

FORMULARZE I WALIDACJA

SPIS TREŚCI

Spis treści	1
Cel zajęć.....	1
Rozpoczęcie.....	1
Uwaga	1
Prezentacja tworzenia CRUD.....	2
Budowa CRUD	5
Lokalizacja	6
Pomiary	8
Commit projektu do GIT.....	10
Podsumowanie.....	10

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- Tworzenie typów formularzy.
- Wizualizacja formularzy w Twig.
- Obsługa formularzy w kontrolerze.
- Walidacja danych.

ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie tworzenia formularzy w Symfony. Powtórzenie security is_granted w Twig i atrybutach kontrolerów.

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.

PREZENTACJA TWORZENIA CRUD

Obserwuj wspólnie z resztą grupy. Prowadzący przedstawi komendę make:crud służącą do budowania kompletnego operacji tworzenia, odczytu, aktualizacji i kasowania encji. Komenda przedstawiona zostanie na przykładzie kontrolera LocationController dla encji Location.

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

The class name of the entity to create CRUD (e.g. GentleChef):
> Location

Choose a name for your controller class (e.g. LocationController)
[LocationController]:
>

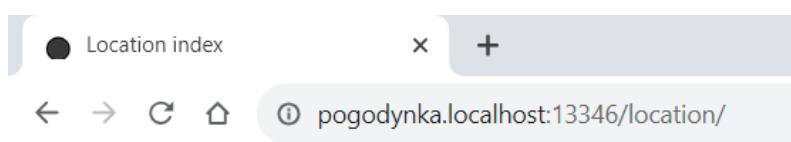
Do you want to generate tests for the controller?. [Experimental] (yes/no) [no]:
>

created: src/Controller/LocationController.php
created: src/Form/LocationType.php
created: templates/location/_delete_form.html.twig
created: templates/location/_form.html.twig
created: templates/location/edit.html.twig
created: templates/location/index.html.twig
created: templates/location/new.html.twig
created: templates/location/show.html.twig

Success!
```

Next: Check your new CRUD by going to /location/

Zgodnie z informacją wyświetlzoną na końcu komendy, można od razu zobaczyć wyniki działania komendy w przeglądarce pod adresem <http://pogodynka.localhost:51641/location>:



ID	City	Country	Latitude	Langitude	actions
1	Szczecin	PL	53.4289	14.553	show edit
2	Police	PL	53.5521	14.5718	show edit

Wspólnie zbadajcie zawartość:

- kontrolera `src/Controller/LocationController.php`;
- typu formularza `src/Form/LocationType.php`;

- widoków templates/location/*.

Wspólnie zmodyfikujcie LocationType, żeby określić typy kontrolek i ich opcje (np. wybór państwa z listy zamiast pola tekstowego). Dodajcie walidacje w dwóch grupach walidacyjnych – dla edit i dla new.

Przykłady:

```
class LocationType extends AbstractType
{
    no usages

    public function buildForm(FormBuilderInterface $builder, array $options): void
    {
        $builder
            ->add( child: 'city', type: null, [
                'attr' => [
                    'placeholder' => 'Enter city name',
                ],
            ])
            ->add( child: 'country', type: ChoiceType::class, [
                'choices' => [
                    'Poland' => 'PL',
                    'Germany' => 'DE',
                    'France' => 'FR',
                    'Spain' => 'ES',
                    'Italy' => 'IT',
                    'United Kingdom' => 'GB',
                    'United States' => 'US',
                ],
            ])
    }
}
```

```
##[Route('/new', name: 'app_location_new', methods: ['GET', 'POST'])]
public function new(Request $request, EntityManagerInterface $entityManager): Response
{
    $location = new Location();
    $form = $this->createForm( type: LocationType::class, $location, [
        'validation_groups' => 'create',
    ]);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager->persist($location);
        $entityManager->flush();

        return $this->redirectToRoute( route: 'app_location_index', [], status: Response::HTTP_SEE_OTHER);
    }

    return $this->render( view: 'location/new.html.twig', [
        'location' => $location,
        'form' => $form,
    ]);
}
```

Na koniec edytujcie twig.yaml oraz base.htm.twig w celu zmiany renderowania strony na Bootstrap 5.

Oczekiwany efekt:

W App

Weather

Measurements

Edit Location

City

Szczecin

Country

Poland

Latitude

53,4289000

Longitude

14,5530000

Update

Delete

[back to list](#)

BUDOWA CRUD

Zbuduj CRUD dla encji Location jak podczas prezentacji oraz analogicznie formularz MeasurementType do tworzenia i edycji prognoz pogody, a następnie kontroler Measurement z akcjami listy, tworzenia, edycji i usuwania. Zaprezentuj osobne walidacje do tworzenia i edycji obu encji. Wykorzystaj konfigurację walidacji w YAML. Na koniec dodaj do szablonu strony (base.html.twig) linki do CRUDA lokalizacji i prognoz.

Wstaw zrzut ekranu listy lokalizacji. Upewnij się, że zastosowano layout Bootstrap 5. Wstaw zrzut wersji desktop i wersji mobilnej:

Location index				
1	Szczecin	PL	53.4289	14.553
2	Police	PL	53.5521	14.5718

Wstaw analogiczne zrzuty dla listy pomiarów:

Measurement index				
1	2024-10-21	18	show edit	
2	2024-10-22	16	show edit	
3	2024-10-21	17	show edit	
4	2024-10-22	20	show edit	

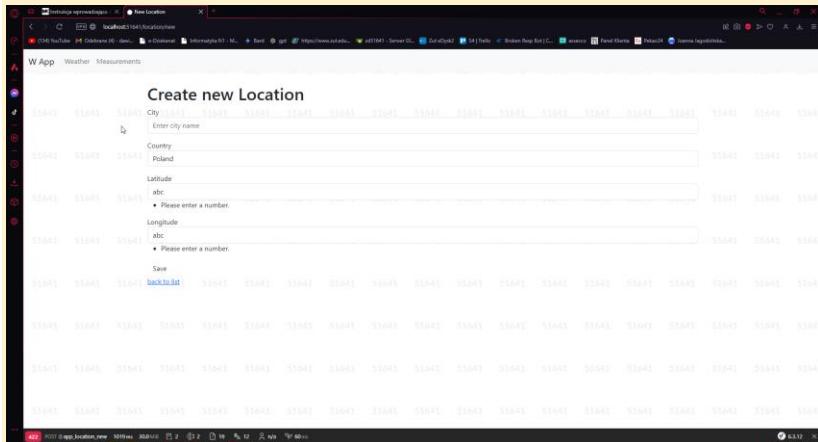
Upewnij się, że na wszystkich powyższych 4 zrzutach ekranu znajdują się linki umożliwiające przejście z lokalizacji do pomiarów i z powrotem.

Punkty:

0	1
---	---

LOKALIZACJA

Wstaw zrzut ekranu tworzenia lokalizacji (tylko desktop). Upewnij się, że na zrzucie ekranu widoczna jest działająca validacja (wyświetlony komunikat o błędzie w którymś polu na formularzu). Uwaga! Tu ma się znaleźć validacja po stronie serwera, a nie frontendowa validacja HTML5 (statystycznie 90% osób prezentuje tutaj złą validację 😊).



Wstaw zrzut ekranu kodu odpowiedzialnego za validację (w formacie YAML).

```
#Route('/new', name: 'app_location_new', methods: ['GET', 'POST'])
public function new(Request $request, EntityManagerInterface $entityManager): Response
{
    $location = new Location();
    $form = $this->createForm(LocationType::class, $location, [
        'validation_groups' => 'create',
    ]);
    $form->handleRequest($request);

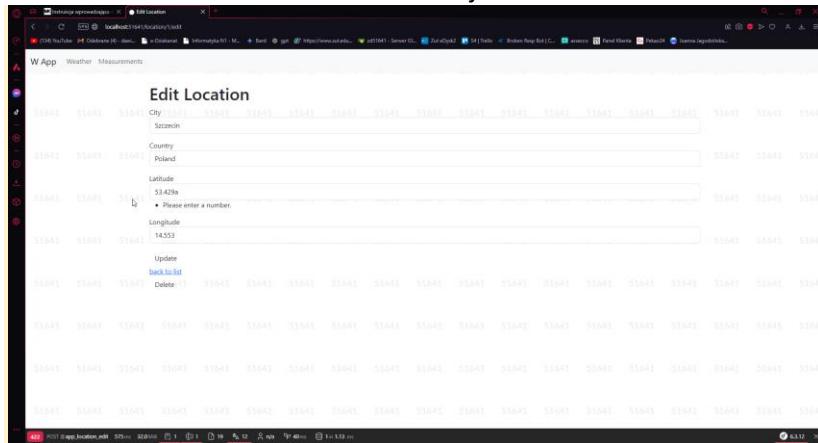
    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager->persist($location);
        $entityManager->flush();

        return $this->redirectToRoute('app_location_index', [], Response::HTTP_SEE_OTHER);
    }
}
```

Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu edycji lokalizacji (tylko desktop). Upewnij się, że na zrzucie ekranu widoczna jest działająca validacja (wyświetlony komunikat o błędzie w którymś polu na formularzu).

AI2 LAB D – Zieliński Dawid – Wersja 1



Wstaw zrzut ekranu kodu odpowiedzialnego za validację (w formacie YAML).

```
#[Route('/{id}/edit', name: 'app_location_edit', methods: ['GET', 'POST'])]
public function edit(Request $request, Location $location, EntityManagerInterface $entityManager): Response
{
    $form = $this->createForm(LocationType::class, $location);
    $form->handleRequest($request);

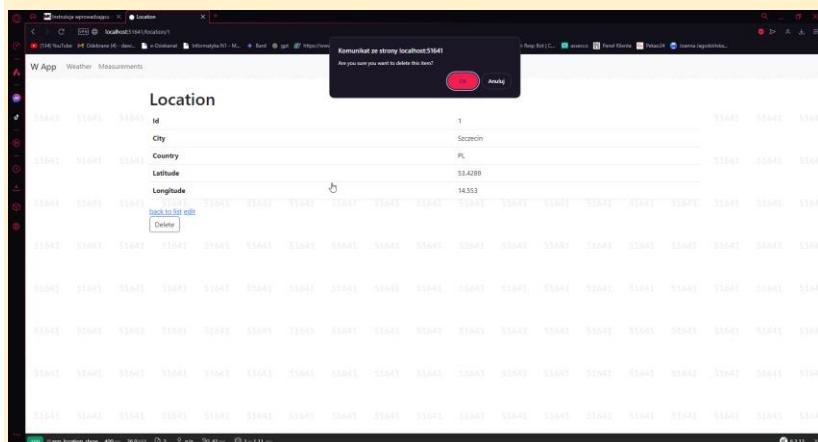
    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager->flush();

        return $this->redirectToRoute('app_location_index', [], Response::HTTP_SEE_OTHER);
    }

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

Punkty:	0	1
---------	---	---

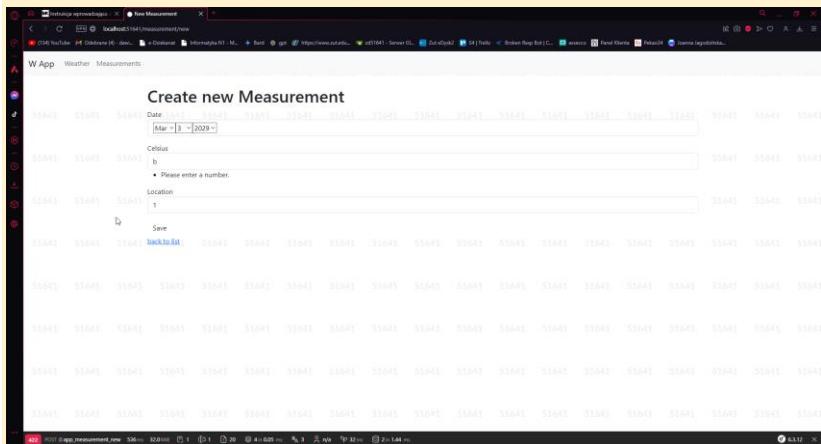
Wstaw zrzuty ekranu kasowania lokalizacji (tylko desktop).



Punkty:	0	1
---------	---	---

POMIARY

Wstaw zrzut ekranu tworzenia pomiarów (tylko desktop). Upewnij się, że na zrzucie ekranu widoczna jest działająca walidacja (wyświetlony komunikat o błędzie w którymś polu na formularzu).



Wstaw zrzut ekranu kodu odpowiedzialnego za validację (w formacie YAML).

```
#Route('/new', name: 'app_measurement_new', methods: ['GET', 'POST'])
public function new(Request $request, EntityManagerInterface $entityManager): Response
{
    $measurement = new Measurement();
    $form = $this->createForm(MeasurementType::class, $measurement);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager->persist($measurement);
        $entityManager->flush();

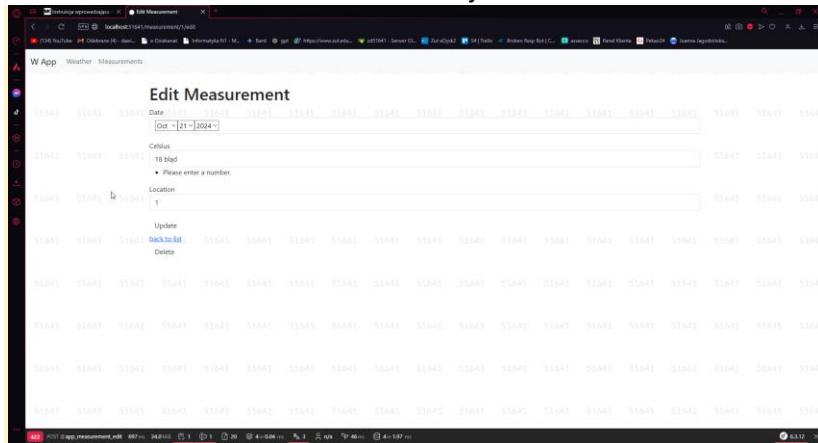
        return $this->redirectToRoute('app_measurement_index', [], Response::HTTP_SEE_OTHER);
    }

    return $this->render('measurement/new.html.twig', [
        'measurement' => $measurement,
        'form' => $form,
    ]);
}
```

Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu edycji pomiarów (tylko desktop). Upewnij się, że na zrzucie ekranu widoczna jest działająca walidacja (wyświetlony komunikat o błędzie w którymś polu na formularzu).

AI2 LAB D – Zieliński Dawid – Wersja 1



Wstaw zrzut ekranu kodu odpowiedzialnego za validację (w formacie YAML).

```
#[Route('/{id}/edit', name: 'app_measurement_edit', methods: ['GET', 'POST'])]
public function edit(Request $request, Measurement $measurement, EntityManagerInterface $entityManager): Response
{
    $form = $this->createForm(MeasurementType::class, $measurement);
    $form->handleRequest($request);

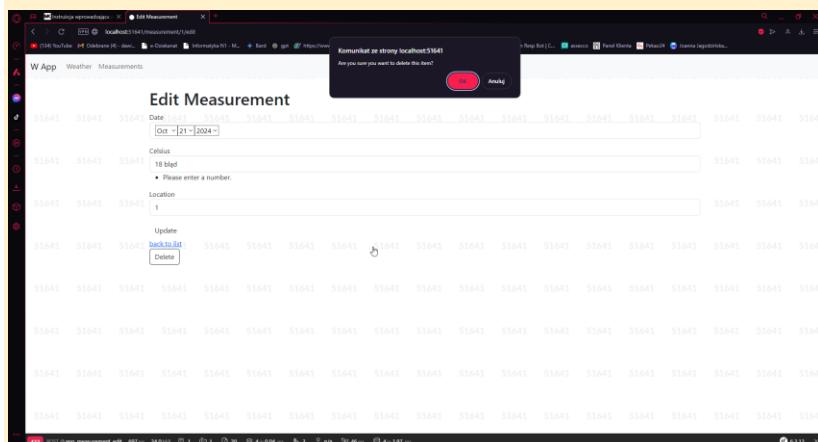
    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager->flush();

        return $this->redirectToRoute('app_measurement_index', [], Response::HTTP_SEE_OTHER);
    }

    return $this->render('measurement/edit.html.twig', [
        'measurement' => $measurement,
        'form' => $form,
    ]);
}
```

Punkty:	0	1
---------	---	---

Wstaw zrzuty ekranu kasowania pomiarów (tylko desktop).



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-d na podstawie głównej gałęzi kodu.

Podaj link do brancha lab-d w swoim repozytorium:

...link, np. <https://github.com/ideaspot-pl/ai2-pogodynka-202310/tree/lab-c...>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.