

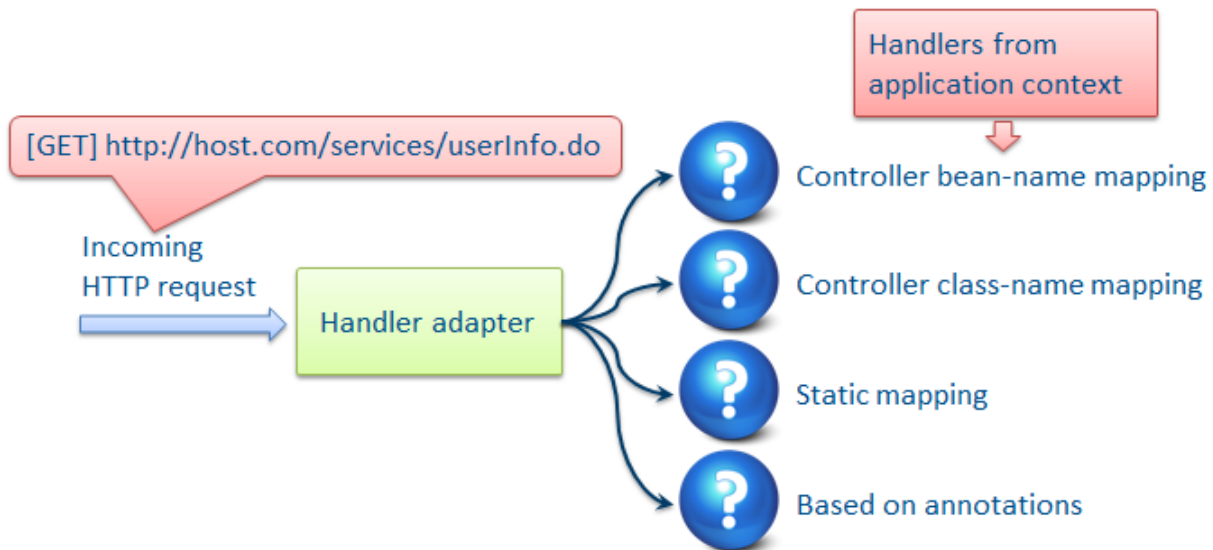


SPRING MVC

HANDLER MAPPING

MARCH, 2015

HANDLER MAPPING



HANDLER MAPPING EXAMPLE

```
<beans>
  <!-- Map based on class name -->
  <bean class="org.springframework...ControllerClassNameHandlerMapping"/>

  <!-- Static mapping -->
  <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
      <value>
        /calc.do=calculationController
      </value>
    </property>
  </bean>

  <bean id="calculationController"
    class="com.epam.example.mvc.CalculationController"/>
</beans>
```

HANDLER MAPPING INTERFACE

```
public interface HandlerMapping {  
    HandlerExecutionChain getHandler(HttpServletRequest request) throws Exception;  
}
```

```
<beans>  
    <bean id="handlerMapping"  
class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping">  
        <property name="interceptors">  
            <bean class="com.epam.example.MyInterceptor"/>  
        </property>  
    </bean>  
</beans>
```

MVC NAMESPACE CONFIGURATION

```
<beans>
  <!-- Support for @Autowire -->
  <context:annotation-config />

  <!-- Support for @Controller -->
  <context:component-scan base-package="com.epam.example.mvc" />

  <!-- Turn @Controller into actual web controllers-->
  <mvc:annotation-driven/>

  <mvc:interceptors>
    <bean class="com.epam.example.interceptors.SecurityInterceptor"/>
  </mvc:interceptors>

  <mvc:resources mapping="/static/**" location="/pages/" />

  <mvc:view-controller path="/" view-name="homePage"/>
</beans>
```

MVC INTERCEPTOR IMPLEMENTATION

```
public class MyInterceptor implements HandlerInterceptor {  
    @Override  
    public boolean preHandle(HttpServletRequest request,  
        HttpServletResponse response, Object handler) throws Exception {  
  
        System.out.println("Pre-handle");  
  
        return true;  
    }  
    //override postHandle() and afterCompletion()  
}
```

BASIC CONTROLLER MAPPING

@Controller

```
public class CalcController {
```

```
    // [GET] http://host.com/example/calculate?first=123&second=456
```

```
    @RequestMapping("/calculate")
```

```
    public ModelAndView calculate(HttpServletRequest request) {
```

```
        String first = request.getParameter("first");
```

```
        String second = request.getParameter("second");
```

```
        int firstInt = Integer.parseInt(first);
```

```
        int secondInt = Integer.parseInt(second);
```

```
        ModelAndView mav = new ModelAndView("displaySum");
```

```
        mav.addObject("sum", Integer.toString(firstInt + secondInt))
```

```
        return mav;
```

```
    }
```

```
}
```

MAPPING APPROACHES

- By path
`@RequestMapping("/welcome")`
- By HTTP method
`@RequestMapping(value = "/welcome", method=RequestMethod.GET)`
- By presence / value of query parameter
`@RequestMapping(params = {"find=ByMake", "form" })`
- By presence / value of request header
`@RequestMapping(value = "/welcome", headers="accept=text/*")`



THANK YOU!

MAKSYM_GOVORISCHEV@EPAM.COM

MARCH, 2015