

Spring RequestMapping Supported Method Return Types

- A `ModelAndView` object, with the model implicitly enriched with command objects and the results of `@ModelAttribute` annotated reference data accessor methods.
- A `Model` object, with the view name implicitly determined through a `RequestToViewNameTranslator` and the model implicitly enriched with command objects and the results of `@ModelAttribute` annotated reference data accessor methods.
- A `Map` object for exposing a model, with the view name implicitly determined through a `RequestToViewNameTranslator` and the model implicitly enriched with command objects and the results of `@ModelAttribute` annotated reference data accessor methods.
- A `View` object, with the model implicitly determined through command objects and `@ModelAttribute` annotated reference data accessor methods. The handler method may also programmatically enrich the model by declaring a `Model` argument (see above).
- A `String` value that is interpreted as the logical view name, with the model implicitly determined through command objects and `@ModelAttribute` annotated reference data accessor methods. The handler method may also programmatically enrich the model by declaring a `Model` argument (see above).
- `void` if the method handles the response itself (by writing the response content directly, declaring an argument of type `ServletResponse` / `HttpServletResponse` for that purpose) or if the view name is supposed to be implicitly determined through a `RequestToViewNameTranslator` (not declaring a response argument in the handler method signature).
- If the method is annotated with `@ResponseBody`, the `return` type is written to the response **HTTP** body. The `return` value will be converted to the declared method argument type using `HttpMessageConverter`s. See [the section called "Mapping the response body with the @ResponseBody annotation"](http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html#mvc-ann-responsebody) (<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html#mvc-ann-responsebody>).
- An `HttpEntity<?>` or `ResponseEntity<?>` object to provide access to the Servlet response **HTTP** headers and contents. The entity body will be converted to the response stream using `HttpMessageConverter`s. See [the section called "Using HttpEntity"](http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html#mvc-ann-httpentity) (<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html#mvc-ann-httpentity>).
- An `HttpHeaders` object to `return` a response with no body.
- A `Callable<?>` can be returned when the application wants to produce the `return` value asynchronously in a thread managed by Spring **MVC**.
- A `DeferredResult<?>` can be returned when the application wants to produce the `return` value from a thread of its own choosing.
- Any other `return` type is considered to be a single model attribute to be exposed to the view, using the attribute name specified through `@ModelAttribute` at the method level (or the `default` attribute name based on the `return` type `class` name). The model is implicitly enriched with command objects and the results of `@ModelAttribute` annotated reference data accessor methods.