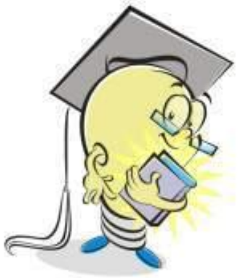
HandlerInterceptor

Cognizant

Passion for building stronger businesses



C3: Protected

About the Author

Created By:	Ramesh C.P.(161646)
Credential Information:	SCJP, 8+ years of experience in Technical training
Version and Date:	Spring MVC/PPT/1011/3.0

Cognizant Certified Official Curriculum

UNDER THE SEAL OF COGNIZANT ACADEMY

Icons Used

Questions

Tools

Hands on Exercise

Coding Standards

Test Your Understanding

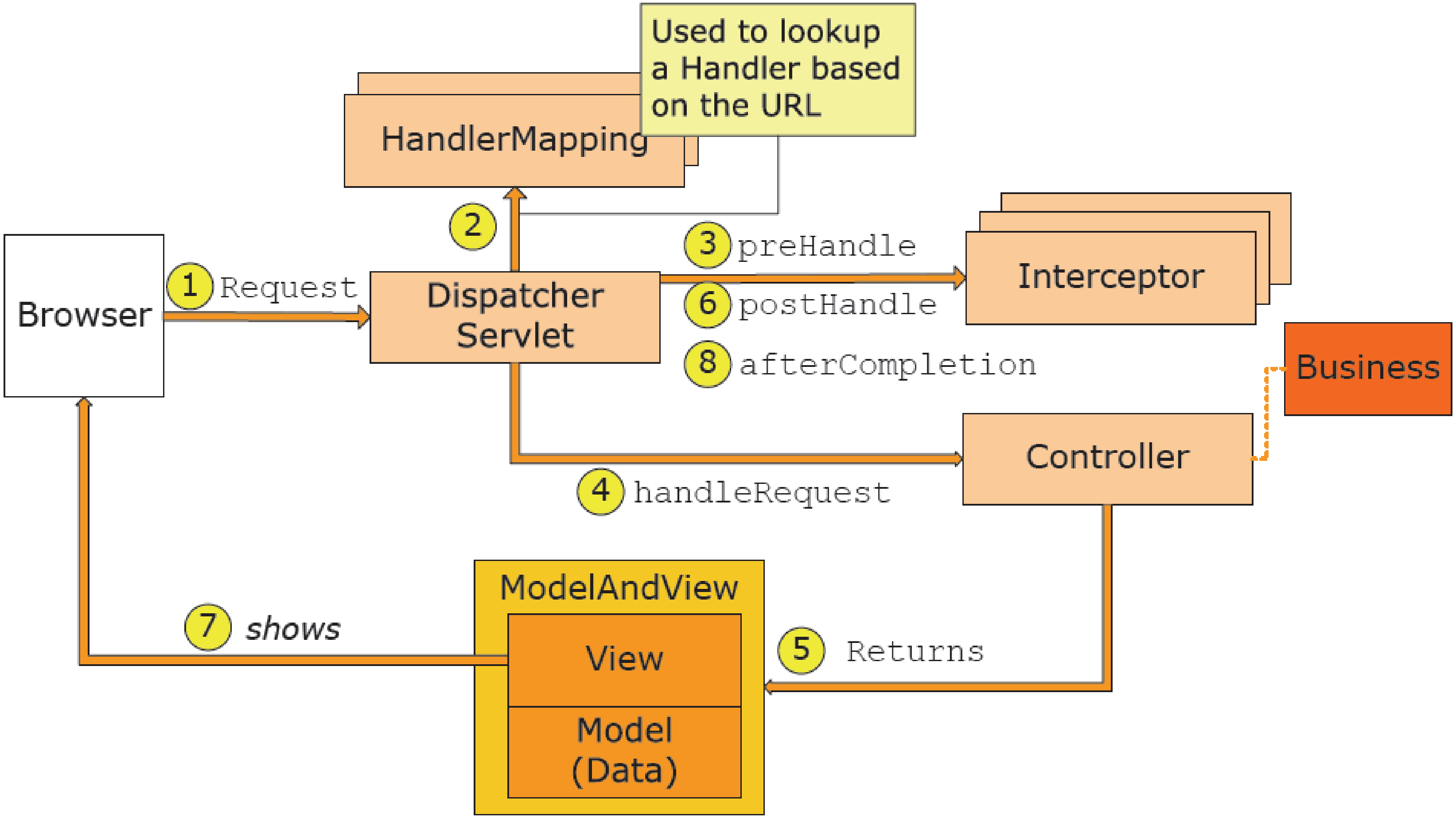
Reference

Demonstration

A Welcome Break

Contacts

Review of Architecture



HandlerInterceptor: Overview

- Introduction:
  - Spring's handler mapping mechanism has the notion of handler interceptors, that can be extremely useful when you want to apply specific functionality to certain requests, for example, checking for a principal.
  - Applications can register any number of existing or custom interceptors for certain groups of handlers, to add common preprocessing behavior without needing to modify each handler implementation.
  - A HandlerInterceptor gets called before the appropriate HandlerAdaptor triggers the execution of the handler itself

HandlerInterceptor: Objectives

❖ Objective:

After completing this chapter you will be able to:

- Write customized Handler Interceptor.
- Use Spring supplied implementation of Handler Interceptor.

HandlerInterceptor interface

- Interceptor Life Cycle Points

Method Name	Description
preHandle	Called before the request handler is invoked. If it returns false, the request handler is never invoked, and the rest of the interceptor's methods are never called
postHandle	Called after the handler finishes but before the view is rendered. Useful for placing common objects into the model
afterCompletion	Called after the view is rendered, even if there was an error in handling the request. Useful for resource cleanup.

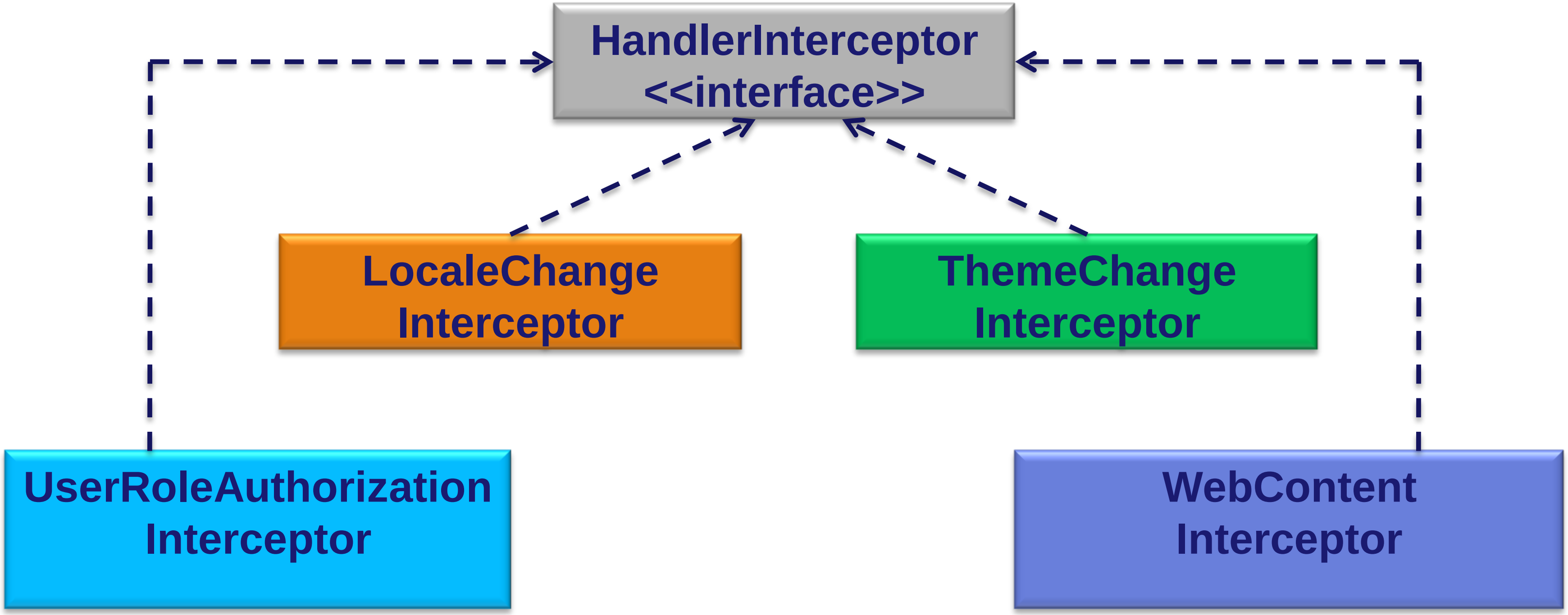
HandlerInterceptor – A Sample

```
public  class DateInsertionInterceptor implements HandlerInterceptor
{
   public boolean preHandle(HttpServletRequest arg0, HttpServletResponse arg1, Object arg2)
                                          throws Exception {
      return true;
   }
   public void postHandle(HttpServletRequest arg0, HttpServletResponse arg1, Object arg2,
                                     ModelAndView arg3) throws Exception {
     arg3.addObject("currentTime",new Date()); // insert date into the model
}
   public void afterCompletion(HttpServletRequest arg0, HttpServletResponse arg1,
                                                Object arg2, Exception arg3) throws Exception {
      //do nothing
}
}
```

HandlerInterceptor – A Sample (Contd.)

```
<bean id="urlMapping"
      class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping" >
   <property name="order" value="1" />
   <property name="mappings">
      <props>
            <prop key="/login.htm">loginForm</prop>
            <prop key="/throw.htm">throwawayController</prop>
      </props>
   </property>
   <property name="interceptors">
      <list>
            <bean class="DateInsertionInterceptor"/>
      </list>
  </property>
</bean>
```

Spring Implementations



Spring Implementations (Contd.)

- The `Statistics` implementation

```
public class Statistics extends HandlerInterceptorAdapter {
  // argument types mentioned earlier (request, response, handler)
  public boolean preHandle(...) throws Exception {
    ...
  }
}
```

- Bundled interceptors

`WebContentInterceptor`	Checks for allowed request methods, makes sure valid session is available, and sets HTTP headers for caching
`UserRoleAuthorizationInterceptor`	Wraps the request with a `isUserInRole` call. The allowed roles are provided using the `authorizedRoles` property.
`HandlerInterceptorAdapter`	Convenient base class to implement a `HandlerInterceptor`. You only override the methods needed.

LocaleChangeInterceptor

- ❖ Interceptor that allows for changing the current locale on every request, via a configurable request parameter.

```
<bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping"> <property
    name="interceptors">
     <list>
       <bean
           class="org.springframework.web.servlet.LocaleChangeInterceptor">
          <property name="paramName">
                         <value>language</value>
           </property>
        </bean>
     </list>
</property>
.... </bean>
```

For the request http://www.springdemo.cts.com/sample/index.view?language=en” the locale will be changed to English

UserRoleAuthorizationInterceptor

- Authorization interceptor that checks a user's Principal with a set of roles you specify

```
<bean id="userRoleAuthorizationInterceptor"
    class="org.springframework.web.servlet.handler.UserRoleAuthorizationInterceptor">
   <property name="authorizedRoles">
           <list>
                   <value>administrator</value>
                    <value>editor</value>
           </list>
    </property>
 </bean>
```

Important: The UsersRoleAuthorizationInterceptor can be used in combination with the JAAS features of an application server. Generally, the UserRoleAuthorization Interceptor suffices in simple situations only.

WebContentInterceptor

- Provides functionality for easy modification of caching behavior of a web application and useful generic tasks such as limiting the methods (GET, POST, PUT) that are supported by the web application.

```
<bean id="contentInterceptor"
    class="org.springframework.web.servlet.mvc.WebContentInterceptor">
    <property name="cacheSeconds"><value>30</value></property>
     <property name="supportedMethods"><value>GET</value></property>
 </bean>
```

Other properties

- cacheSeconds
- supportedMethods
- requireSession

HandlerInterceptor

Time for a Break !

HandlerInterceptor

- Questions from participants

Test Your Understanding

- Handler interceptor is more powerful than filters. Say true or false
- Handler interceptor can be used to authorize user to a resource. Say True or false
- A controller can have n number of interceptors. S true or false.

HandlerInterceptor: Summary

- HandlerInterceptors are an excellent opportunity to apply common business logic to many Controllers.
- They act much like filters, in that they intercept the request handling pipeline.
- They are capable of bypassing the request handling altogether, placing common objects into the model, and cleaning up resources after every request.

HandlerInterceptor: Source

- http://book.javanb.com/Professional-Java-Development-with-the-Spring-Framework/BBL0100.html
- http://book.javanb.com/Professional-Java-Development-with-the-Spring-Framework/BBL0103.html

Disclaimer: Parts of the content of this course is based on the materials available from the Web sites and books listed above. The materials that can be accessed from linked sites are not maintained by Cognizant Academy and we are not responsible for the contents thereof. All trademarks, service marks, and trade names in this course are the marks of the respective owner(s).

You have successfully completed HandlerInterceptor

Click here to proceed

Cognizant™
Passion for building stronger businesses