AI2 LAB F	Szabat Krystian Album 51114 Grupa 4	**Wersja 1**

SERWISY I KOMENDY

SPIS TREŚCI

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- zamykanie reużywalnej logiki biznesowej w serwisach;
- wykorzystanie serwisów w kontrolerach;
- wykorzystanie serwisów w komendach;
- tworzenie komend konsolowych.

ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie zasad tworzenia serwisów, komend.

Wejściówka?

UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do `Plik` -> `Informacje` -> `Właściwości` -> `Właściwości zaawansowane` -> `Niestandardowe` i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub `Ctrl+A` -> `F9`.

SERWIS WEATHERUTIL

Utwórz nowa klasę `src/Service/WeatherUtil.php`, a w niej deklaracje dwóch metod:

- `getWeatherForLocation($location)` – odpowiedzialna za pobranie pomiarów (prognoza pogody) na podstawie encji lokalizacji;
- `getWeatherForCountryAndCity($countryCode, $cityName)` – odpowiedzialna za pobranie pomiarów na podstawie kodu kraju i nazwy miasta. Wewnętrzna implementacja sprowadza się do pobrania lokalizacji na podstawie kodu kraju i nazwy miasta, a następnie wywołania metody `getWeatherForLocation` dla otrzymanej lokalizacji.

```php
<?php
declare(strict_types=1);

namespace App\Service;

use App\Entity\Location;
use App\Entity\Measurement;

class WeatherUtil
{
    /**
     * @return Measurement[]
     */
    public function getWeatherForLocation(Location $location): array
    {
        return [];
    }

    /**
     * @return Measurement[]
     */
    public function getWeatherForCountryAndCity(string $countryCode, string $city): array
    {
        return [];
    }
}
```

Zmodyfikuj WeatherController, żeby wykorzystywał nowy serwis:

```php
class WeatherController extends AbstractCon
{
    #[Route('/weather/{country}/{city}', na
    public function city(
        #[MapEntity(mapping: ['country' =>
        Location $location,
        MeasurementRepository $repository,
    ): Response
    {
        $measurements = $repository->findBy

        return $this->render('weather/city.
            'location' => $location,
            'measurements' => $measurements
        ]);
    }
}
```

```php
class WeatherController extends AbstractController
{
    #[Route('/weather/{country}/{city}', name: 'app_weather', requ
    public function city(
        #[MapEntity(mapping: ['country' => 'country', 'city' => 'c
        Location $location,
        WeatherUtil $util,
    ): Response
    {
        $measurements = $util->getWeatherForLocation($location);

        return $this->render('weather/city.html.twig', [
            'location' => $location,
            'measurements' => $measurements,
        ]);
    }
}
```

Sprawdź, czy strona prognozy pogody wciąż działa. Nie powinno być żadnych błędów, jednakże zwracana lista pomiarów będzie pusta.

Zaimplementuj ciało metody `getWeatherForLocation()`. Wstaw zrzut ekranu kodu:

```php
public function __construct(
    private MeasurementEntryRepository $measurementEntryRepository,
    private LocationRepository $locationRepository,
) {}

/**
 * @return Measurement[]
 */
public function getWeatherForLocation(Location $location): array
{
    return $this->measurementEntryRepository->findByLocation($location);
}
```

Punkty:	0	1

Zaimplementuj ciało metody `getWeatherForCountryAndCity()`. Wstaw zrzut ekranu kodu:

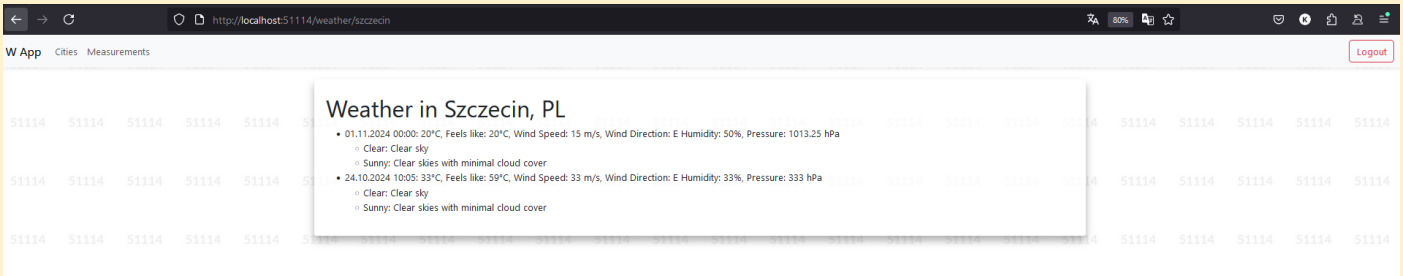
```php
/**
 * @return Measurement[]
 */
public function getWeatherForCountryAndCity(string $countryCode, string $city): array
{
    $city = str_replace('-', ' ', $city);
    $city = ucfirst($city);

    $location = null;
    if ($countryCode) {
        $countryCode = strtoupper($countryCode);

        $location = $this->locationRepository->findOneByCityAndCountry($city, $countryCode);
    }else {
        $location = $this->locationRepository->findOneByCity($city);
    }

    return $this->getWeatherForLocation($location);
}
```

Punkty:	0	1

Wstaw zrzut ekranu kodu kontrolera wykorzystującego metodę z serwisu:

```php
class WeatherController extends AbstractController
{

    #[Route('/weather/{city}/{country?}', name: 'app_weather', requirements: ['city' => '[a-zA-Z-]+', 'country' => '[a-zA-Z]{2}'])]
    public function city(
        // #[MapEntity(mapping: ['city' => 'city', 'country' => 'country'])]
        // Location $location,
        string $city,
        ?string $country,
        WeatherUtil $weatherUtil,
        LocationRepository $locationRepository,
        ): Response
    {
        $city = str_replace('-', ' ', $city);
        $city = ucfirst($city);

        $location = null;
        if ($country) {
            $country = strtoupper($country);

            $location = $locationRepository->findOneByCityAndCountry($city, $country);
        }else {
            $location = $locationRepository->findOneByCity($city);
        }

        $measurementEntries = $weatherUtil->getWeatherForLocation($location);

        return $this->render('weather/city.html.twig', [
            'location' => $location,
            'measurement_entries' => $measurementEntries,
        ]);      You, 7 days ago • feat: Add WeatherController

    }
}
```

Wstaw zrzut ekranu prognozy pogody z pomiarami pobranymi z serwisu:



Punkty:	0	1

KOMENDY

Wykorzystaj komendę `make:command` do utworzenia komendy `weather:location`, służącej do pobierania prognozy pogody dla lokalizacji:

```
php .\bin\console make:command

Choose a command name (e.g. app:agreeable-pizza):
> weather:location

created: src/Command/WeatherLocationCommand.php


 Success!
```

```
Next: open your new command class and customize it!
Find the documentation at https://symfony.com/doc/current/console.html
```

Edytuj utworzony plik `src/Command/WeatherLocationCommand.php`:

- podłącz serwis do konstruktora;
- ustaw wymagany argument id lokalizacji;
- w execute() wykorzystaj serwis do pobrania prognozy pogody;
- wydrukuj prognozę pogody na widok.

Przykładowe wywołanie:

```
php .\bin\console weather:location 1
Location: Szczecin
        2023-09-26: 18
        2023-09-27: 17
```

Przykładowy kod:

```
protected function execute(InputInterface $input, OutputInterface $output): int
    {
        $io = new SymfonyStyle($input, $output);
        $locationId = $input->getArgument('id');
        $location = $this->locationRepository->find($locationId);

        $measurements = $this->weatherUtil->getWeatherForLocation($location);
        $io->writeln(sprintf('Location: %s', $location->getCity()));
        foreach ($measurements as $measurement) {
            $io->writeln(sprintf("\t%s: %s",
                $measurement->getDate()->format('Y-m-d'),
                $measurement->getCelsius()
            ));
        }

        return Command::SUCCESS;
    }
```

Wstaw zrzuty ekranu kodu całości komendy:

```php
#[AsCommand(
    name: 'weather:location',
    description: 'Add a short description for your command',
)]
class WeatherLocationCommand extends Command
{
    public function __construct(
        private WeatherUtil $weatherUtil,
        private LocationRepository $locationRepository,
    ) {
        parent::__construct();
    }

    protected function configure(): void
    {
        $this
            ->setDescription("Get the weather for a location")
            ->setHelp("This command allows you to get the weather for a location")
            ->addArgument('location_id', InputArgument::REQUIRED, "Location id")
        ;
    }

    protected function execute(InputInterface $input, OutputInterface $output): int
    {
        $io = new SymfonyStyle($input, $output);
        $location_id = $input->getArgument('location_id');

        $location = $this->locationRepository->find($location_id);

        if (!$location) {
            $io->error("Location not found");
            return Command::FAILURE;
        }

        $measurementEntries = $this->weatherUtil->getWeatherForLocation($location);

        if (empty($measurementEntries)) {
            $io->error("No weather data found for location");
            return Command::FAILURE;
        }

        $io->success("Weather data for location: " . $location->getCity() . ", " . $location->getCountry());
        $io->table(
            ['Date', 'Temperature', 'Feels like' , 'Humidity', 'Pressure', 'Wind Speed', 'Wind Direction'],
            array_map(function ($measurementEntry) {
                return [$measurementEntry->getDateTime()->format('Y-m-d H:i:s'), $measurementEntry->getTemperatureCelcius(), $measurementEntry->getFeelsLike() ,
                $measurementEntry->getHumidity(), $measurementEntry->getPressure(), $measurementEntry->getWindSpeed(), $measurementEntry->getWindDirection()];
            }, $measurementEntries)
        );


        return Command::SUCCESS;
    }
}
    You, 19 minutes ago • feat: Add WeatherLocationCommand and WeatherUti…
```

Wstaw zrzuty ekranu wyniku działania komendy dla dwóch lokalizacji:

```
PS C:\studia\AI2-lab\pogodynka> php .\bin\console weather:location 1

[OK] Weather data for location: Szczecin, PL

 --------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  Date                  Temperature   Feels like   Humidity   Pressure   Wind Speed   Wind Direction
 --------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  2024-11-01 00:00:00   20            20           50         1013.25    15           E
  2024-10-24 10:05:00   33            59           33         333        33           E
 --------------------- ------------- ------------ ---------- ---------- ------------ ----------------
```

```
PS C:\studia\AI2-lab\pogodynka> php .\bin\console weather:location 2

[OK] Weather data for location: Stargard, PL

 ---------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  Date                   Temperature   Feels like   Humidity   Pressure   Wind Speed   Wind Direction
 ---------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  2024-11-01 00:00:00    23            21           50         1033.25    15           NE
 ---------------------- ------------- ------------ ---------- ---------- ------------ ----------------
```

Punkty:	0	1

W analogiczny sposób utwórz komendę do pobierania lokalizacji na podstawie kodu kraju i nazwy miejscowości.

Wstaw zrzuty ekranu kodu całości komendy:

```php
#[AsCommand(
    name: 'weather:CityCountry',
    description: 'Add a short description for your command',
)]
class WeatherCityCountryCommand extends Command
{
    public function __construct(
        private WeatherUtil $weatherUtil,
        private LocationRepository $locationRepository,
    ) {
        parent::__construct();
    }

    protected function configure(): void
    {
        $this
            ->setDescription('Get the weather for a city and country')
            ->setHelp('This command allows you to get the weather for a city and country')
            ->addArgument('city', InputArgument::REQUIRED, 'City')
            ->addArgument('country', InputArgument::OPTIONAL, 'Country')
        ;
    }

    protected function execute(InputInterface $input, OutputInterface $output): int
    {
        $io = new SymfonyStyle($input, $output);
        $city = $input->getArgument('city');
        $country = $input->getArgument('country');

        $city = str_replace('-', ' ', $city);
        $city = ucfirst($city);

        $location = null;
        if ($country) {
            $country = strtoupper($country);

            $location = $this->locationRepository->findOneByCityAndCountry($city, $country);
        } else {
            $location = $this->locationRepository->findOneByCity($city);
        }

        if (!$location) {
            $io->error('Location not found');
            return Command::FAILURE;
        }

        $measurementEntries = $this->weatherUtil->getWeatherForLocation($location);

        if (empty($measurementEntries)) {
            $io->error('No weather data found for location');
            return Command::FAILURE;
        }

        $io->success('Weather data for location: ' . $location->getCity() . ', ' . $location->getCountry());
        $io->table(
            ['Date', 'Temperature', 'Feels like', 'Humidity', 'Pressure', 'Wind Speed', 'Wind Direction'],
            array_map(function ($measurementEntry) {
                return [$measurementEntry->getDateTime()->format('Y-m-d H:i:s'), $measurementEntry->getTemperatureCelcius(), $measurementEntry->getFeelsLike(),
                    $measurementEntry->getHumidity(), $measurementEntry->getPressure(), $measurementEntry->getWindSpeed(), $measurementEntry->getWindDirection()];
            }, $measurementEntries)
        );
```

Wstaw zrzuty ekranu wyniku działania komendy dla dwóch miejscowości:

```
PS C:\studia\AI2-lab\pogodynka> php .\bin\console weather:CityCountry szczecin pl

[OK] Weather data for location: Szczecin, PL

 ---------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  Date                   Temperature   Feels like   Humidity   Pressure   Wind Speed   Wind Direction
 ---------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  2024-11-01 00:00:00    20            20           50         1013.25    15           E
  2024-10-24 10:05:00    33            59           33         333        33           E
 ---------------------- ------------- ------------ ---------- ---------- ------------ ----------------
```

```
PS C:\studia\AI2-lab\pogodynka> php .\bin\console weather:CityCountry police pl

[OK] Weather data for location: Police, PL

 --------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  Date                  Temperature   Feels like   Humidity   Pressure   Wind Speed   Wind Direction
 --------------------- ------------- ------------ ---------- ---------- ------------ ----------------
  2024-11-01 00:00:00   23            22           55         1003.25    15           W
 --------------------- ------------- ------------ ---------- ---------- ------------ ----------------
```

Punkty:	0	1

COMMIT PROJEKTU DO GIT

Zacommituj zmiany. Wyślij zmiany do repozytorium (`push`). Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie `lab-f` na podstawie głównej gałęzi kodu.

Podaj link do brancha `lab-f` w swoim repozytorium:

https://github.com/szvbvtk/ai2-pogodynka/tree/lab-f

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Podczas laboratorium zdobyłem umiejętność tworzenia komend konsolowych w Symfony oraz umieszczania wielokrotnie używanych fragmentów kodu w oddzielnych plikach, w tym przypadku Service/WeatherUtill.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.