Wyższa Szkoła Informatyki i Umiejętności w Łodzi

Przedmiot: Programowanie komponentowe dla WWW

Temat: Pierwszy servlet.

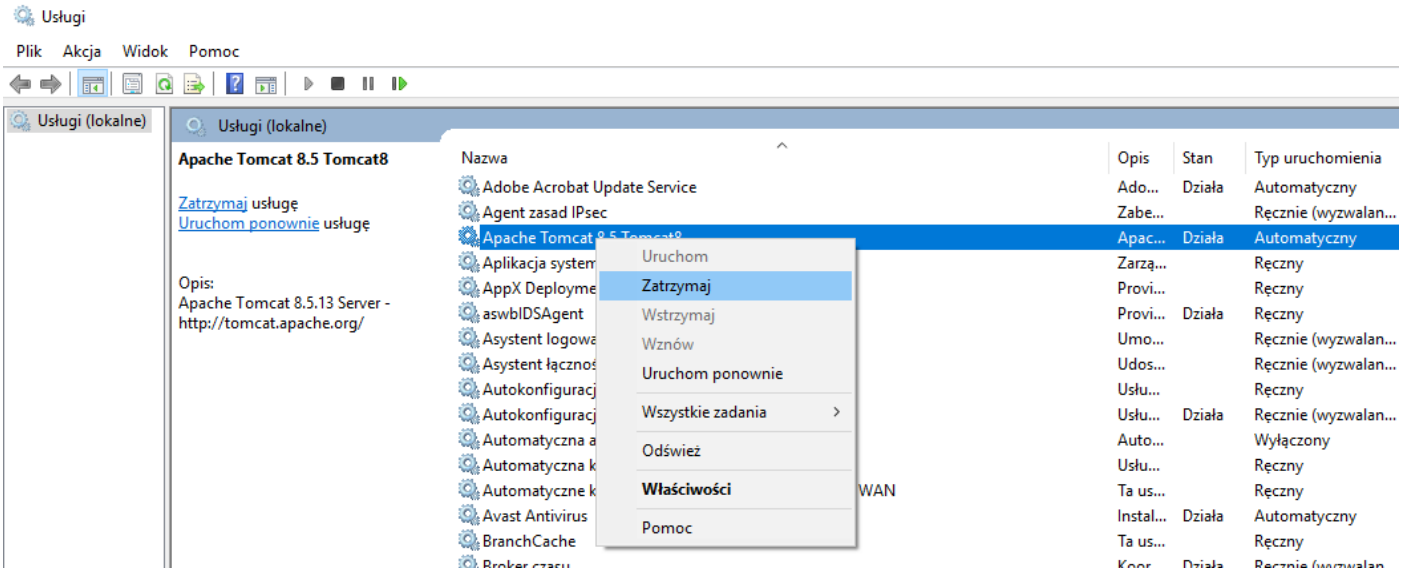
Autor Łukasz Budrewicz – 30525

Spis treści

1. Przygotowanie serwisu Tomcat

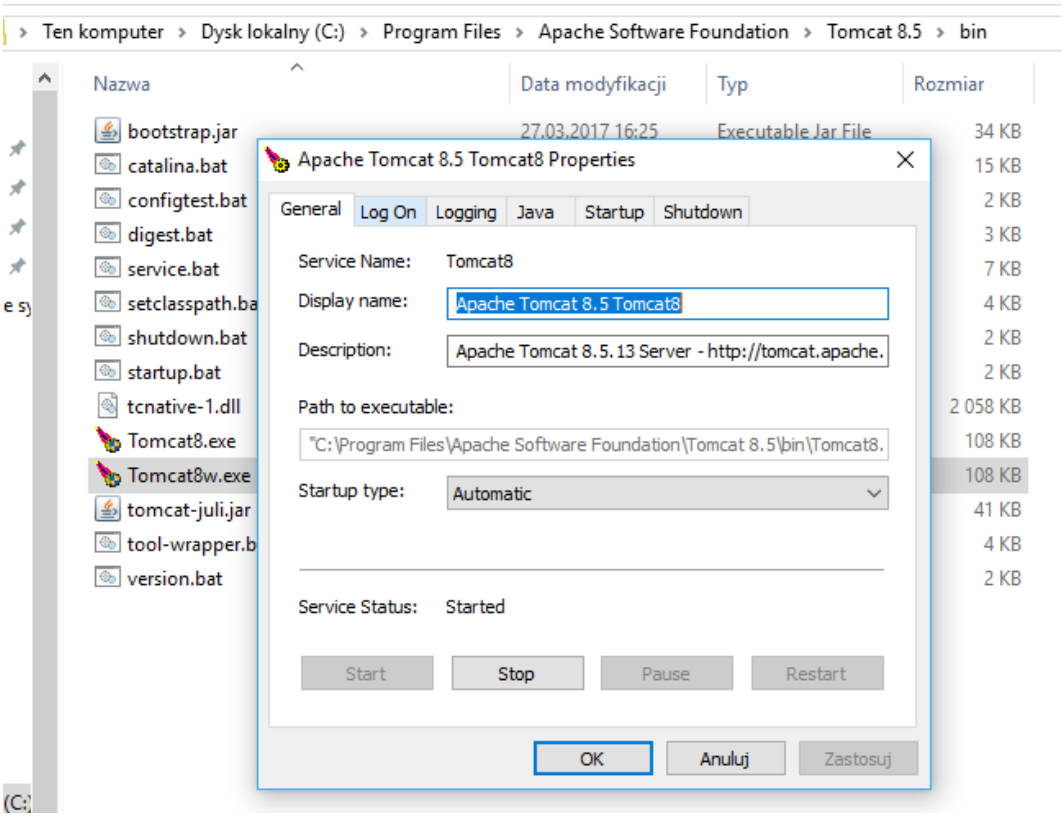
Na początek zatrzymujemy serwis (po to by przygotować sobie go później z poziomu Eclipsa). Możemy to wykonać na dwa sposoby.

Pierwszy poprzez systemowe „Usługi” gdzie możemy wyszukać nasz serwis Tomcat, w zależności od tego jak go nazwaliśmy podczas instalacji/w oprogramowaniu Tomcat.



Rysunek 1 Zatrzymanie procesu

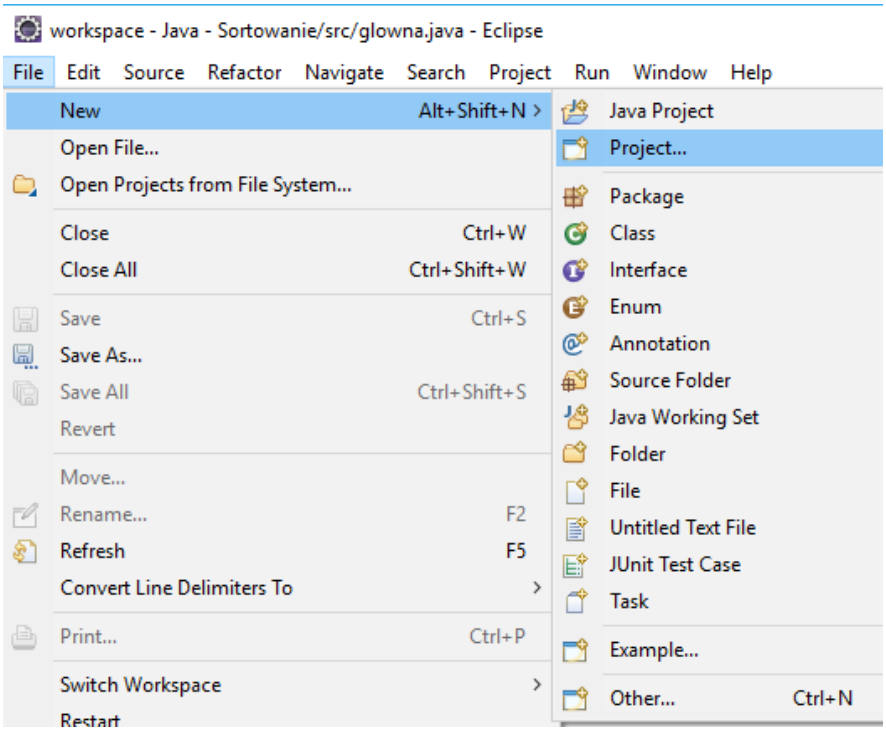
Drugi sposób przez nasze oprogramowanie Tomcat, gdzie również widzimy jego status.



Rysunek 2 Drugi sposób

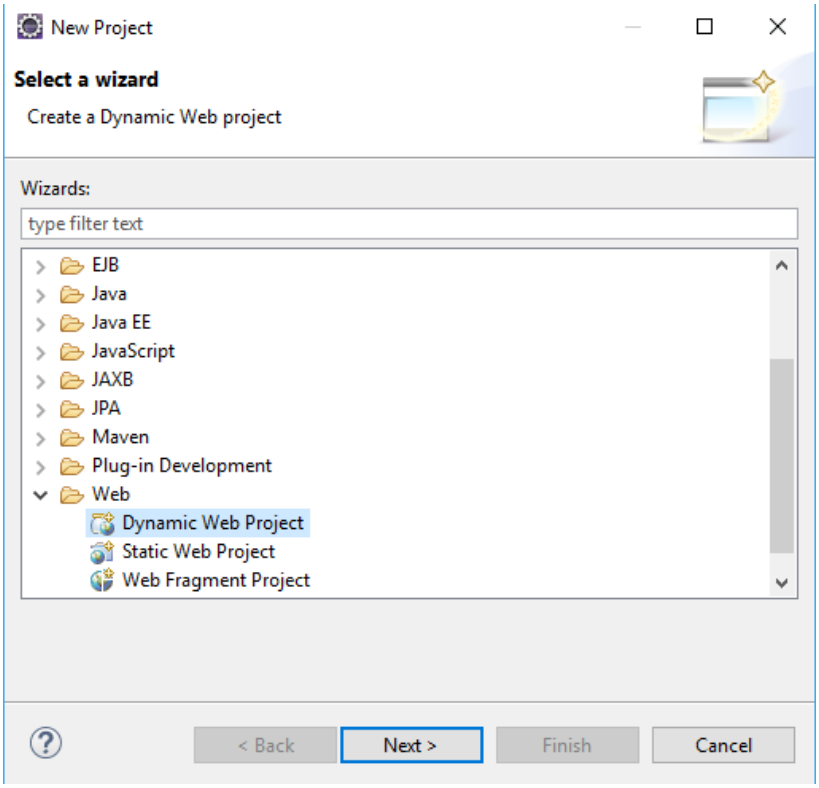
2. Tworzenie pierwszej aplikacji WWW

Otwieramy Eclipsa, gdzie tworzymy nowy projekt.

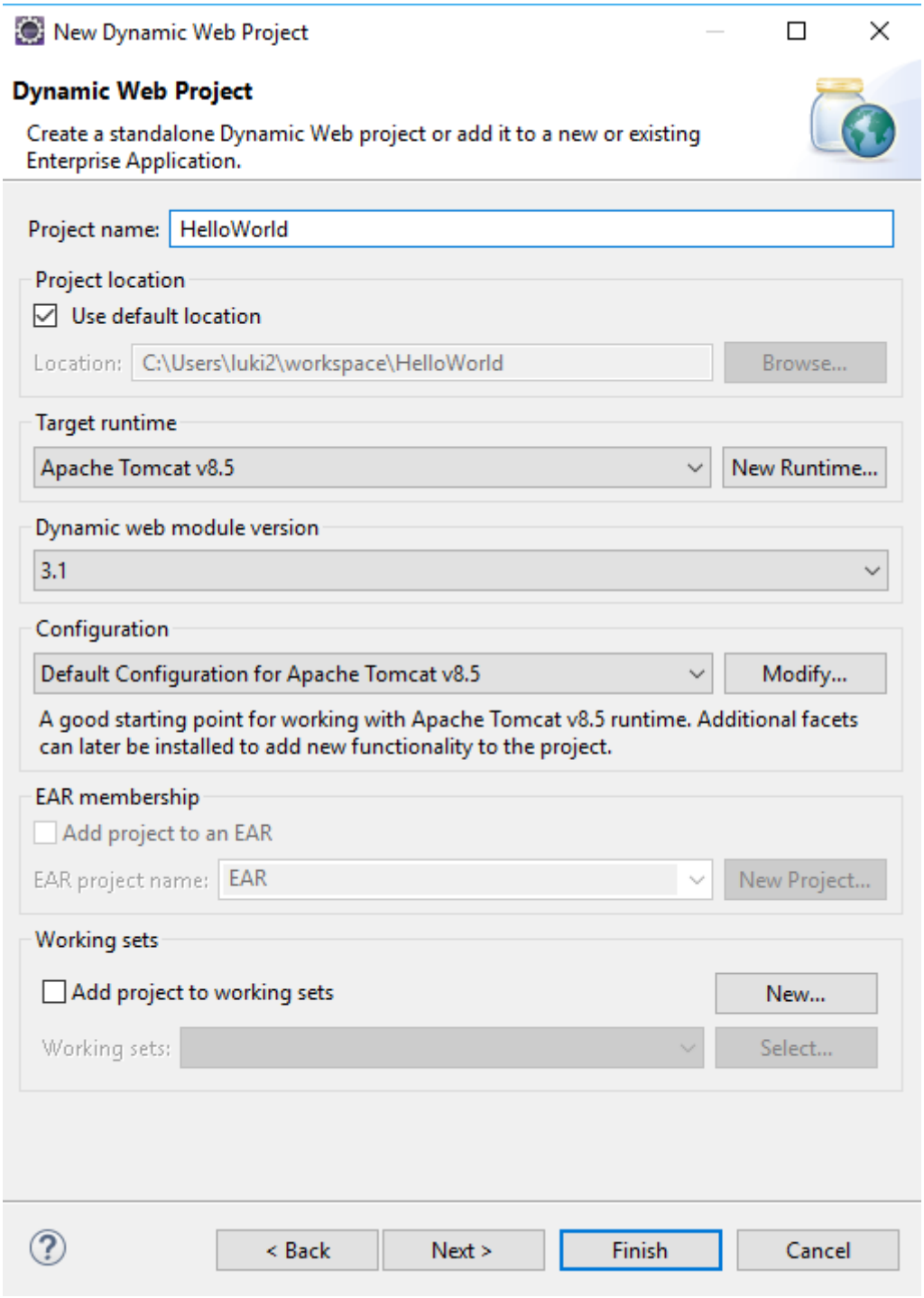


Rysunek 3 Tworzenie nowego projektu

Z dostępnego menu wybieramy Web > Dynamic Web Project



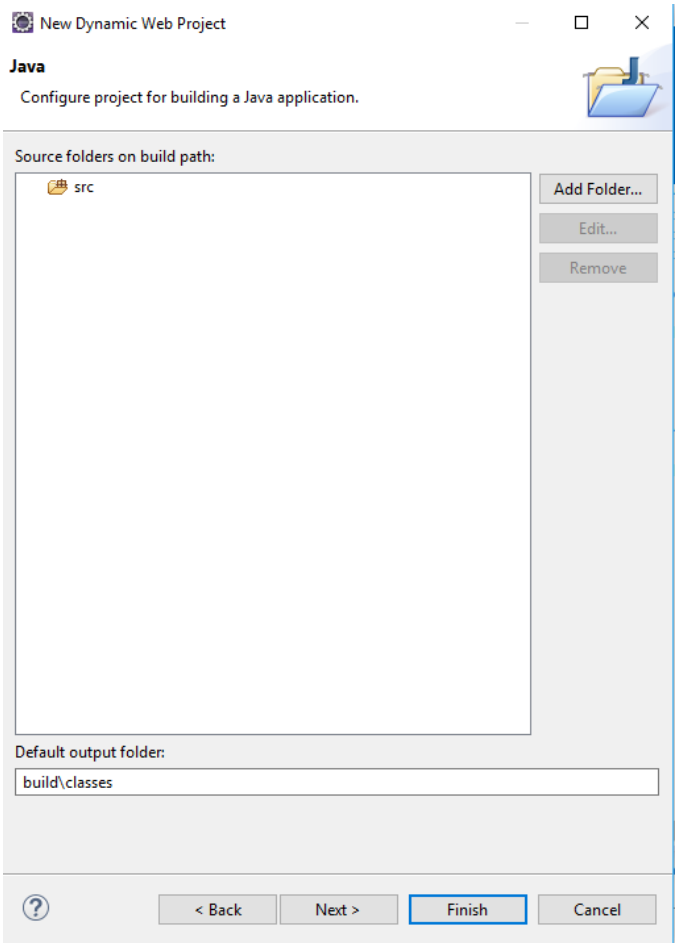
Rysunek 4 Aplikacja WWW



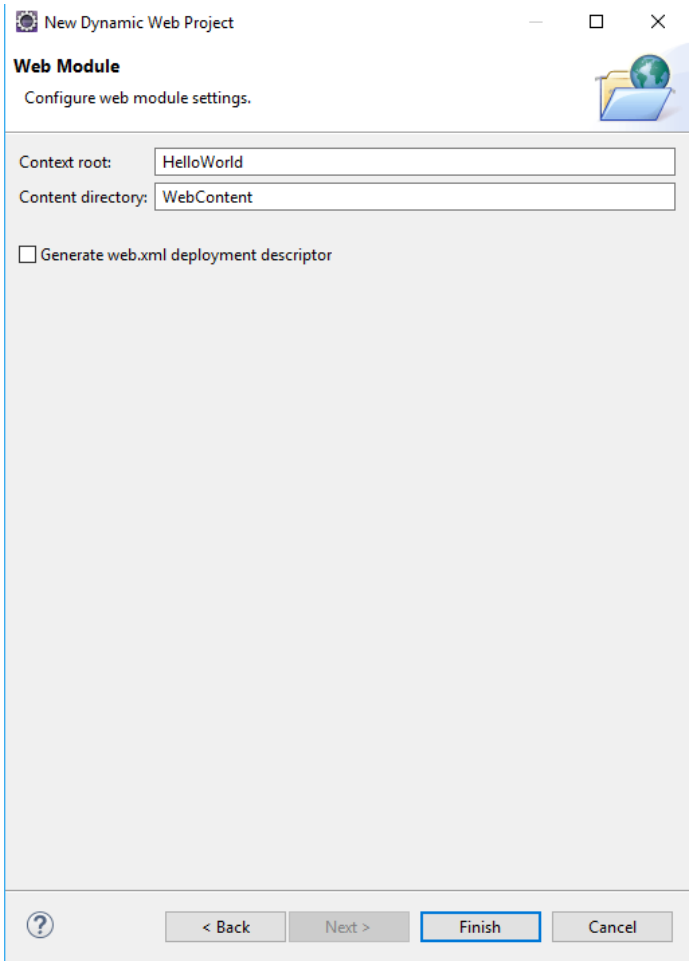
Rysunek 5 Kreator

Ukaże nam się kreator, gdzie większość opcji jest już gotowa. Z naszej strony nie jest konieczne żadne ustawianie, jeśli wcześniej wszystko ustawialiśmy wg. instrukcji.

Jedynie musimy uzupełnić nazwę naszego projektu, wpisujemy „HelloWorld”.

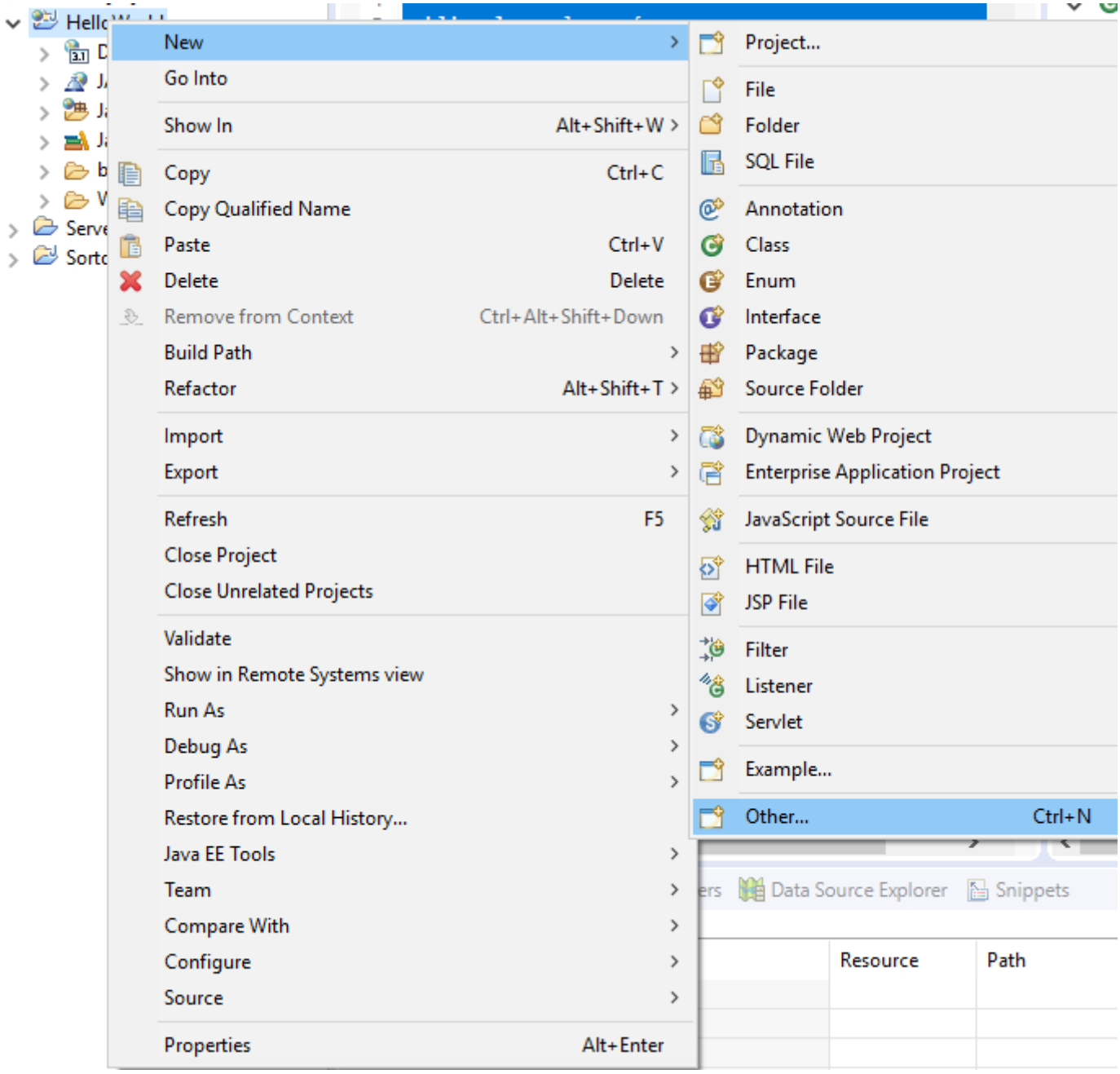


Rysunek 6 Lokalizacja Javy

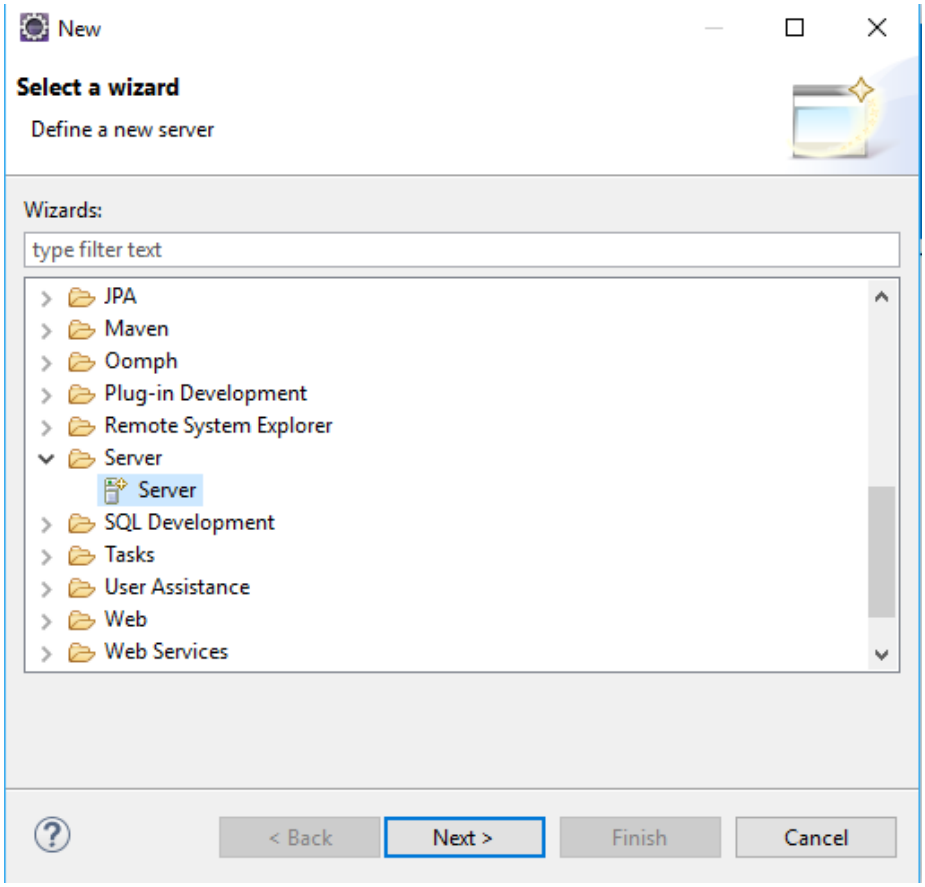


Rysunek 7 Nazwa programu

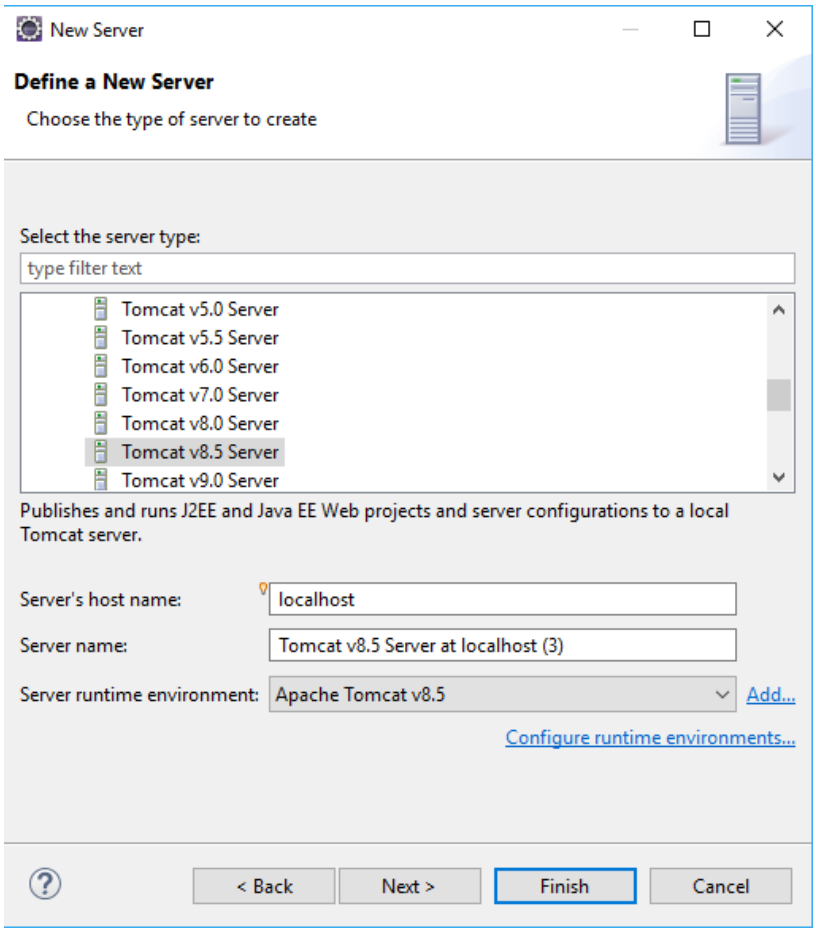
Następnym krokiem jest podpięcie wcześniej zdefiniowanego serwera Tomcat.



Rysunek 8 Podpięcie serwera Tomcat

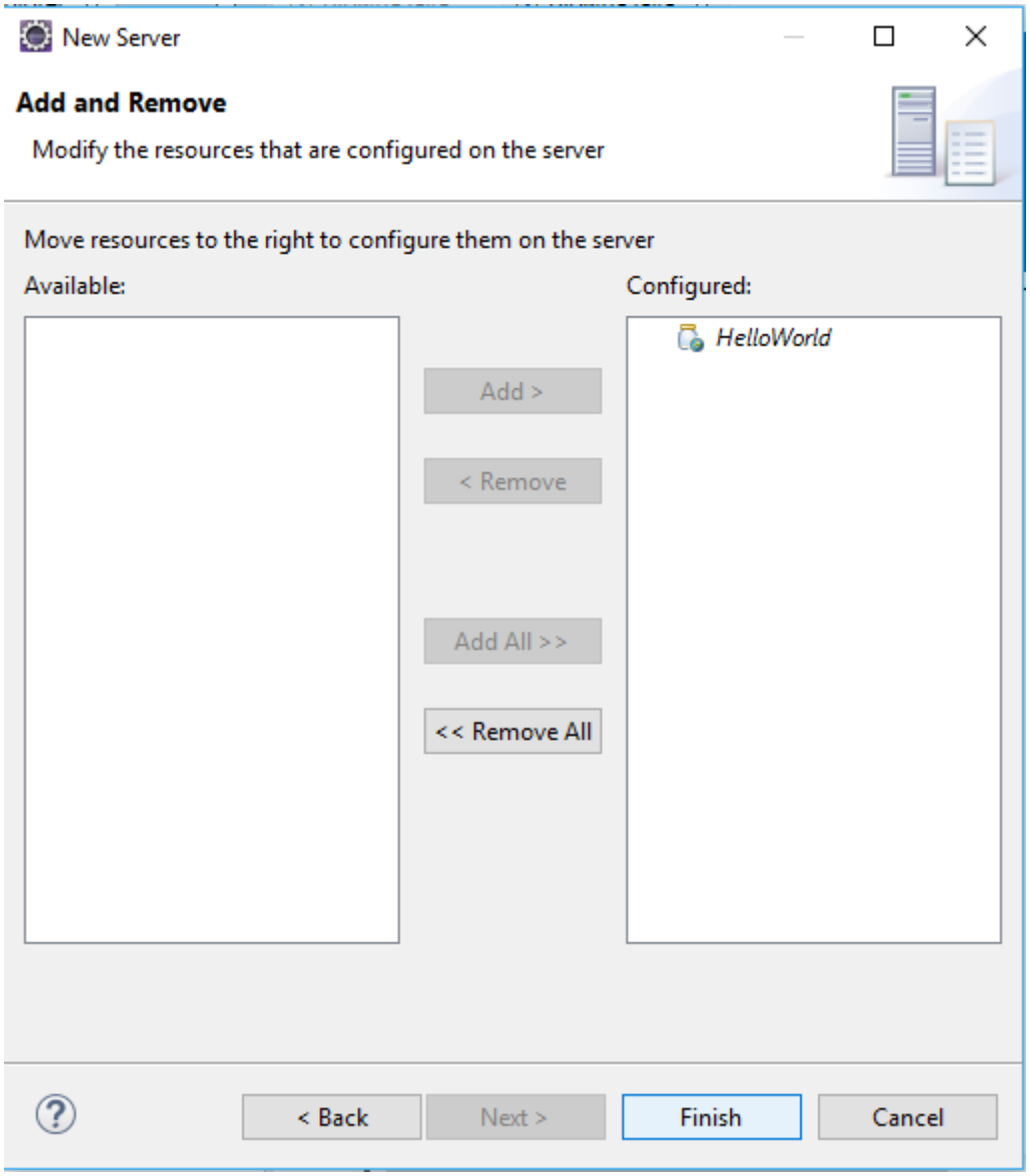


Rysunek 9 Serwer Tomcat



Rysunek 10 Serwer Tomcat

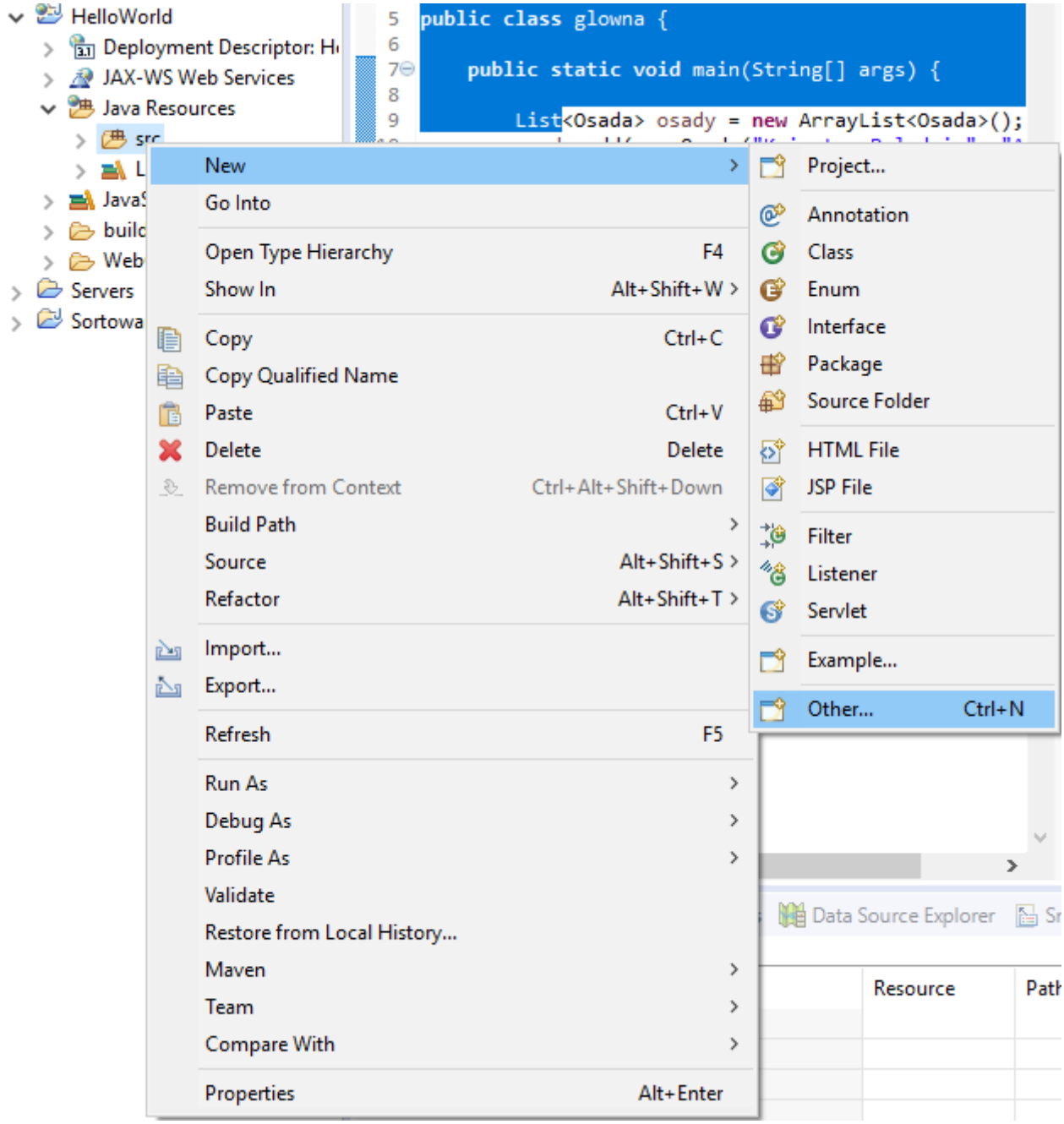
W ostatnim okienku należy przerzucić zasób naszej aplikacji WWW by była obsługiwana przez serwer Tomcat.



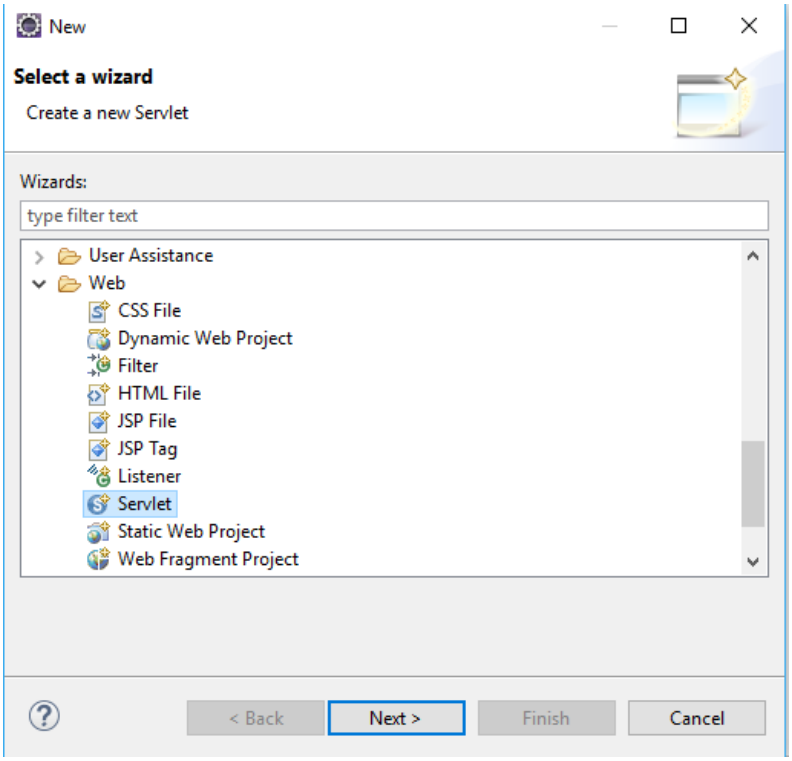
Rysunek 11 Podpięcie programu

3. Pierwszy Servlet

Do naszej aplikacji WWW dodajemy komponent „Servlet”.



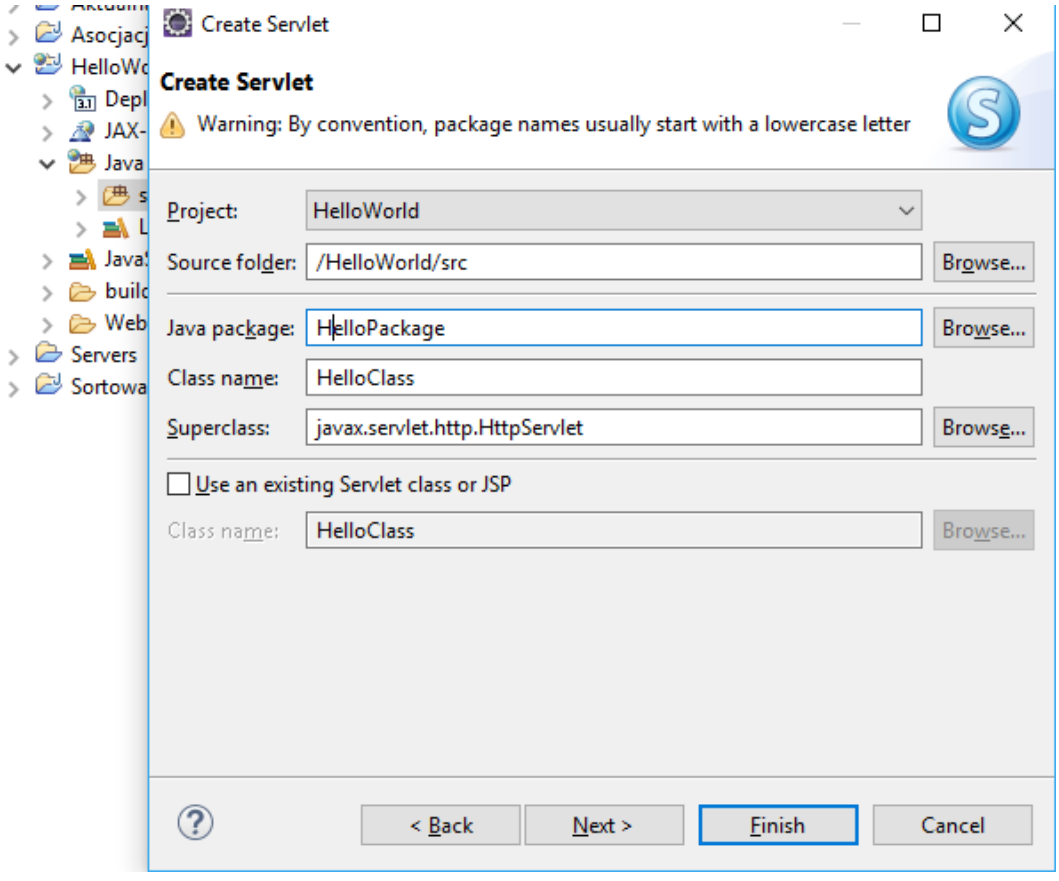
Rysunek 12 Dodawanie Servlet'a



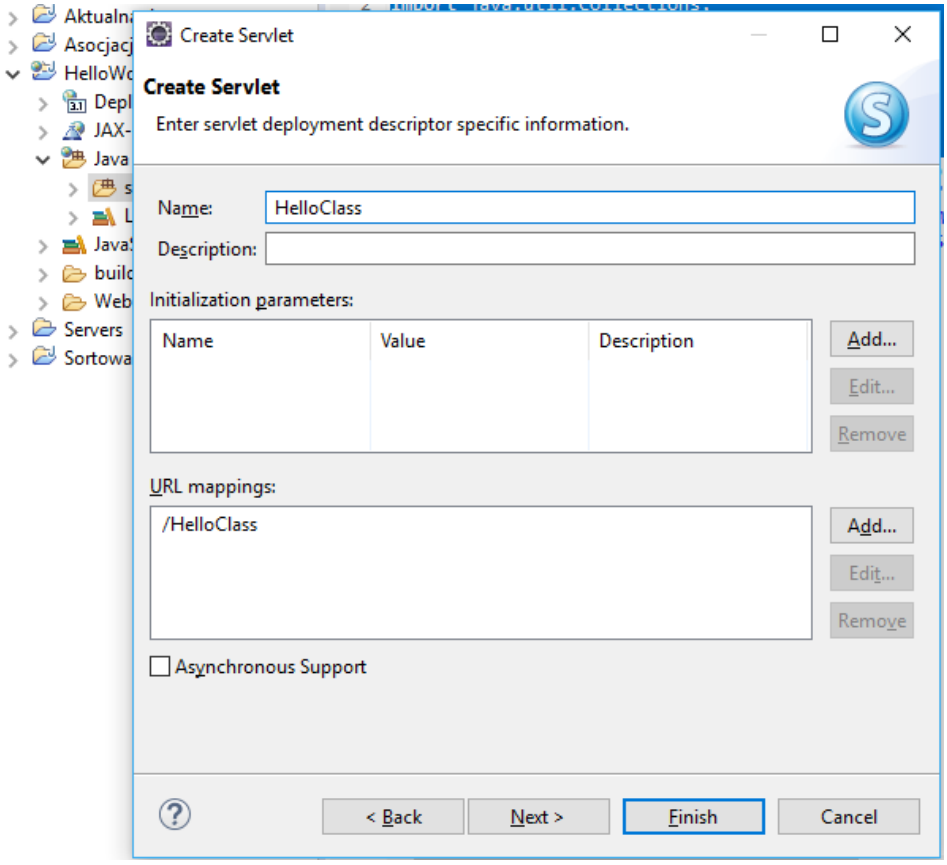
Rysunek 13 Servlet

Z dostępnego menu wybieramy nasz pierwszy Servlet.

W następnym oknie ważne jest aby nazwać nasz pakiet w którym będzie się mieścić nasza klasa „Servlet” jak i sam Servlet. Używamy tutaj nazwy pakietu „HelloPackage” oraz klasy „HelloClass”. W tym momencie ukaże się nam komunikat o złym przyjętym nazewnictwie, nie jest to błąd lecz konwencje programowania zalecają by nazwa pakietu zaczynała się z małej litery, w związku z tym wprowadzamy małą korektę.

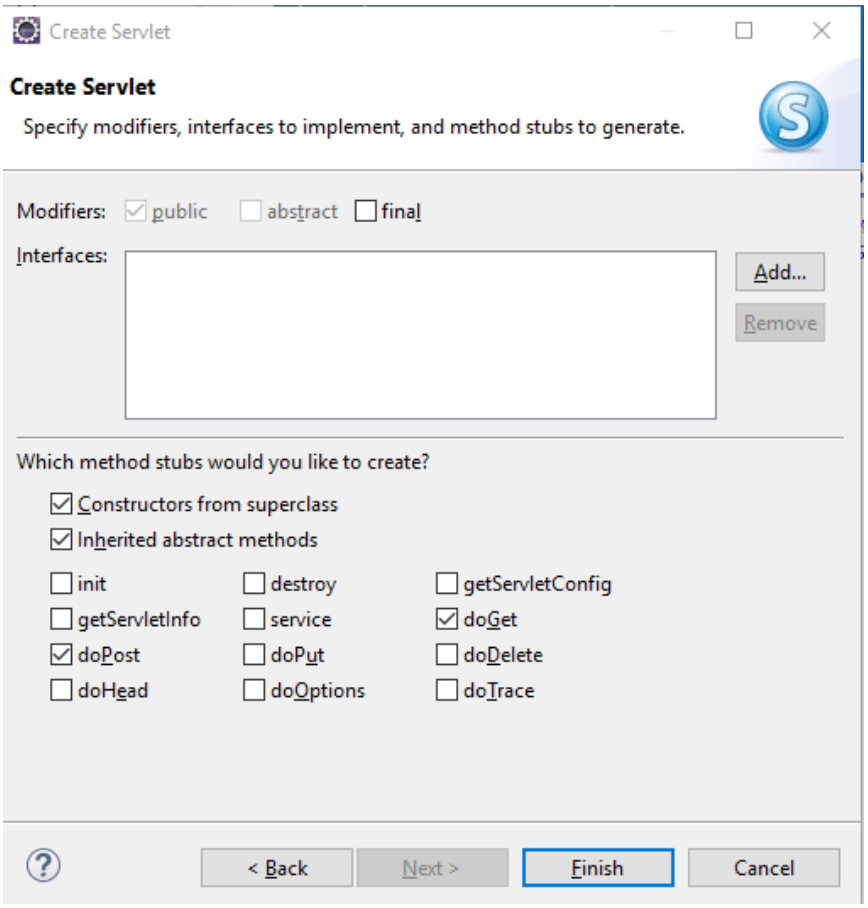


Rysunek 14 Nazewnictwo

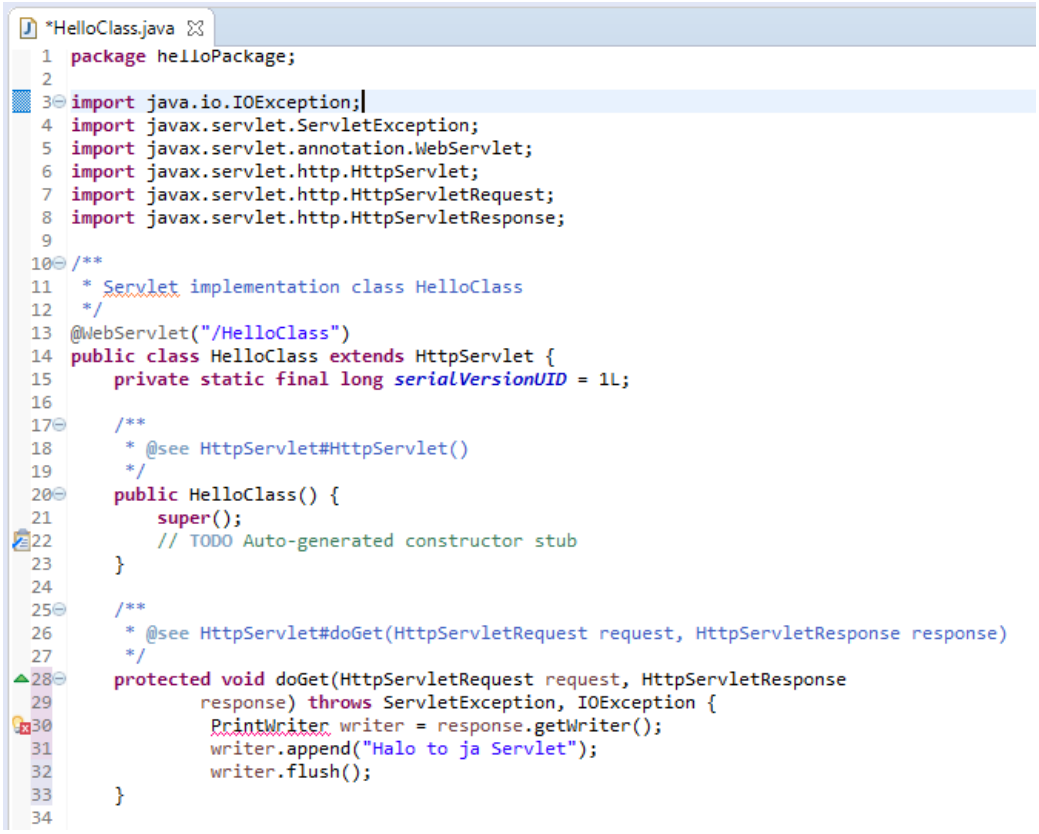


Rysunek 15 Złe nazewnictwo

Na koniec możemy jeszcze ustawić pod jakim adresem nasza klasa będzie dostępna, można by rzec trasa routingu. Domyślnie jest ustawiona na taką jak nazwa naszej klasy. Na koniec zaznaczamy jeszcze jakie metody kompilator powinien zaimplementować podczas tworzenia klasy, nic nie stoi na przeszkodzie by je dodać później.



Rysunek 16 Dodawanie metod

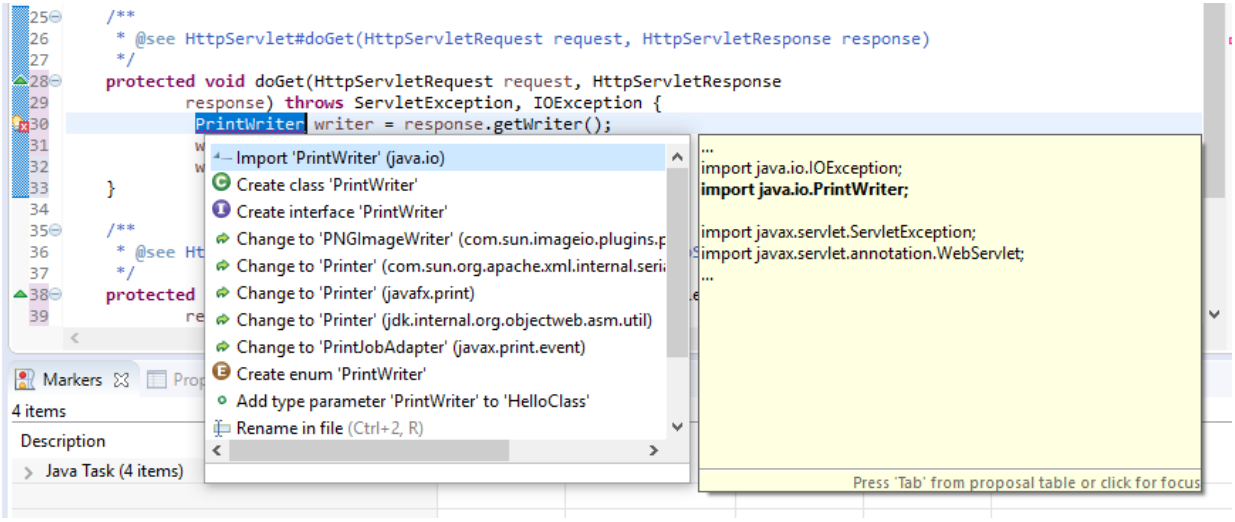


Rysunek 17 Nasz pierwszy program

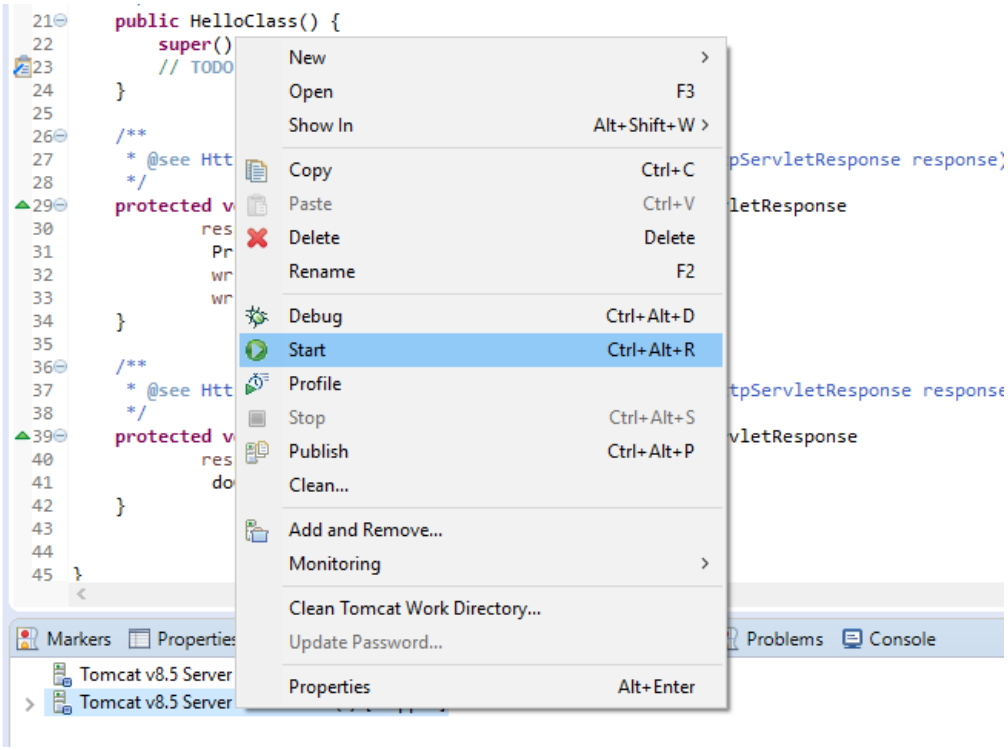
```
protected void doGet(HttpServletRequest request, HttpServletResponse
            response) throws ServletException, IOException {
             PrintWriter writer = response.getWriter();
             writer.append("Halo to ja Servlet");
             writer.flush();
    }

protected void doPost(HttpServletRequest request, HttpServletResponse
                  response) throws ServletException, IOException {
                   doGet(request, response);
      }
```

Podmieniamy domyślny kod. Na ten zamieszony powyżej.

Po utworzeniu klasy należy jeszcze dodatkowo, zaimportować bibliotekę „PrintWriter”.



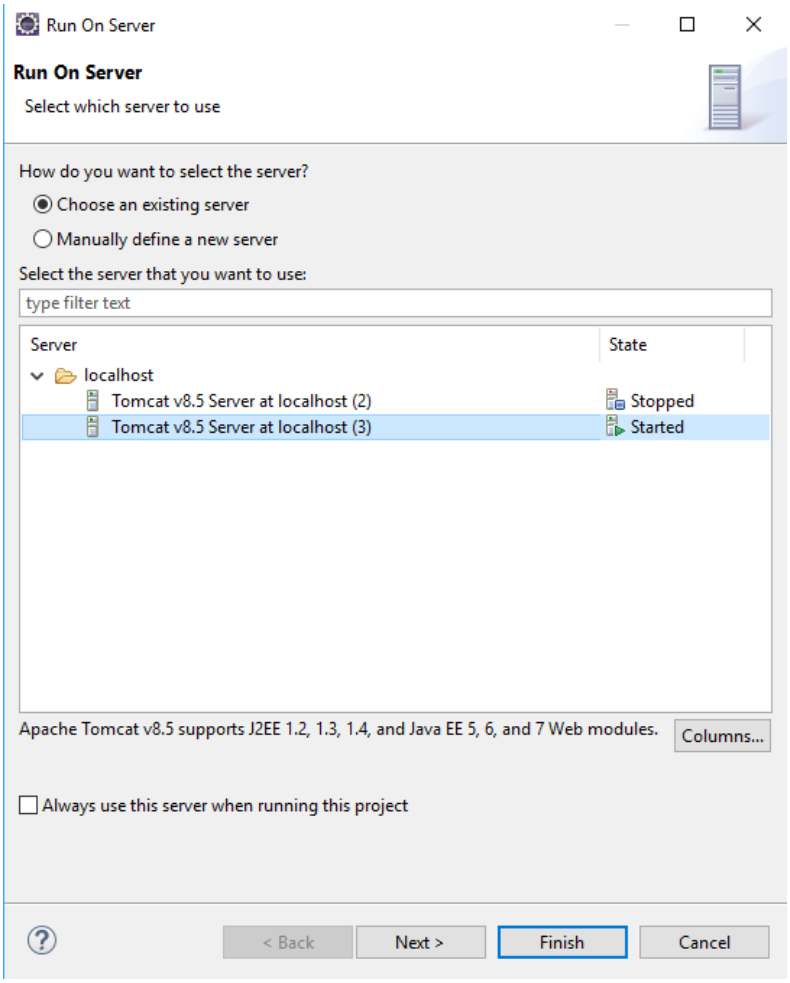
Rysunek 18 Import biblioteki



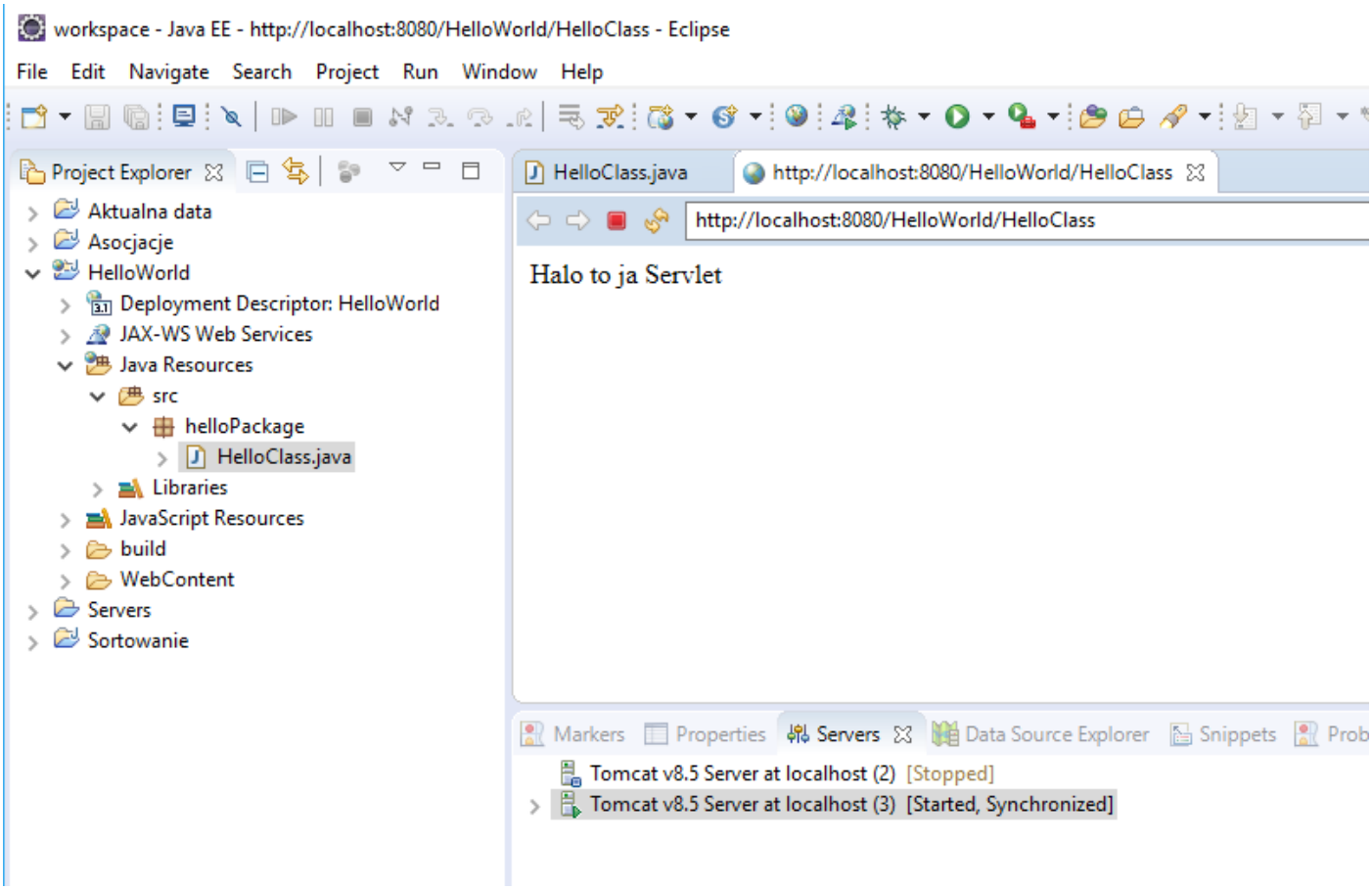
Rysunek 19 Start serwera

Zanim jednak uruchomimy nasza aplikacje musimy wystartować z serwerem Tomcat.

Możemy tego dokonać na dwa sposoby pierwszy (powyżej) klikając PPM w zakładce serwery na nasz Tomcat i następnie start. Drugi klikając prawym na naszym projekcie, następnie „Run As > Run on Server”.



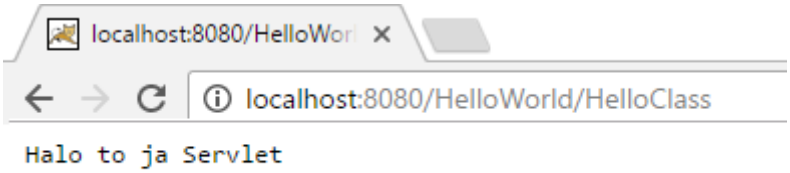
Rysunek 20 Wystartowany serwer



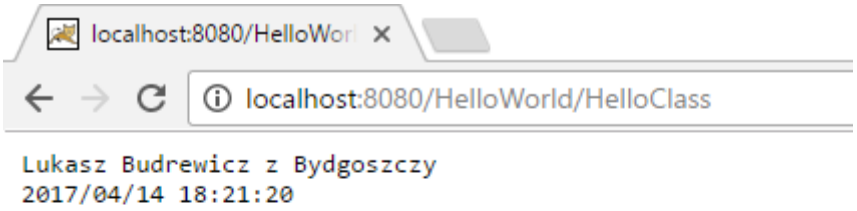
Rysunek 21 Nasza strona

Uzywając drugi sposób automatycznie uruchomi nam się strona w programie Eclipse, dostępna jest ona również z poziomu przeglądarki, pod adresem

- http://localhost:8080/HelloWorld/HelloClass, gdzie HelloClass oznacza naszą trasę routingu wcześniej zdefiniowaną przy zakładaniu klasy.



Rysunek 22 Pierwsza strona



Rysunek 23 Przerobiona pierwsza strona

Zadanie zmiany wyświetlanego tekstu, analogicznie do zadań 1 i 2.

```
protected void doGet(HttpServletRequest request, HttpServletResponse
             response) throws ServletException, IOException {
              PrintWriter writer = response.getWriter();
              writer.append("Lukasz Budrewicz z Bydgoszczy");
              writer.append(System.lineSeparator());
              DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
              Date date = new Date();
              writer.append(dateFormat.format(date));
              writer.flush();
    }

protected void doPost(HttpServletRequest request, HttpServletResponse
             response) throws ServletException, IOException {
             doGet(request, response);
}
```