Spring Security:

Q: How to limit only one session per user and block the subsequent login attempt?

Ans:

The solution is [in the documentation](http://static.springsource.org/spring-security/site/docs/3.1.x/reference/springsecurity-single.html#ns-concurrent-sessions):

Often you would prefer to prevent a second login, in which case you can use

<http>

...

<session-management>

<concurrency-control max-sessions="1" error-if-maximum-exceeded="true" />

</session-management>

</http>

The second login will then be rejected. By “rejected”, we mean that the user will be sent to the authentication-failure-url if form-based login is being used. If the second authentication takes place through another non-interactive mechanism, such as “remember-me”, an “unauthorized” (402) error will be sent to the client. If instead you want to use an error page, you can add the attribute session-authentication-error-url to the session-management element.

So basically set error-if-maximum-exceeded to "true" and remove expired-url attribute from <concurrency-control>

Q: Exception Handling in Spring?

@ControllerAdvice

public class GlobalExceptionController {

@ExceptionHandler(CustomGenericException.class)

public ModelAndView handleCustomException(CustomGenericException ex) {

ModelAndView model = new ModelAndView("error/generic\_error");

model.addObject("errCode", ex.getErrCode());

model.addObject("errMsg", ex.getErrMsg());

return model;

}

@ExceptionHandler(Exception.class)

public ModelAndView handleAllException(Exception ex) {

ModelAndView model = new ModelAndView("error/generic\_error");

model.addObject("errMsg", "this is Exception.class");

return model;

}

}

@ControllerAdvice

public class RestResponseEntityExceptionHandler extends ResponseEntityExceptionHandler {

@ExceptionHandler(value = { IllegalArgumentException.class, IllegalStateException.class })

protected ResponseEntity<Object> handleConflict(RuntimeException ex, WebRequest request) {

String bodyOfResponse = "This should be application specific";

return handleExceptionInternal(ex, bodyOfResponse,

new HttpHeaders(), HttpStatus.CONFLICT, request);

}

}