Struts 2

Zagadnienia Programowania Obiektowego

Agata Hejmej

3.11.2008

Plan prezentacji

Struts 2

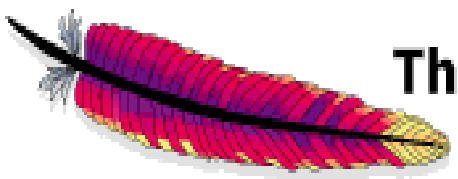
- co to jest Struts2 ?
- krótki przegląd typów podejść przy tworzeniu aplikacji webowych
- architektura MVC i MVC2
- architektura Struts2
- cykl życia żądania w Struts2
- akcje
- interceptory
- podsumowanie
- dlaczego Struts2 ?

Struts 2

Struts 2 to elastyczny i łatwo rozszerzalny framework do tworzenia aplikacji webowych w Javie

- cel Struts 2:
  uczynić rozwijanie aplikacji webowej łatwiejszym dla dewelopera

Struts 2



The Apache Software Foundation
http://www.apache.org/

JAVA i aplikacje webowe

Struts 2

SERWLET – pierwszy oparty na Javie sposób tworzenia aplikacji webowych, program wykonywany na serwerze

* URL jest mapowany na klasy, których metody będą wywoływane
* HTML w Javie na większą skalę jest koszmarem w utrzymywaniu
* przy każdej zmianie w aplikacji konieczna rekompilacja itd.

JSP – Java Server Pages – technologia polegająca na umieszczaniu kodu Javy (scriptlet) wewnątrz kodu HTML

* każdy JSP zapewnia zarówno logikę dla przetwarzania zapytania jak i sposób wyświetlania
* kod Javy nieuporządkowany w klasy i metody sprzyja copy&paste
* dlatego do specyfikacji JSP dodano tagi

JAVA i aplikacje webowe cd.

Action-Based Frameworks – kombinacja pomysłów z serwletów i JSP

* oddzielenie logiki przetwarzania od logiki prezentacji

* MVC, w którym serwlet jest kontrolerem, modelem są **akcje**, a za widok odpowiadają strony JSP

Struts 2 to oparty o model MVC2 Action-Based Framework

Component-Based Frameworks – aplikacja webowa nie jest logicznie dzielona na strony, tylko na komponenty (widgety)

* każdemu widgetowi odpowiada jego własna logika przetwarzania

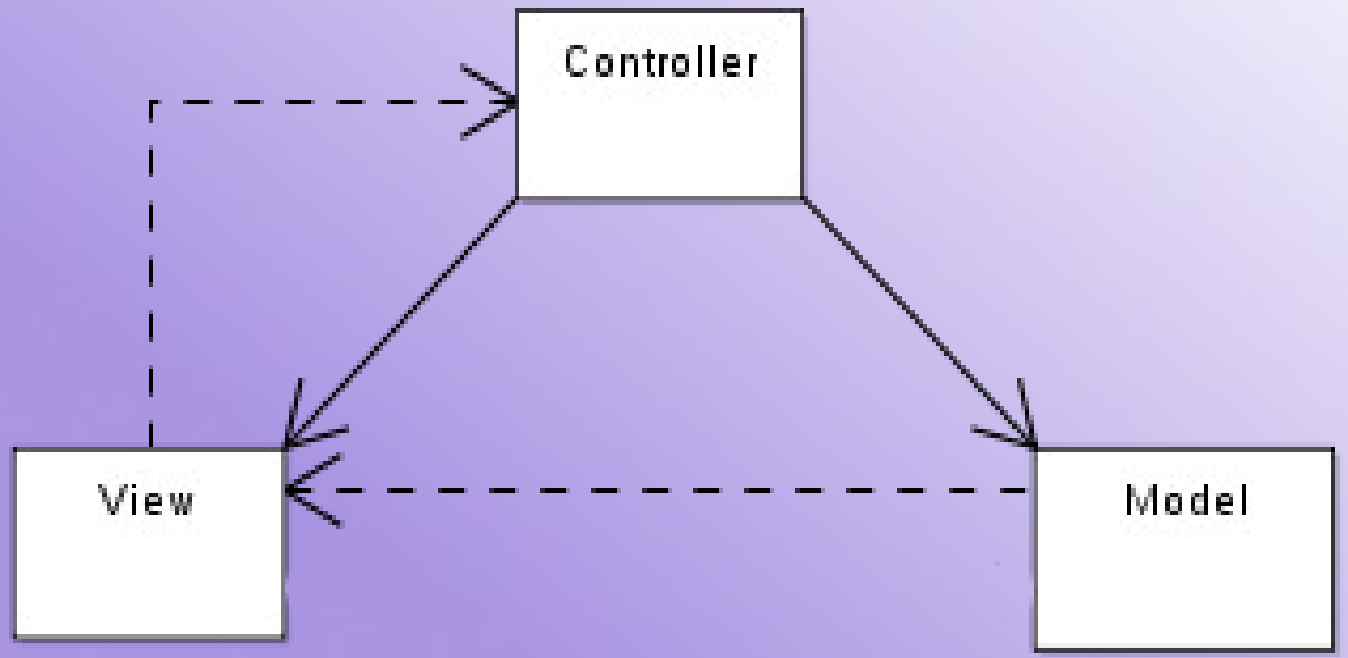
* re-use komponentów

* przykłady: JSF, Wicket, Tapestry

Struts 2

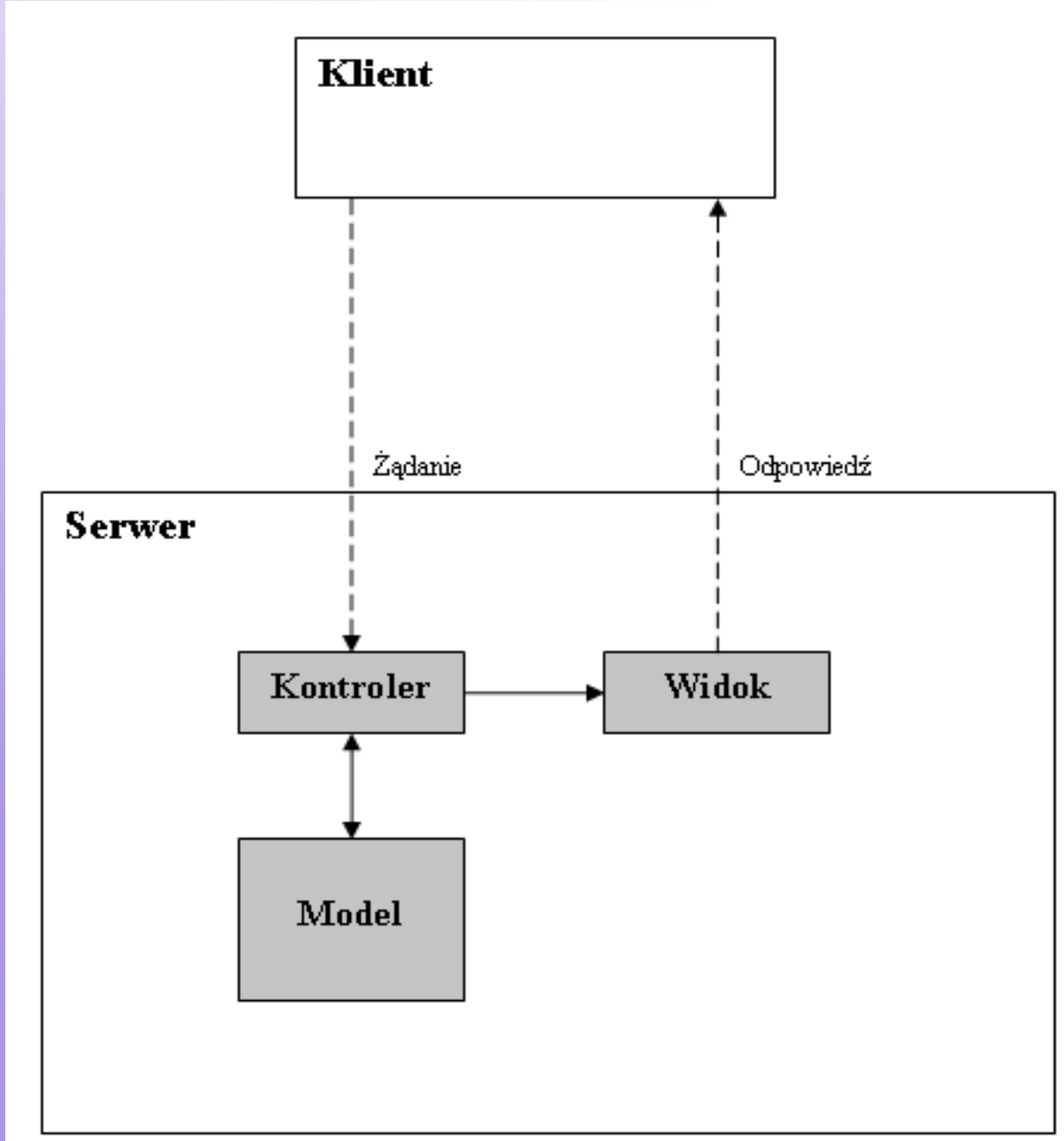
Architektura **MVC** (Model – View – Controller)

- Model – logika biznesowa, baza danych
- Widok – wygląd strony
- Kontroler – kod nawigacyjny

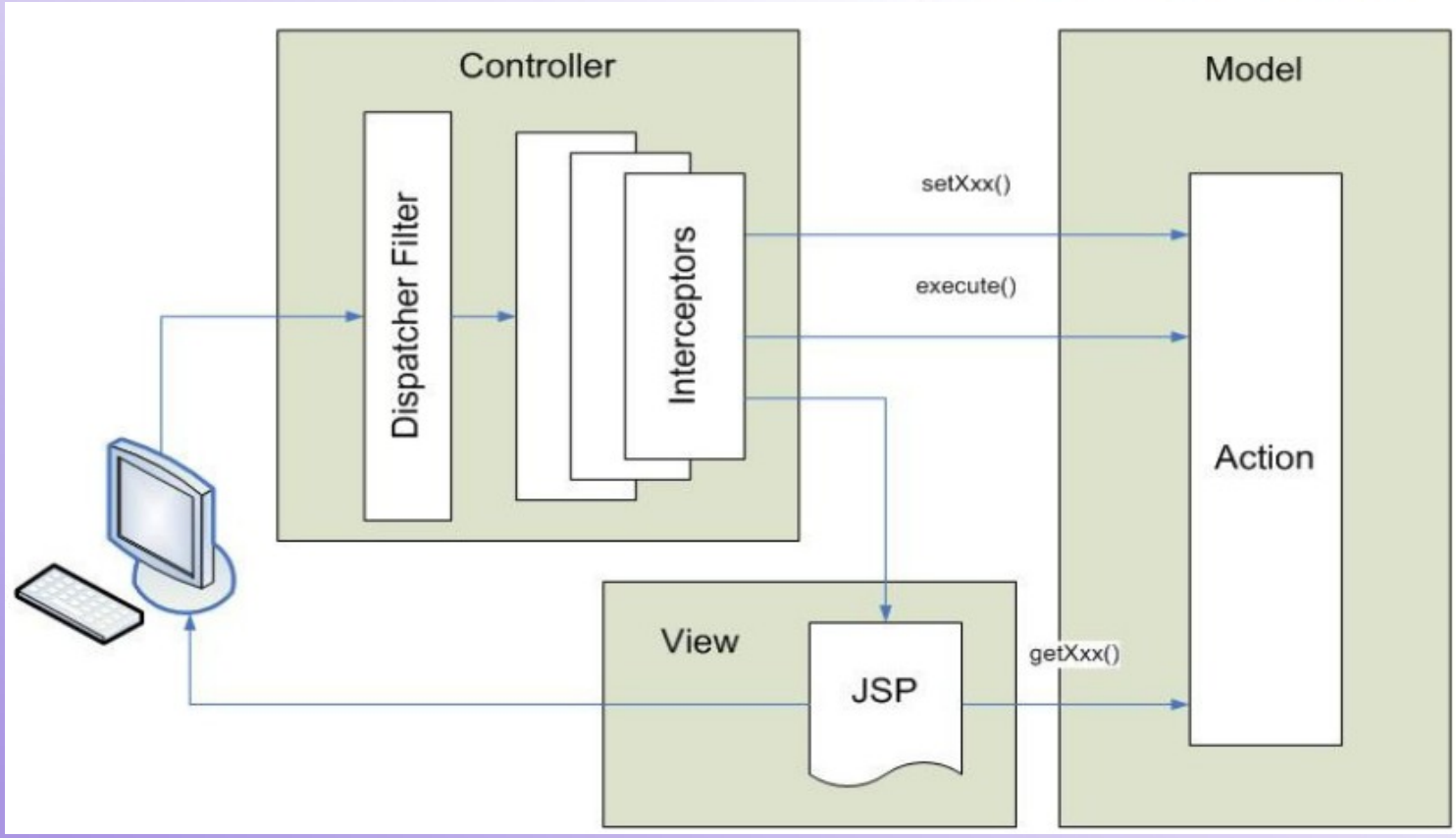


Architektura **MVC2**

- standardowe MVC nie jest najlepszym rozwiązaniem dla aplikacji webowych
- Struts 2 (jak i wiele innych frameworków) używa jego modyfikacji MVC2



Architektura **Struts 2**



- **Kontroler:**
  * Struts2 dispatch servlet filter
  * interceptory
- **Model:**
  * akcje
- **Widok:**
  * typy wynikowe
  * wyniki / view technologies

Struts2

- Struts2 jest bardzo elastyczny i rozszerzalny i dla danej aplikacji webowej właściwie wszystko jest konfigurowalne.

- Framework Struts2 dostarcza warstwę kontrolera.

- Deweloperzy mogą użyć tej warstwy łącząc ją z innymi standardowymi technologiami do obsługi:

  * logiki biznesowej np. POJOs, XWork

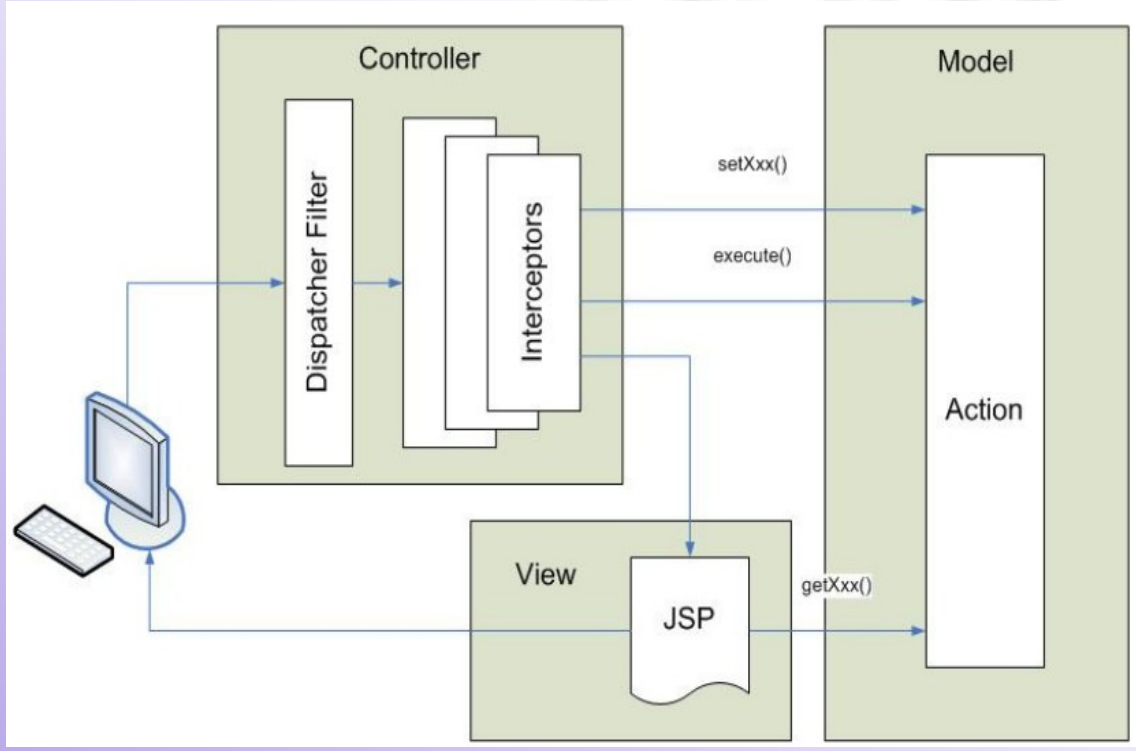
  * dostępu do danych np. DAOs, Cayenne, EJB, Hibernate

  * warstwy prezentacji np. JSP, Velocity, Freemarker

Taka elastyczność jest wielką zaletą, lecz jednocześnie konieczność konfiguracji wszystkiego jest bardzo uciążliwa i czasochłonna

- dlatego zastosowano zasadę Convention over Configuration (ograniczono konfigurację XMLową przez defaulty)
- w dalszej części prezentacji omówię Struts2 w połączeniu z defaultowymi elementami

Struts2 krok po kroku
(cykl życia żądania)



1. Każde żądanie od użytkownika przechodzi najpierw przez Dispatcher Filter, który odpowiednio mapuje URL na akcję, która jest obiektem javowym

2. Interceptory wykonują pre-processing związany z daną akcją

3. Akcja zostaje wykonana

4. Interceptory wykonują post-processing

5. W zależności od wyniku akcji zostaje wygenerowany odpowiedni widok (default: JSP)

Struts 2

Kluczowe we frameworku Struts 2 są:

- akcje
- interceptory

Struts 2

Akcje

AKCJE

Struts 2

- akcje to podstawowy element we frameworkach typu action-based
- są najbardziej podstawową jednostką pracy związaną z żądaniem HTTP pochodzącym od użytkownika
- akcje w Struts 2 to zwykłe POJO (Plain Old Java Object) – nie muszą rozszerzać żadnych klas
- konfiguracje akcji dodajemy do pliku konfiguracyjnego struts.xml
- akcje muszą posiadać metodę która zwraca wartość typu String, domyślnie nazywa się execute()

AKCJE cd.

Struts 2

```
class MyAction {

    public void String execute() throws Exception {
        if( myLogicWorked() ) {
            return "success";
        } else {
            return "error";
        }
    }
}
```

AKCJE cd.

Struts 2

Wpis w pliku konfiguracyjnym:

```
<action name="my" class="com.fdar.infoq.MyAction" >
    <result>view.jsp</result>
    <result name="error" type="freemarker">error.ftl</result>
</action>
```

- atrybut “name” dla action dostarcza URL związany z wykonaniem akcji, w tym przypadku to “/my.action”
- atrybut “class” dostarcza pełną nazwę pakietu i klasy akcji, która ma zostać wykonana
- domyślny atrybut “name” dla result to “success”
- generowane wyniki są domyślnie stronami jsp (type = “dispatcher”) ale są dostępne inne typy, można również tworzyć własne

AKCJE cd.

Struts 2

- akcje, żeby wiedziały w jaki sposób działać, potrzebują dostępu do wartości z żądania oraz formularzy

- jeśli akcja chce dostać się do danych musi dostarczać gettery i/lub settery do danego pola

- np.
  jeśli JSP robi wywołanie:

  “/home.action?framework=struts&version=2”

  akcja powinna dostarczać settery “setFramework(String frameworkName)” oraz “setVersion(int version)”

- resztę, łącznie z konwersją typów dostarcza Struts2

AKCJE cd.

Struts 2

- żeby akcja mogła wykonać jakieś przetwarzanie trzeba udostępnić jej różne obiekty (business objects, data access objects i inne zasoby)
- jednocześnie chcemy żeby w aplikacji były luźne powiązania (loose coupling)
- w tym celu w Struts2 stosuje się technikę zwaną wstrzykiwaniem zależności (dependency injection), a dokładniej setter injection – tzn. że żeby obiekty były dostępne dla akcji, musi ona dostarczać odpowiedni setter

AKCJE cd.

Struts 2

- preferowanym frameworkiem do obsługi dependency injection jest Spring Framework, który jest konfigurowany jako wtyczka
- żeby udostępniać modyfikowane przez akcję obiekty, do akcji trzeba dodać gettery do tych obiektów
- stosowanie wstrzykiwania zależności dodatkowo ułatwia testowanie

Struts 2

Interceptory

INTERCEPTORY

Struts 2

- koncepcyjnie interceptory odpowiadają używanym w serwletach filtrom
- interceptory pozwalają na wstępne oraz wyjściowe przetwarzanie akcji
- wiele możliwości dostarczanych przez Struts2 jest zaimplementowanych jako interceptory
  np. obsługa wyjątków, uploadowanie plików, walidacja
- podobnie jak filtry mogą tworzyć uszeregowane warstwy
- mają dostęp do wykonywanej akcji, zmiennych środowiskowych i własności wywołania

INTERCEPTORY cd.

Struts 2

- W Struts 2 mamy dostępnych wiele interceptorów „out of the box”
- np. interceptory implementujące wstrzykiwanie zależności:
  - * dla Spring Framework – ActionAutowiringInterceptor
  - * żądanie i wartości z formularzy – ParametersInterceptor

INTERCEPTORY cd.

Struts 2

- Wpis konfiguracyjny do struts.xml:

```
<interceptors>
   ...
   <interceptor name="autowiring"
      class="interceptor.ActionAutowiringInterceptor"/>
</interceptors>
```

- Interceptory dodajemy do akcji:

```
<action name="my" class="com.fdar.infoq.MyAction" >
   <result>view.jsp</result>
   <interceptor-ref name="autowiring"/>
</action>
```

- Lub do wszystkich akcji w pakiecie:

```
<default-interceptor-ref name="autowiring"/>
```

INTERCEPTORY cd.

- Struts2 opiera większość swojej funkcjonalności na interceptorach, więc często do jednej akcji przypisanych jest po ok. 8-10 interceptorów

- Dlatego wygodnie jest tworzyć stosy interceptorów

```
<interceptor-stack name="basicStack">
    <interceptor-ref name="exception"/>
    <interceptor-ref name="servlet-config"/>
    <interceptor-ref name="prepare"/>
    <interceptor-ref name="checkbox"/>
    <interceptor-ref name="params"/>
    <interceptor-ref name="conversionError"/>
</interceptor-stack>
```

```
<action name="my" class="com.fdar.infoq.MyAction" >
    <result>view.jsp</result>
    <interceptor-ref name="basicStack"/>
</action>
```

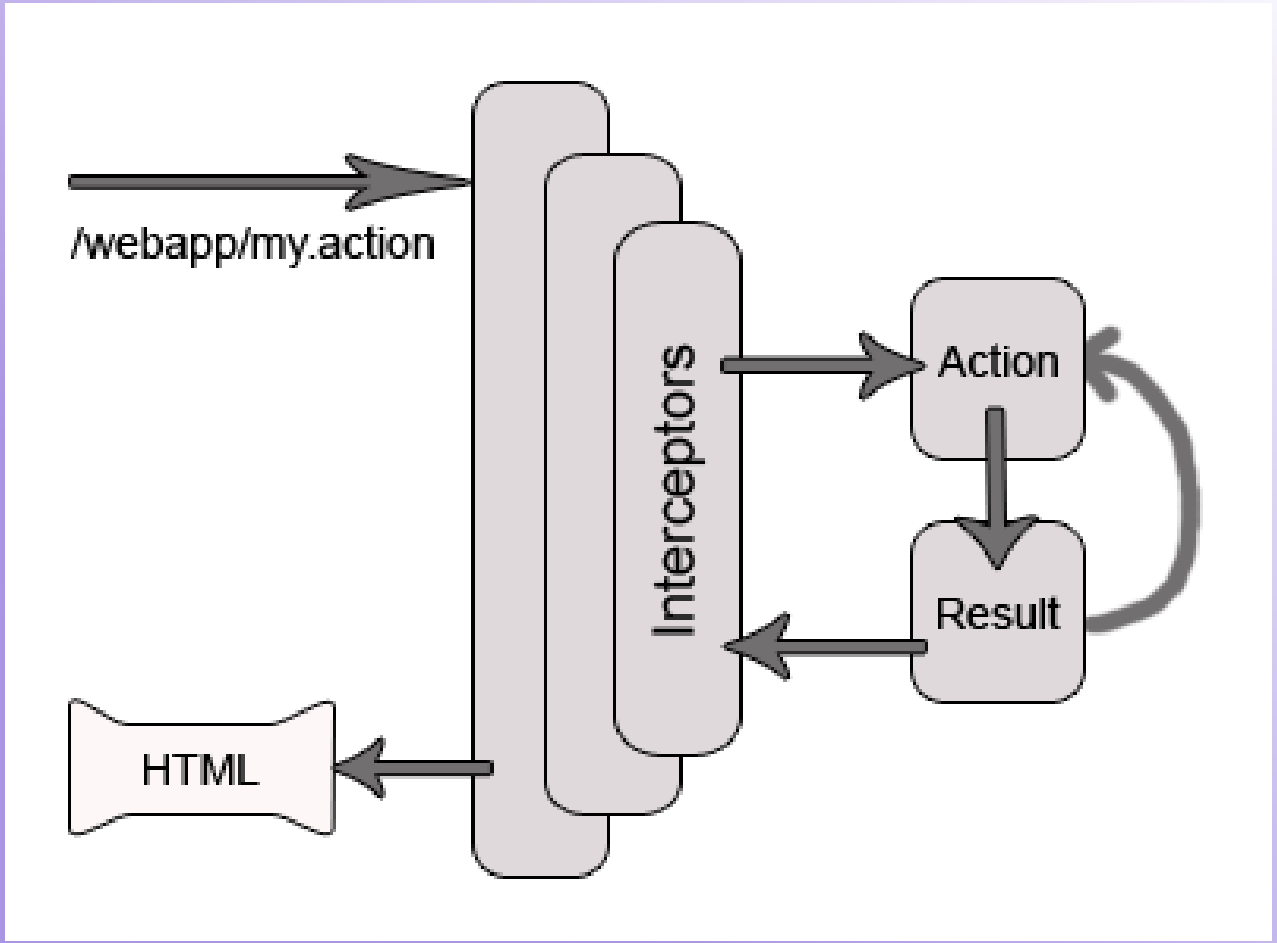
```
public interface Interceptor extends Serializable {

    void destroy();

    void init();

    String intercept(ActionInvocation invocation)
        throws Exception;
}
```

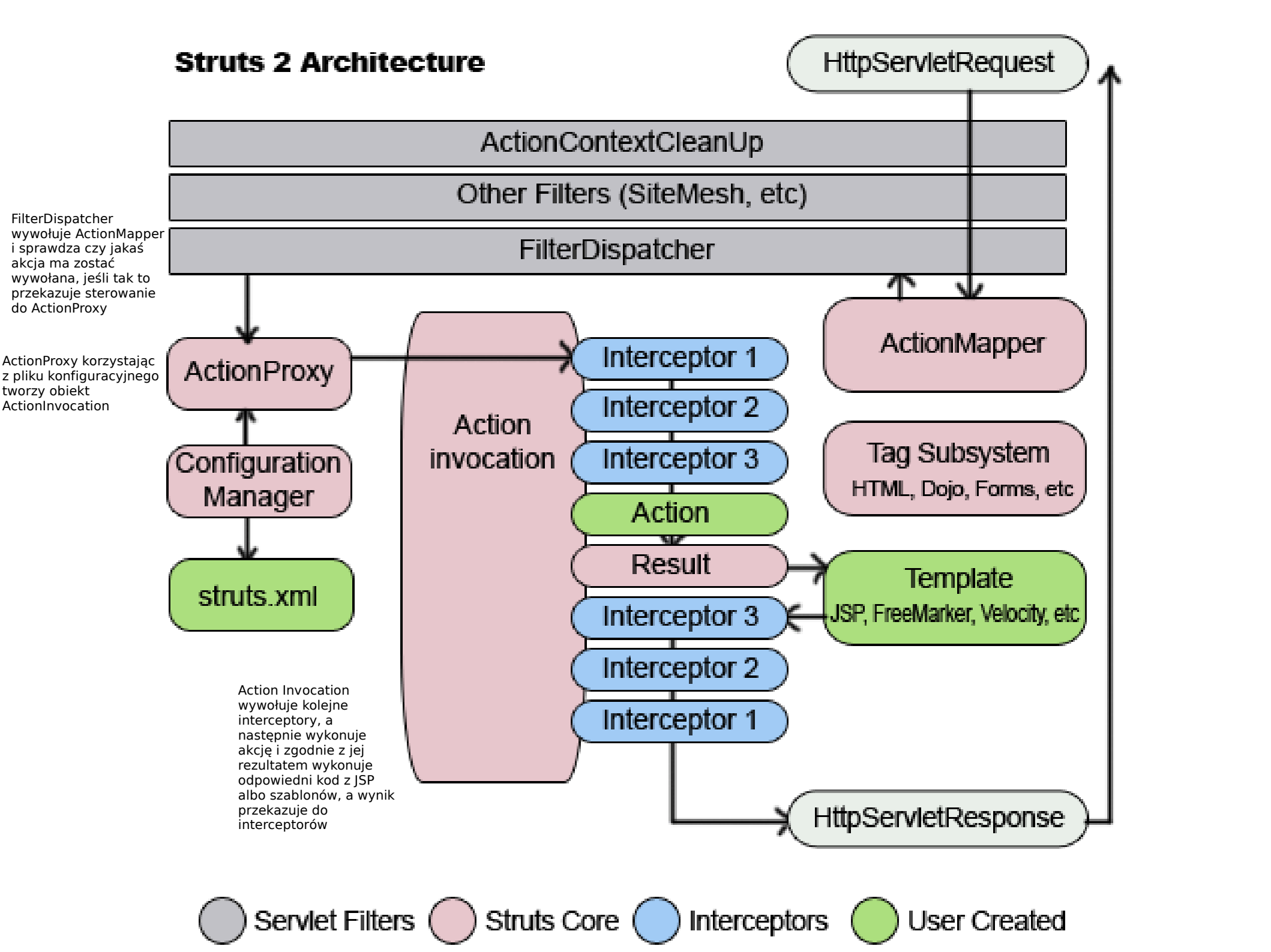
- obiekt ActionInvocation zapewnia dostęp do środowiska uruchomieniowego, co pozwala na:

  * dostęp do wywoływanej akcji
  * kontekstu (parametrów żądania, sesji itd.)
  * wyniku wykonania akcji

Podsumowanie

Struts 2




Struts 2 Architecture
HttpServletRequest
ActionContextCleanUp
Other Filters (SiteMesh, etc)
FilterDispatcher
FilterDispatcher wywołuje ActionMapper i sprawdza czy jakaś akcja ma zostać wywołana, jeśli tak to przekazuje sterowanie do ActionProxy
ActionMapper
ActionProxy
ActionProxy korzystając z pliku konfiguracyjnego tworzy obiekt ActionInvocation
Action invocation
Interceptor 1
Interceptor 2
Interceptor 3
Action
Result
Interceptor 3
Interceptor 2
Interceptor 1
Configuration Manager
struts.xml
Tag Subsystem
HTML, Dojo, Forms, etc
Template
JSP, FreeMarker, Velocity, etc
Action Invocation wywołuje kolejne interceptory, a następnie wykonuje akcję i zgodnie z jej rezultatem wykonuje odpowiedni kod z JSP albo szablonów, a wynik przekazuje do interceptorów
HttpServletResponse
Servlet Filters
Struts Core
Interceptors
User Created

Dlaczego Struts2 ?

Struts 2

- **uproszczony projekt frameworku** – uniezależnienie akcji od frameworku; komponenty frameworku są luźno powiązane
- **uproszczone akcje** – POJOs, dependency injection
- **ułatwione testowanie** – akcje są niezależne od HTTP i od frameworku + dependency injection
- **nie używa ActionForms**
- **inteligentne defaulty** – większość elementów konfiguracji ma wartości domyślne

Dlaczego Struts2 ? cd.

Struts 2

- **dostarcza biblioteki tagów** i pozwala na wykorzystywanie innych bibliotek (np. JSP, FreeMarker) – mniej kodu
- **wprowadzono adnotacje**
- **QuickStart** – wiele zmian może być przeprowadzonych w locie, bez konieczności restartu kontenera webowego
- **łatwa integracja ze Springiem**
- **łatwa rozszerzalność poprzez dodawanie wtyczek** – wystarczy dodać JARa
- **wsparcie dla AJAXa** – framework dostarcza zestaw tagów, które uczynią aplikację bardziej interaktywną

Struts 2

Dziękuję za uwagę

KONIEC