Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Spring MVC

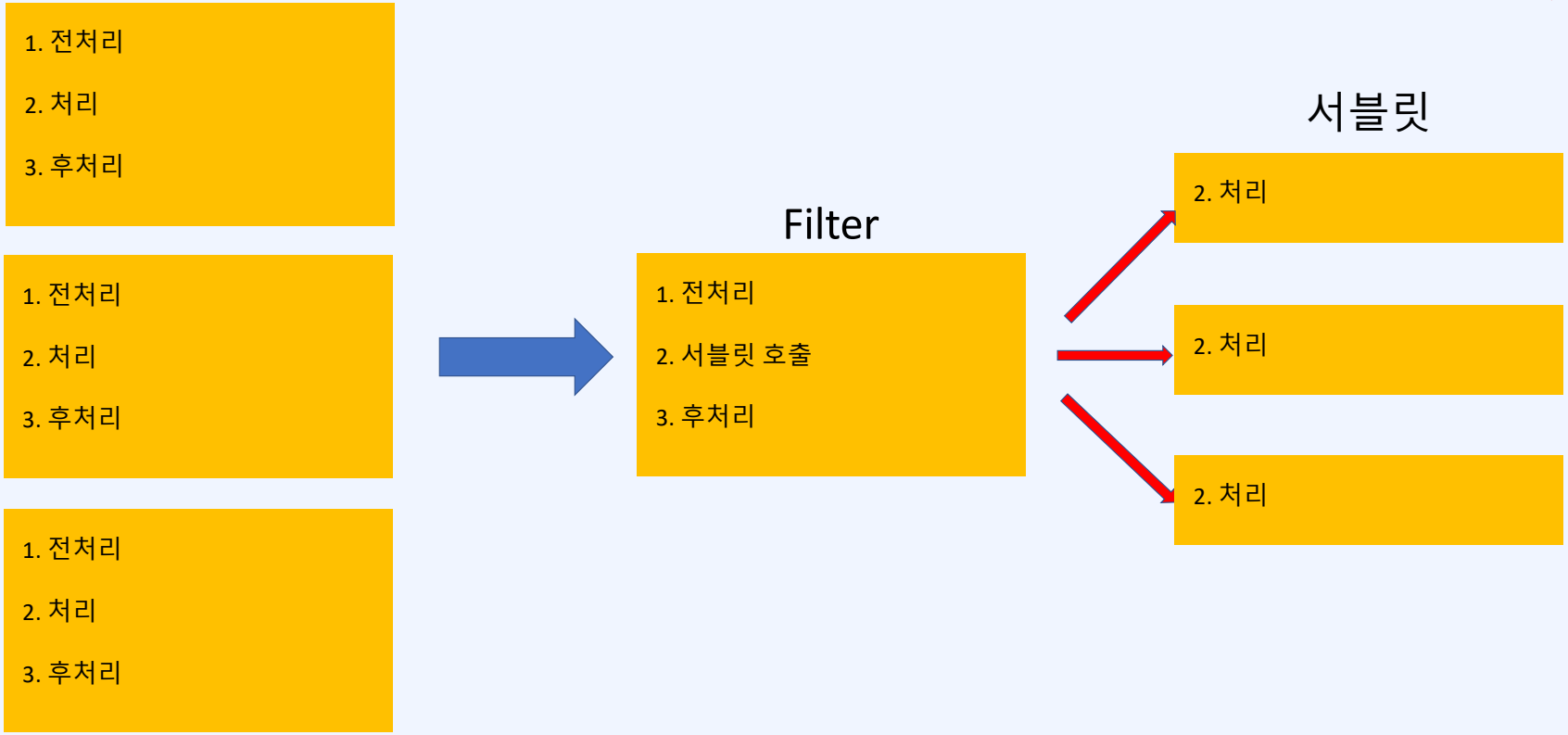
13 Filter와 Interceptor

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Filter와 Interceptor

1. Filter

공통적인 요청 전처리와 응답 후처리에 사용. 로깅, 인코딩 등

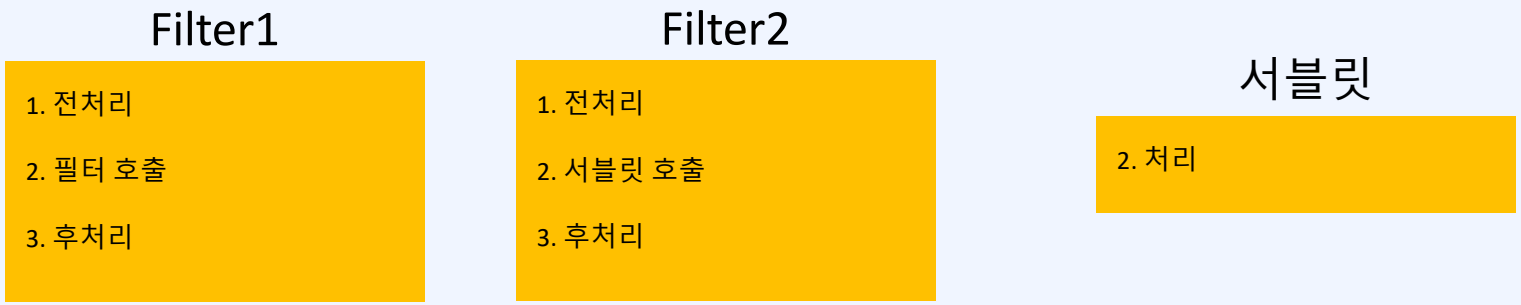


Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Filter와 Interceptor

1. Filter

공통적인 요청 전처리와 응답 후처리에 사용. 로깅, 인코딩 등



Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Filter와 Interceptor

1. Filter

```java
@WebServlet("/rollDice2")
public class TwoDiceServlet extends HttpServlet {
    public void service(HttpServlet
        // 1. 전처리 작업
        long startTime = System.curr

        int idx1 = (int)(Math.rando
        int idx2 = (int)(Math.rando

        response.setContentType("te
        response.setCharacterEncodi
        PrintWriter out = response.
        out.println("<html>");
        out.println("<head>");
        out.println("</head>");
        // ...

        // 3. 후처리 작업
        System.out.print(" [" + ((H
        System.out.println(" time="
    }
```

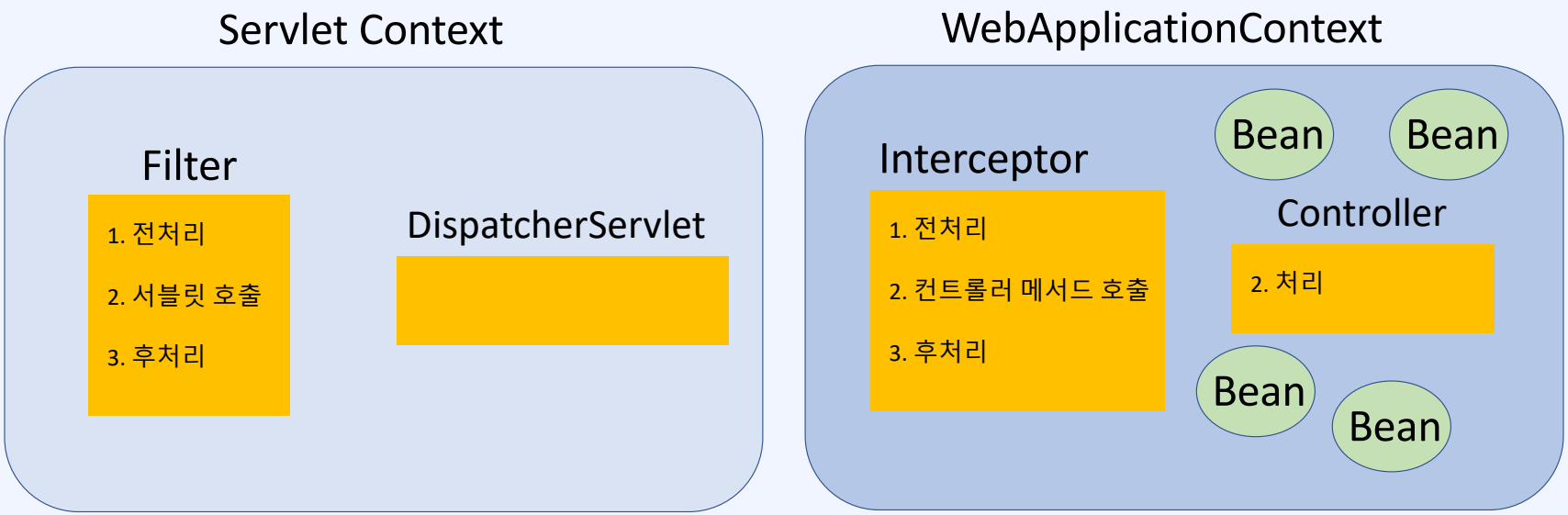
```java
@WebFilter(urlPatterns="/*")  // 모든 요청에 필터를 적용.
public class PerformanceFilter implements Filter {
    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
            throws IOException, ServletException {
        // 1. 전처리 작업
        long startTime = System.currentTimeMillis();

        // 2. 서블릿 또는 다음 필터를 호출
        filterChain.doFilter(request, response);

        // 3. 후처리 작업
        System.out.print("["+((HttpServletRequest)request).getRequestURI()
        System.out.println("time="+(System.currentTimeMillis() - startTime
    }
}
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Filter와 Interceptor

2. Interceptor

Filter와 유사한 기능. Filter와 달리 WAC내에 위치. 빈 주입 가능



Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Filter와 Interceptor

3. Filter와 Interceptor의 비교

2.

```
@WebFilter(urlPatterns=
public class Performanc
    @Override
    public void doFilte
            throws IOEx
        // 1. 전처리 작업
        long startTime =

        // 2. 서블릿 또는 D
        filterChain.doF

        // 3. 후처리 작업
        System.out.print
        System.out.print
    }
}
```

```
public class PerformanceInterceptor implements HandlerInterceptor {
    long startTime;

    // 1. 전처리 작업
    public boolean preHandle(HttpServletRequest request,
                             HttpServletResponse response, Object handler) throws Exception {
        startTime = System.currentTimeMillis();

        // handler는 호출된 메서드
        HandlerMethod method = (HandlerMethod) handler;
        System.out.println("Bean:"+ method.getBean());
        System.out.println("Method:" + method.getMethod());

        return true;   // false면 다음 인터셉터나 컨트롤러를 호출하지 않는다.
    }

    // 3. 후처리  작업
    public void postHandle(HttpServletRequest request, HttpServletResponse response,
                           Object handler, ModelAndView modelAndView) throws Exception {
        System.out.print("["+((HttpServletRequest)request).getRequestURI()+"]");
        System.out.println(" time222="+(System.currentTimeMillis() - startTime)+"ms");
    }
}
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Filter와 Interceptor
4. Interceptor의 등록

```java
@Configuration
public class WebMvcConfig implements WebMvcConfigurer {
    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(new PerformanceInterceptor())
                .exclud
    }
}
```

```java
@Configuration
public class WebMvcConfig implements WebMvcConfigurer {
    @Bean
    PerformanceInterceptor performanceInterceptor() {
        return new PerformanceInterceptor();
    }

    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(performanceInterceptor())
                .excludePathPatterns("/css/**", "/images/**", "/js/**");
    }
}
```