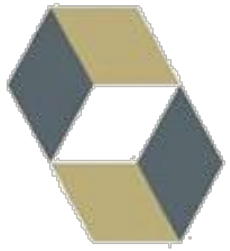


HIBERNATE

Trainer: Nilesh Ghule

Wake up from Hibernate, Spring up!!!

Agenda

- Session and Request scoped beans
- File upload and download
- @RequestMapping
- Internationalization / Localization
- Validation
- Spring exception handling
- Static resources in Web MVC
- MVC using Annotation config

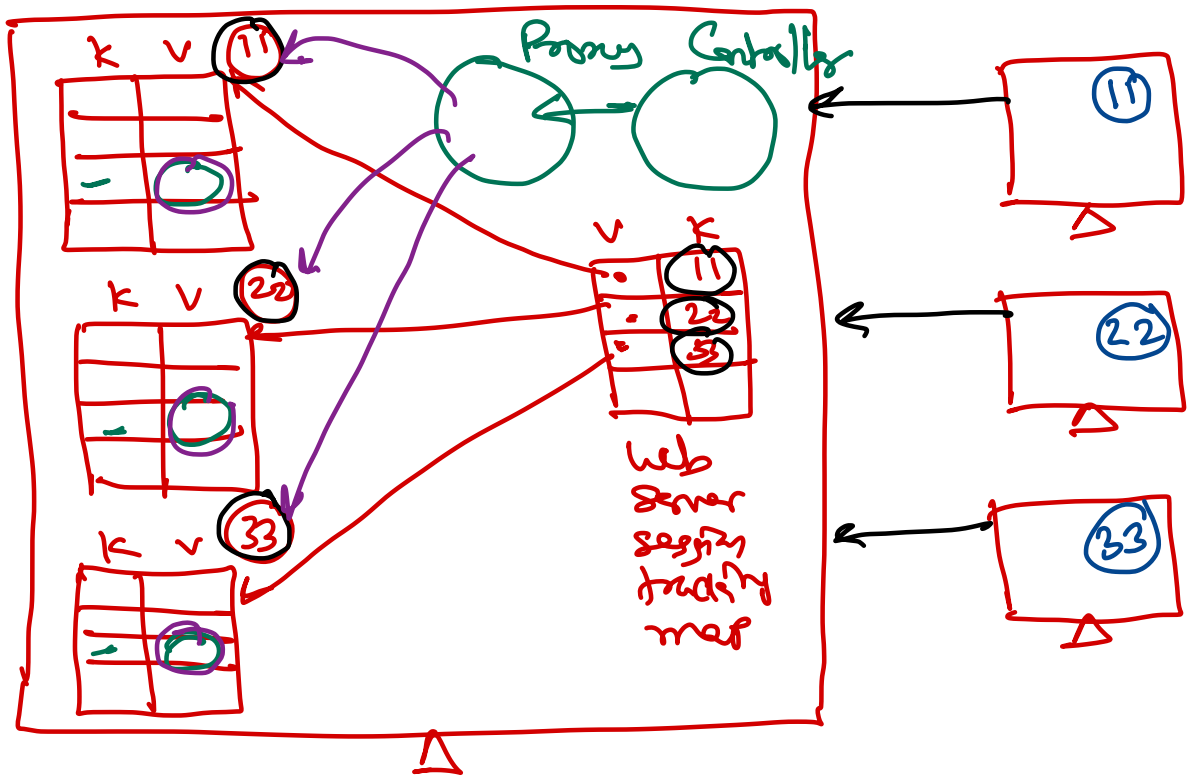
Session and Request scoped beans

- Spring bean scopes
  - Singleton
  - Prototype
  - Session
  - Request

@Component ...
@Scope
web applns.

session.setAttribute("cart", new ArrayList<>());
Cart → add cart or show cart.
cart = session.getAttribute("cart");

- Session and Request scope beans are only possible in web applications.
- The scope management is done via proxies.
- Bean proxy is auto-wired in controller class. Depending on session/request internally new bean is created and made available in that context.



File upload and download

- File upload
  - Add commons-fileupload dependency in pom.xml.
  - In spring mvc config add CommonsMultipartResolver bean.
  - In view form, add HTML file input tag. Make form enctype="multipart/form-data", method="post" and mention action of spring request handler.
  - In request handler take arg @RequestParam("file") CommonsMultipartFile file.
  - File data will be accessible using file.getBytes(). Process that data.
- Hibernate blob handling
  - Create table with binary data as BLOB type.
  - In entity class map column with @Column and @Lob to byte[].
- File download
  - In spring request handler take HttpServletResponse as arg and return type void.
  - resp.setContentType(" application/octet-stream");
  - resp.getOutputStream().write(byteArray);

@RequestMapping

- @RequestMapping attributes
  - value/path = "url-pattern"
  - method = GET | POST | PUT | DELETE
  - params/header = … (map request only if given param or header is present).
  - consumes = … (map request only if given request body type is available).
  - produces = … (produce given response type from handler method)
- One request handler method can be mapped to multiple request methods.
- One request handler method can be restrict to set of request methods.
- To restrict handler method to single request method, shorthand annotations available.
  - @GetMapping
  - @PostMapping
  - @PutMapping
  - @DeleteMapping

Internationalization / Localization

- Adapting computer software to different languages, regional peculiarities and technical requirements of a target locale.
- Internationalization is the process of designing a software application so that it can be adapted to various languages and regions.
- Localization is the process of adapting internationalized software for a region or language. Localization is done for each locale by added respective components.
- Current locale can be accessed in request handler as Locale argument.
- Spring application steps
  - Create messageSource bean and provide base path of properties file.
  - Create properties file for different locale.
  - In views add <s:message code="property-key-name"/>.
- Current locale can be stored using SessionLocaleResolver or CookieLocaleResolver.
- Locale can be changed dynamically using LocaleChangeInterceptor (configured using <mvc:interceptors/>).

Spring MVC validation

- For web applications client side validations are preferred for better user experience
- However client side validations can be bypassed/skipped by client.
- To ensure only validity of data provide server side validations (as well).
- Spring validation framework helps for server side validations.
- Spring supports JSR-303 validations.
  - @NotBlank, @NotEmpty, @Size, @Email, @Pattern, @DateTimeFormat, @Min, @Max
- Steps:
  - Add dependency hibernate-validator in pom.xml
  - Use appropriate annotations on model classes.
  - Use @Valid on @ModelAttribute in request handler method.
  - Next argument in request handler should be BindingResult method.
  - Use <sf:errors/> in view to show error codes.

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>