Spring Professional Exam Tutorial v5.0

Question 10

Question 10 - What other annotations might you use on a controller method parameter?

- `@RequestParam` - access to the Servlet request parameters, including multipart files, parameters will be automatically converted to declared method argument types, parameters can be made optional with usage of `required` attribute or `Optional` from Java 8, for optional request parameters `defaultValue` can be set as well
- `@PathVariable` - access to URI template variables, parameters can be made optional with usage of `required` attribute or `Optional` from Java 8
- `@MatrixVariable` - access to name-value pairs in URI path segments as described in RFC 3986, allows mapping variables from requests like `/employees/id=1;name=John`
- `@CookieValue` - bind the value of an HTTP cookie to a method argument in a controller, you can bind against simple types or `Cookie` class, cookie can be set with usage of `HttpServletResponse`, cookie can be set as required or optional via `required` attribute or with `Optional` from Java 8, when using `required` attribute, `defaultValue` can be used as well
- `@RequestHeader` - access request header values or all header key and values when binding against a Map

Question 10 - What other annotations might you use on a controller method parameter?

- `@RequestBody` - allows access to HTTP request body, content will be converted to method controller type by `HttpMessageConverter`, request body can be made optional with usage of `required` attribute or Java 8 `Optional`, can be used with `@Valid` for bean validation
- `@RequestPart` - allows to bind multipart HTTP requests to method parameter, content will be converted to method controller type, request part can be made optional with usage of `required` attribute or Java 8 `Optional`, can be used with `@Valid` for bean validation
- `@RequestAttribute` - allows access to HTTP request attributes populated on server-side during HTTP request by filter or interceptor, can be made optional with usage of `required` attribute or Java 8 `Optional`
- `@ModelAttribute` - access to an existing attribute in the model (instantiated if not present) with data binding and validation applied
- `@SessionAttribute` - access to pre-existing session attributes that are managed globally, can be made optional with usage of `required` attribute or Java 8 `Optional`
- `@SessionAttributes` - used to store model attributes in the HTTP Servlet session between requests, useful for multi step from processing