

Spring MVC Request Flow Documentation

1) End-to-End Request → Response (Baseline)

Big picture flow without filters or interceptors:

- Client → Tomcat Connector (Coyote) → Servlet mapping → DispatcherServlet
- DispatcherServlet → Spring MVC pipeline → Response written → Tomcat → Client

Detailed steps:

- Tomcat accepts socket, parses HTTP request
- DispatcherServlet.service(...) → FrameworkServlet.doGet/doPost → DispatcherServlet.doDispatch
- HandlerMapping resolves handler → HandlerExecutionChain
- HandlerAdapter invokes handler, resolves arguments
- Controller executes and returns response (String, ModelAndView, ResponseEntity, etc.)
- Return handled via HttpMessageConverters or ViewResolver
- HandlerExceptionResolvers handle exceptions
- Response committed and flushed to client

2) With Filters (Servlet API)

Filters run before DispatcherServlet and can short-circuit the request.

- Filter implements `javax.servlet.Filter#doFilter`
- Can wrap requests/responses or stop chain.`doFilter()`
- Typical filters: `CharacterEncodingFilter`, CORS filter, `SpringSecurityFilterChain`
- Ordering is important and can be controlled with annotations/config

3) With Interceptors (Spring MVC)

Interceptors run inside the DispatcherServlet, wrapping handler execution.

- `preHandle`: runs before controller, can block processing
- `postHandle`: runs after controller, before view rendering
- `afterCompletion`: runs after view rendering, like finally block
- `AsyncHandlerInterceptor` supports async request flow

Filters vs Interceptors (Comparison)

- Filters: Managed by servlet container (Tomcat), run before servlet

- Interceptors: Managed by Spring, run inside DispatcherServlet
- Filters see raw requests, Interceptors see handler info
- Filters apply broadly, Interceptors apply only to Spring-mapped requests

Complete Flow with Both Filters and Interceptors

- Client → Tomcat → Filters → DispatcherServlet
- DispatcherServlet → HandlerMapping → HandlerExecutionChain with interceptors
- `Interceptor.preHandle` → Controller → Return handling → `Interceptor.postHandle` → Response
- `Interceptor.afterCompletion` → Filters (reverse order) → Tomcat → Client