data. Names of files inside the archive should not include your name or last name. Before downloading the archive, be sure to open it and check that the folder names don't add your solution automatically.

The organizers and judges reserve the right to disqualify the work of the participant if the work:

- 2.1 contains any indication of the name, surname, e-mail, company, address or other personal data of the participant;
- 2.2 executed in a different format than specified in the task;
- 2.3 performed by third parties, and not by a person in person.

## **3. Limitations and evaluation criteria**

1. Judges will pay attention to:
  - a. Delivering the requirements of the task.
  - b. Code quality (modularity, maintainability, comments & easiness of understanding).
  - c. Performance.
  - d. Interface usability (examples: auto-complete or drop-downs for city selectors)..
2. Points will be scored according to the following criteria:
  - a. 15% - backend API for the flights/ end-point.
  - b. 15% - backend API for the trips/ end-point (simple cases).
  - c. 20% - providing correct itineraries for a wide range of flights scenarios.
  - d. 10% - frontend for customers.
  - e. 10% - frontend for travel agency operators.
  - f. 15% - Speed in navigating through the frontend.
  - g. 15% - Usability of the frontend interface (intuitive, clear UI).
3. Additional points can be obtained for:
  - a. 10% bonus - responsive interfaces (that adapt to mobile/tablet screen sizes).
  - b. 10% bonus - unit-tests.

## 4. Experts



### **Vlad Dascalu**

Software development Manager  
@Amazon Romania  
Vlad Dascalu currently works as a Software Development Manager at Amazon Romania, since February 2016. Previously, he worked as a Software Engineer at Google (2007-2010) and he won a gold medal at the International Olympiad in Informatics (Finland, 2001).



### **Alexander Trough**

Erlang/Elixir Developer EdenLab  
Alexander has been a judge of DEV Challenge for many consecutive seasons. Thanks to his important comments teams either stay together forever or fall apart immediately. His skills as a Fullstack developer were extremely useful for DOU.UA and Adtena.com; moreover, he is a founder of the latter.



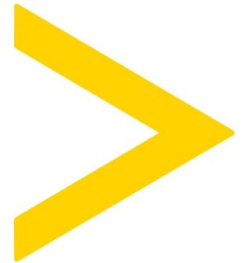
### **Bohdan Vashchuk**

Solutions Architect, Team Lead  
@Teamvoy  
He started his career as a front end developer and system administrator, then moved to the full stack. Bohdan has worked as a freelancer and in outsourcing before starting to develop his own business. He participates in a community life, delivering speeches at meetups with @kottans; he is a part of the core team in pivorak meetup. Currently Bohdan is mainly working on fintech & ecommerce projects.

## 5. Contacts

1. The decision must be unloaded before 18:00, 1st of July **to Google Drive via personal link that you have received in email**. Judges will not check solutions uploaded via official DEV Challenge site. After the time expires, the opportunity to unload work on the site will be automatically blocked.
2. Questions and clarifications on the content of the task you can ask in your nomination channel in Slack - [Final DEV 11](#).
3. Judges will ignore questions that are not relevant to the Championship
4. Organizational questions please send to #general in Slack - [Final DEV 11](#).
5. **Announcement of the winners will take place at 16:00, 02/07.**

Завдання  
Етапу Фіналу  
DEV Challenge 11  
Номінація Fullstack developer



## 1. Завдання

Люди люблять подорожувати. Але з такою великою кількістю авіаліній і польотів як зараз, може бути складно зрозуміти, як дістатись з міста А в місто В ефективно. Окрім того, кожен має власні критерії ефективності:

- A. **час**: оптимізувати для найшвидшої подорожі (включаючи очікування вильотів).
- B. **ціна**: оберіть маршрут з найнижчою загальною ціною
- C. **з'єднання**: оптимізувати для найменшої кількості пересадок.

Ваша ціль - допомогти цим клієнтам створити їх плани подорожі. Ви повинні надати end-to-end рішення, яке включає backend компонент (доступний через API) та вебсайт що обслуговує потреби клієнтів; вебсайт буде взаємодіяти з backend компонентом через його API.

### **Backend**

Backend компонент буде являти собою REST сервіс (який отримує запити через HTTP протокол і надсилає відповідь в форматі JSON). Повинні бути визначені наступні точки доступу:

### **/flights**

- може бути використана для додавання, видалення чи оновлення польотів в системі. Важливе обмеження - всі польоти повторюються щоденно. Отже, для того, щоб визначити політ, ви повинні лише зазначити його ID, час вильоту, час польоту (в хвиликах), місто відправлення та призначення, а також ціна, яку має заплатити клієнт (в USD):

### **Запит**

```
PUT /flights?id=1&start=10:30&duration=60&origin=Kyiv&destination=Bucharest&cost=200
```

### **Відповідь**

```
http response status code: 200 OK
```

```
{}
```

Якщо ви хочете оновлювати, в наступній фазі, ціну польоту, ви робите наступний виклик:

### **Запит**

```
POST /flights?id=1&cost=100
```

### **Відповідь**

```
http response status code: 200 OK
```

```
}
```

Ви можете також запитувати польоти, що відповідають певним умовам:

### Запит

```
GET /flights?id=1
```

### Відповідь

*http response status code: 200 OK*

```
[  
  { "id": "1",  
    "start": "10:30",  
    "duration": "60",  
    "origin": "Kyiv",  
    "destination": "Bucharest",  
    "cost": "100"  
  }  
]
```

Або видалити політ з системи:

### Запит

```
DELETE /flights?id=1
```

### Відповідь

*http response status code: 200 OK*

```
}
```

### /trips

Це буде інша точка, яка буде використана для того щоб отримати оптимальний маршрут для певних міст відправки і призначення, використовуючи певні критерії ефективності: ціна, час чи зв'язки.

```
PUT /flights?id=1&start=10:30&duration=60&origin=Kyiv&destination=Bucharest&cost=200
```

```
PUT /flights?id=2&start=12:30&duration=60&origin=Bucharest&destination=Iasi&cost=200
```

```
GET /trips?origin=Kyiv&destination=Iasi&efficiency=time&results=1
```

Response:

```
[  
  {  
    "flights": ["1", "2"],  
    "time": "180",  
    "price": "400",  
    "connections": 2  
  }  
]
```

В випадку коли запитано більш як один результат, вони повинні бути повернені відсортованими по зазначеному критерію ефективності.

P.S.: максимальна тривалість для польоту в один кінець - 1440 хвилин. Клієнти потребують як мінімум 30 хвилин для пересадки на інший політ в тому ж аеропорті.

### **Frontend**

Більшість користувачів фронтенду - це потенційні клієнти, що хочуть знайти інформацію про найкращий маршрут для їх відпустки. Ви повинні створити UI, що дозволяє їм ввести місто відправлення та призначення, а також їх критерій ефективності. UI повинен показувати результати для запиту відсортованими.

На другій місці - користувачі фронтенду, оператори з туристичних агенцій, які повинні мати можливість додавати, оновлювати та видаляти польоти з системи.

## **2. Формат представлення результатів**

Завантажте рішення в персональний кабінет на сайті в ОДНОМУ архіві з назвою в форматі Name\_Sign.zip, яке має містити наступне:

- 1.1. README.txt файл який в першому рядку містить команду для запуску веб сервера. README.txt також повинен містити головний URL бекенду та головний URL домашньої сторінки фронтенду;
- 1.2. вихідний код рішення;
- 1.3. Makefile для того щоб зібрати бінарні файли з вихідного коду.

Зверніть увагу, що назва архіву - єдине місце, де ви вказуєте свої персональні дані. Назви та зміст файлів всередині архіву не мають містити вашого ім'я чи прізвища. Перед завантаженням архіву обов'язково відкрийте його і перевірте, чи в назвах папок ваше рішення не додалося автоматично.

2. Організатори та судді залишають за собою право дискваліфікувати роботу учасника, якщо робота:
  - 2.1. містить будь-яку вказівку на ім'я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника;
  - 2.2. виконана у іншому форматі, ніж вказано у завданні;
  - 2.3. виконана за допомогою сторонніх осіб, а не учасником особисто.

## **3. Обмеження та критерії оцінювання**

1. Судді будуть звертати увагу на:
  - a. Виконання умов завдання.
  - b. Якість коду (модульність, підтримка, коментарії & простота розуміння).
  - c. Працездатність.
  - d. Дружність інтерфейсу (наприклад: автодоповнення чи випадючий список для вибору міст)..
2. Очки будуть надаватись за наступними критеріями:
  - a. 15% - backend API точки flights/.
  - b. 15% - backend API точки trips/ (прості випадки).
  - c. 20% - надання правильних маршрутів для широкого спектру сценаріїв польоту.
  - d. 10% - фронтенд для клієнтів.
  - e. 10% - фронтенд для операторів туристичних операторів.

- f. 15% - швидкість навігацій через фронтенд.
  - g. 15% - дружність фронтенд інтерфейсу (інтуїтивність, простота UI).
3. Додаткові очки будуть надані за:
- a. 10% бонус - адаптивний інтерфейс (такий, який адаптується під розміри екранів мобільних/планшетів).
  - b. 10% бонус - юніт тести.

#### 4. Експерти



**Vlad Dascalu**  
Software Development Manager  
@Amazon Romania  
Vlad Dascalu – менеджер з розробки програмного забезпечення в Amazon Romania з лютого 2016 року. Раніше він працював інженером з програмного забезпечення в Google (2007-2010), і отримав золоту медаль на міжнародній олімпіаді з інформатики (Фінляндія, 2001).



**Олександр Труш**  
Erlang/Elixir Developer @Urbox  
Олександр судить вже багато Чемпіонатів поспіль при чому абсолютно різні номінації: back-end, web team, fullstack. Його вмінням в якості Fullstack розробника завдячує DOU.UA та [Adtena.com](http://Adtena.com), засновником якого він був. Зараз очолює розробку у компанії Digital Hanger.



**Богдан Варшук**  
Solutions Architect, Team Lead @Teamvoy  
Починав кар'єру як front end розробник, згодом перейшов у full stack. Встиг попрацювати на фрілансі та в аутсорсі перед тим як почати розвивати свою компанію. Виступає на мітапах, викладає разом із @kottans, частина core team у pivotak meetup. Зараз працює в основному над fintech & ecommerce проектами.

#### 5. Контакти

1. Рішення необхідно вивантажити **до 18:00, 1 липня (EEST)** на Google Drive **за своїм особистим лінком, який ви отримали в листі від організаторів**. Після вичерпання часу ваш лінк стане неактивним. Після вивантаження рішення скачайте свій архів, відкрийте його і перевірте, чи не залишилося в ньому ваших персональних даних.
2. Питання та уточнення щодо змісту завдання ви можете задати в каналі своєї номінації в Slack - [Final DEV 11](#).
3. Судді ігноруватимуть питання, які не стосуються завдання Чемпіонату.



4. Організаційні запитання надсилайте у #general в Slack - [Final DEV 11](#).
5. Оголошення переможців відбудеться о 16:00, 2 липня (EEST).



General Partner



Strategic Partner



Strategic Partner

Development Center  
Romania