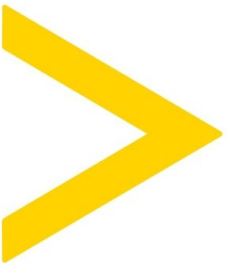
Final task

DEV Challenge 11

Back-end developer

Standard



1. Task

Call center software: There is a call center with employees that have certain expertise areas; customers are calling and indicate their desired area.

You have to write the server that directs the calls (binds customers to employees) maximizing the number of customer contacts that are fulfilled.

The server should handle 3 types of requests:

1. http://server.name/register?name=EmployeeName1&area=bills&area=contracts&area=special-offers

   This request defines EmployeeName1 as an employee with three areas of expertise: bills, contracts and special-offers. Only one employee can be defined in a request. The same name can be redefined multiple times, because employees are becoming available after handling a customer. The response must be code 200, with content:
   WELCOME

2. http://server.name/call?area=bills&area=leases&area=bills

   This indicates that a new batch of 3 customers are on the line. Two of them are interested in "bills" and one in "leases". Based on the list of registered employees, you have to assign (or decline) each call, in the order they appear in request. You have to respond with HTTP code 200 OK, and content is a JSON indicating the assigned employees. For example:

```
{
  "totalAssignments" : "1",
  "assignments" : [
        {
        "area" : "bills",
        "employee" : "EmployeeName1"
        },
        {
        "area" : "leases",
        "employee" : ""
        },
        {
        "employee" : "",
        "area" : "bills"
        }
```

```
  ]
}
```

The reply above indicates that only one assignment was possible: EmployeeName1 takes the first call about "bills". The other calls should be postponed because no employee is available. This is indicated by empty "employee" value. Once an assignment is set, the employees listed are busy taking the call, so they are no longer available. They may return later (via a register request).

3. http://server.name/reset

This is done usually at the end of a work day. It means that all registered employees leave home: the system can restart tomorrow.

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 files:

1.1. the source code of the solution
1.1. solution must be provided in Docker environment, especially using docker-compose. So all your containers must be linked between each other and all startup scripts or any other pre configuration procedures must be run automatically. You can check yourself before you'll send your solution by running 'docker-compose up', after docker-compose up will be finished your app should be available at the location defined in the README file. Remember judges should not additionally configure your solution to avoid misconfiguration and underestimate your solution.
1.2. a README file indicating the algorithm choice, instalation instructions and deploy instructions

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

Judges will pay attention to:

1. Application's overall operability;
2. Algorithm choice and its results on judges' test data;
3. Unit tests and documentation.

Points will be scored according to the following criteria:

1. Results (number of call assignments set by the server on judges' test data);
2. Ease of instalation (documentation, standard compliance for deployment);
3. Architecture choices: can it be extended for scaling.

Additional points can be obtained for:

1. Extensibility of the algorithm;
2. Documentation for operating the solution.

4. Experts

Oleksiy Milotsky
CTO/Co-Founder @ Mil's

The winner and repeated referee of the Championship, whose team Mil`s has grown to be a technical partner of We_Challenge. Engaged in the development of Front-end and Back-end architectures for high-volume Web solutions. Significant evidence of his experience is ZCE/ZFCE/ZFCA and ZF2 contributor certification. Web development over 10 years.

Sergiy Ilyukhin
.NET Backend Team Lead @Ciklum

More than 10 years of web application development experience (ASP.NET, MVC, WebApi, AngularJs, Azure services). Supports development of open data services, author and developer of monitoring service of public information of local self-government bodies. Co-founder of Freelance Brovary community, trainer at volunteer Java courses in Brovary

Corneliu Rudeanu
Senior developer @ Amazon Development Center Romania

Corneliu Rudeanu is a senior developer at Amazon Development Center Romania, identifying metrics that indicate Customer Experience. He finds competition a method of personal evolution and a motivation for performance. He's eager to learn new stuff and also likes to share his findings.

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 Back-end developer

Категорія Standard

1. Завдання

Програмне забезпечення для кол центру: Дано кол центр із співробітниками, які мають певні експертні області, клієнти телефонують і та обирають необхідну їм область.
Ви повинні написати сервер, що спрямовує дзвінки (з'єднує клієнтів та співробітників) максимізуючи кількість дзвінків, які були задоволені.
Сервер повинен обробляти 3 типи запитів:

1. http://server.name/register?name=EmployeeName1&area=bills&area=contracts&area=special-offers

   Цей запит визначає EmployeeName1 як співробітника з трьома експертними областями: рахунки, контракти та спеціальні пропозиції. Лише один співробітник може бути визначений в запиті. Одне ім'я може бути перевизначене необмежену кількість разів, адже співробітники стають доступними після обробки користувача. Відповідь має бути код 200, з тілом:
   WELCOME

2. http://server.name/call?area=bills&area=leases&area=bills

   Це повідомляє про 3х клієнтів в черзі. Двоє з них зацікавлені в "bills" і один зацікавлений в "leases". Опираючись на список зареєстрованих користувачів ви повинні призначити (або відхилити) кожен виклик в порядку в якому вони з'являються в запиті. Ви повинні відповідати HTTP code 200 OK, і тілом яке буде містити JSON, що буде показувати призначених співробітників. Наприклад:

```
{
  "totalAssignments" : "1",
  "assignments" : [
        {
        "area" : "bills",
        "employee" : "EmployeeName1"
        },
        {
        "area" : "leases",
        "employee" : ""
        },
        {
        "employee" : "",
```

```
            "area" : "bills"
            }
  ]
}
```

Відповідь вище показує, що можливе лише одне призначення: EmployeeName1 приймає перший виклик про "bills". Інші дзвінки повинні бути відкладені, адже немає доступних співробітників. Цей факт показаний пустим значенням "employee". Коли призначення виконано, співробітники у списку зайняті обробкою виклику, а отже недоступні. Вони можуть повернутись пізніше (зробивши запит реєстрації).

3. http://server.name/reset

Цей запит робиться, зазвичай, в кінці робочого дня. Це означає, що усі зареєстровані співробітники йдуть додому: система може перезавантажитись завтра.

2. Формат представлення результатів

1. Рішення вивантажувати в особистому кабінеті на сайті в ОДНОМУ файлі-архіві з назвою у форматі **Ім'я_Прізвище.zip**, у якому мають бути наступні файли:

   1.1. вихідний код рішення;
   1.2. рішення має бути надане в середовищі Docker, особливо, використовуючи docker-compose. Отже, всі ваші контейнери повинні бути пов'язані один з одним і всі скрипти запуску чи інші процедури преконфігурації повинні запускатись автоматично. Ви можете перевірити себе перед тим як відправляти рішення запустивши 'docker-compose up', після того, як docker-compose up відпрацює, ваш додаток повинен бути доступним в місці, зазначеному в файлі README. Пам'ятайте, судді не повинні додатково конфігурувати ваше рішення для того щоб уникнути невірної конфігурації і недооцінення вашої роботи;
   1.3. README файл, що описує обраний алгоритм, інструкції по встановленню та розгортанню.

   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

Судді будуть звертати увагу на:

1. загальну працездатність додатку;
2. обраний алгоритм та його результати на тестових даних, підготовлених суддями;
3. Unit тести та документація.

Бали нараховуватимуться за наступними критеріями:

1. результати (кількість призначених сервером викликів на тестових даних приготованих суддями);
2. простота встановлення (документація, відповідність процесу розгортання стандартам);
3. вибір архітектури: чи рішення розширюване.

Додаткові бали можна отримати за:

1. розширюваність алгоритму;
2. документація по використанню рішення.

4. Експерти

Олексій Милоцький
CTO/Co-Founder @Mil's

Переможець та неодноразовий суддя Чемпіонату, чия команда Mil`s виросла до технічного партнера компанії “We_Challenge”. Займається розробкою архітектури Front-end та Back-end частин для високонавантажених Web рішень. Вагомим доказом його досвіду є сертифікація ZCE/ZFCE/ZFCA та ZF2 contributor. У Web-розробці понад 10 років.

Сергій Ілюхін
.NET Backend Team Lead @Ciklum

Більше 10 років досвіду розробки веб-додатків (ASP.NET, MVC, WebApi, AngularJs, сервіси Azure). Підтримує розвиток сервісів відкритих даних, автор та розробник сервісу моніторингу публічної інформації органів місцевого самоврядування. Співзасновник спільноти Freelance Brovary, тренер на волонтерських Java курсах в Броварах.

Corneliu Rudeanu
Senior developer @ Amazon Development Center Romania

Corneliu Rudeanu – старший розробник в Amazon Development Center Romania, займається визначенням показників, які вказують на якість обслуговування клієнтів. На його думку, змагання – чудовий стимул до особистого розвитку і мотивація до відмінного виконання завдання. Corneliu подобається вчитися новому і ділитися своїми знахідками.

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