



Dokumentacja

Przedmiot:	Usługi Webowe
Temat zajęć:	Projekt Api
Data:	23 stycznia 2018 roku
Imię i nazwisko:	Maciej Zawadzki
Rok, grupa:	3 rok, informatyka, studia stacjonarne

Cel ćwiczenia

Celem ćwiczenia jest stworzenie usługi webowej, która zapewni nam funkcjonalność taką jak dodawanie, usuwanie, pobieranie i edytowanie danych za pomocą zapytań HTTP. Język, w jakim utworzona została usługa, to PHP przy użyciu frameworka Phpstorm. Przechowywanie danych odbywa się za pomocą bazy danych mysql zarządzanej przez Phpmyadmin.

Spis treści

Funkcjonalności usługi	2
Baza danych	2
Dodatki użyte w aplikacji	3
Routing	3
Kontrolery	5
Wygląd	10

Funkcjonalności usługi

- Operacje typu CRUD (create, read, update and delete) za pomocą zapytań API oraz interfejsu,
- Interfejs stworzony z wykorzystaniem Bootstrapa 4.0.0,
- JSON.

Baza danych

Baza danych składa się z dwóch tabel: Produkt i Product. Obie tabele zawierają podstawowe dane, takie jak Nazwa, Cena, Typ, Utworzono i Zaktualizowano.

Tabela Produkt zawiera następujące pola:

- ID,
- Nazwa,
- Cena,
- Waga,
- Hak,
- Producent haka,
- Typ
- Stworzono (createdAt),
- Zaktualizowano(updatedAt).

Tabela Product zawiera następujące pola:

- ID,
- Nazwa,
- Cena,
- Typ,
- Utworzono(createdAt),
- Zaktualizowano(updatedAt).

Dodatki użyte w aplikacji

- Bootstrap 4.0.0 - framework umożliwiający tworzenie responsywnych stron WWW, ułatwiający odświeżanie stron. Zawiera wiele gotowych komponentów HTML. Wylicza marginesy, tworzy formy i ramki, zapewnia responsywność na tablecie, komputerze, smartfonie. Opiera swoje działanie na strukturze siatki.

Routing

Dzięki niemu jesteśmy w stanie tworzyć w aplikacji dynamiczne URL-e, ułatwia to zarządzanie funkcjonalnością aplikacji.

```
student@student-virtual-machine:/var/www/symfony$ bin/console debug:router
```

Name	Method	Scheme	Host	Path
_wdt	ANY	ANY	ANY	/_wdt/{token}
_profiler_home	ANY	ANY	ANY	/_profiler/
_profiler_search	ANY	ANY	ANY	/_profiler/search
_profiler_search_bar	ANY	ANY	ANY	/_profiler/search_bar
_profiler_phpinfo	ANY	ANY	ANY	/_profiler/phpinfo
_profiler_search_results	ANY	ANY	ANY	/_profiler/{token}/search/results
_profiler_open_file	ANY	ANY	ANY	/_profiler/open
_profiler	ANY	ANY	ANY	/_profiler/{token}
_profiler_router	ANY	ANY	ANY	/_profiler/{token}/router
_profiler_exception	ANY	ANY	ANY	/_profiler/{token}/exception
_profiler_exception_css	ANY	ANY	ANY	/_profiler/{token}/exception.css
_twig_error_test	ANY	ANY	ANY	/_error/{code}.{_format}
json_index	ANY	ANY	ANY	/json
json_index2	ANY	ANY	ANY	/json2
homepage	ANY	ANY	ANY	/
product_index	GET	ANY	ANY	/product/
product_new	GET POST	ANY	ANY	/product/new
product_sort	GET	ANY	ANY	/product/sortbyid57
product_show	GET	ANY	ANY	/product/{id}
product_edit	GET POST	ANY	ANY	/product/{id}/edit
product_delete	DELETE	ANY	ANY	/product/{id}
produkt_sort	GET	ANY	ANY	/produkt/sortbyid57
produkt_index	GET	ANY	ANY	/produkt/
produkt_new	GET POST	ANY	ANY	/produkt/new
produkt_show	GET	ANY	ANY	/produkt/{id}
produkt_edit	GET POST	ANY	ANY	/produkt/{id}/edit
produkt_delete	DELETE	ANY	ANY	/produkt/{id}

```
student@student-virtual-machine:/var/www/symfony$
```

Wpisanie `http://127.0.0.1:8000/produkt/` w przeglądarce da nam następujący efekt:

← → ↻ 127.0.0.1:8000/produkt/ ☆ ↻

Aplikacje phpMyAdmin phpinfo()

Lista produktów

JSON

Id	Nazwa	Cena [zł]	Waga [g]	Hak	Producent haka	Typ	Stworzono	Zaktualizowano	Akcje
1	Główka jigowa	0.40	20.00	3/0	Mustad	Okoniowa	2018-01-01 00:00:00	2018-01-01 00:02:00	<a>Pokaż <a>Edytuj
2	Ciężarek	1.00	500.00			Ciężki	2013-01-01 00:00:00	2013-01-01 00:00:00	<a>Pokaż <a>Edytuj
56	Główka jigowa	0.50	1.00	2/0	Mustad	Okoniowa	2013-01-01 00:00:00	2013-01-01 00:00:00	<a>Pokaż <a>Edytuj
57	Kotwiczka	2.00	0.10			Do zaczepów	2013-01-01 00:00:00	2013-01-01 00:00:00	<a>Pokaż <a>Edytuj

Jak widać na poprzednim zrzucie, nasza strona główna ma adres `http://127.0.0.1:8000/` (/ w routingu)



Witaj na Rybaczkach!

Czy myślałeś kiedyś o tym...
Jak łowić ryby?



Kontrolery

Aplikacja posiada trzy kontrolery:

- ProductController.php,
- ProduktController.php,
- DefaultController.php.

```
/**
 * Lists all produkt entities.
 *
 * @Route("/sortbyid57", name="product_sort")
 * @Method("GET")
 */
public function sortByid57()
{
    $em = $this->getDoctrine()->getManager();
    $products = $em->getRepository('AppBundle:Product')->findBy([
        'id' => 57 ]);
    return $this->render('product/index.html.twig', array(
        'products' => $products,
    ));
}
```

Wpisanie tego kodu w ProductController.php umożliwi nam wyświetlenie produktu o id = 57, gdy wpiszemy w przeglądarce adres <http://127.0.0.1:8000/product/sortbyid57>



Lista produktów

JSON

Id	Nazwa	Cena [zł]	Typ	Utworzono	Zaktualizowano	Akcje
----	-------	-----------	-----	-----------	----------------	-------

Stwórz

Wskaż przedmiot o numerze id = 57

Powrót na stronę główną

Jak widać, nie mamy produktu o takim id.

```

/**
 * Creates a new product entity.
 *
 * @Route("/new", name="product_new")
 * @Method({"GET", "POST"})
 */
public function newAction(Request $request)
{
    $product = new Product();
    $form = $this->createForm( type: 'AppBundle\Form\ProductType', $product);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $em->persist($product);
        $em->flush();

        return $this->redirectToRoute( route: 'product_show', array('id' => $product->getId()));
    }

    return $this->render( view: 'product/new.html.twig', array(
        'product' => $product,
        'form' => $form->createView(),
    ));
}

```

Ten fragment kodu w ProductController.php daje nam możliwość tworzenia nowych produktów po wpisaniu `http://127.0.0.1:8000/product/new`. Tak samo będzie to działać dla kontrolera ProduktController.php, gdy wpisze `http://127.0.0.1:8000/produkt/new`:



Tworzenie produktu

Name

Price

Weight

Hook

Hookproducent

Type

Created at

Jan ▼ 1 ▼ 2013 ▼

00 ▼ : 00 ▼

Updated at

Jan ▼ 1 ▼ 2013 ▼

00 ▼ : 00 ▼

- [Wróć do spisu artykułów](#)

```

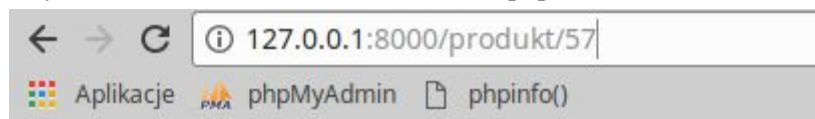
/**
 * Finds and displays a product entity.
 *
 * @Route("/{id}", name="product_show")
 * @Method("GET")
 */
public function showAction(Product $product)
{
    $deleteForm = $this->createDeleteForm($product);

    return $this->render( view: 'product/show.html.twig', array(
        'product' => $product,
        'delete_form' => $deleteForm->createView(),
    ));
}

```

Funkcja showAction umożliwia ukazanie naszego produktu, wraz z możliwością jego usunięcia i edycji.

Przykład dla kontrolera ProduktController.php:



Produkty

Id 57
Nazwa Kotwiczka
Cena 2.00
Waga 0.10
Hak
Producent haka
Typ Do zaczepów
Stworzono 2013-01-01 00:00:00
Zaktualizowano 2013-01-01 00:00:00

- [Wróć do listy artykułów](#)
- [Edytuj](#)
-


```

/**
 * Displays a form to edit an existing product entity.
 *
 * @Route("/{id}/edit", name="product_edit")
 * @Method({"GET", "POST"})
 */
public function editAction(Request $request, Product $product)
{
    $deleteForm = $this->createDeleteForm($product);
    $editForm = $this->createForm( type: 'AppBundle\Form\ProductType', $product);
    $editForm->handleRequest($request);

    if ($editForm->isSubmitted() && $editForm->isValid()) {
        $this->getDoctrine()->getManager()->flush();

        return $this->redirectToRoute( route: 'product_edit', array('id' => $product->getId()));
    }

    return $this->render( view: 'product/edit.html.twig', array(
        'product' => $product,
        'edit_form' => $editForm->createView(),
        'delete_form' => $deleteForm->createView(),
    ));
}

```

Za pomocą editAction w kontrolerze ProductController.php możemy edytować nasz produkt. Ponadto mamy możliwość jego usunięcia w trybie edycji.

Przykład dla kontrolera ProduktController.php dla produktu o ID = 1.

Edytuj produkt

Name

Price

Weight

Hook

Hookproducent

Type

Created at

Jan ▼ 1 ▼ 2018 ▼

00 ▼ : 00 ▼

Updated at

Jan ▼ 1 ▼ 2018 ▼

00 ▼ : 02 ▼

[Wróć do spisu artykułów](#)


```

/**
 * Deletes a product entity.
 *
 * @Route("/{id}", name="product_delete")
 * @Method("DELETE")
 */
public function deleteAction(Request $request, Product $product)
{
    $form = $this->createDeleteForm($product);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $em->remove($product);
        $em->flush();
    }

    return $this->redirectToRoute( route: 'product_index');
}

```

Funkcja deleteAction umożliwia nam usuwanie produktów, ukazuje się między innymi po wejściu w akcję "edytuj" czy też "pokaż".

```

/**
 * Lists all product entities.
 *
 * @Route("/", name="product_index")
 * @Method("GET")
 */
public function indexAction()
{
    $em = $this->getDoctrine()->getManager();

    $products = $em->getRepository( className: 'AppBundle:Product')->findAll();

    return $this->render( view: 'product/index.html.twig', array(
        'products' => $products,
    ));
}

```

Funkcja indexAction wyświetla nam wszystkie rekordy z tabeli.



Lista produktów

JSON						
Id	Nazwa	Cena [zł]	Typ	Utworzono	Zaktualizowano	Akcje
1	Robaczki 300 sztuk	20.00	1	2013-01-01 00:00:00	2013-01-01 00:00:00	<ul style="list-style-type: none"> Pokaż Edytuj
2	Zielone przypony	30.00	6	2013-01-01 00:00:00	2013-01-01 00:00:00	<ul style="list-style-type: none"> Pokaż Edytuj
3	Kalosze	30.00	1	2013-01-23 00:00:00	2013-01-01 00:00:00	<ul style="list-style-type: none"> Pokaż Edytuj
4	Rybaczki	40.00	1	2018-01-23 00:00:00	2013-01-01 00:00:00	<ul style="list-style-type: none"> Pokaż Edytuj
5	Ochroniacze na łokcie	33.00	1	2018-01-23 00:00:00	2013-01-01 00:00:00	<ul style="list-style-type: none"> Pokaż Edytuj

