



**Trainer: Niles Ghule**

*Wake up from Hibernate, Spring up!!!*



# Agenda

- Spring exception handling ✓
- Static resources in Web MVC ✓
- MVC using Annotation config ✓
- MVC security ✓
- REST introduction ✓
- @RestController ✓
- RestTemplate



<Servlet>

↳ name → Spring5

↳ init-param

↳ contextClass, locale

↳ path.

→ default :




spring5-servlet.xml.

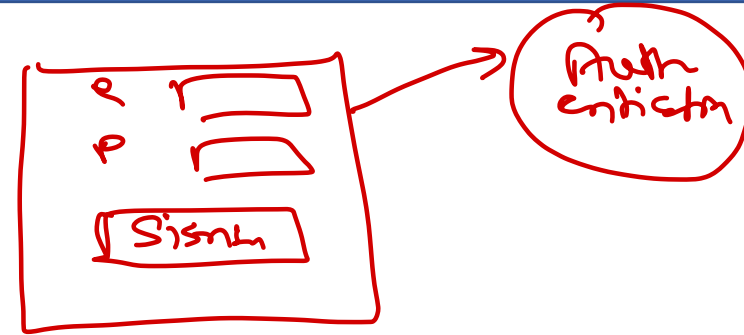
.xml

include.

hibernate-ent5.xml

# Spring WebMVC Security

- Authentication 
  - Checking user/client credentials.
  - Who am I?
- Authorization: 
  - Checking whether user/client is allowed to access the resource.
  - Am I authorized?
- Role Based Security: 
  - Each user is assigned ROLE. A user may be in multiple ROLEs.
  - Each ROLE is allowed access for certain resources & denied access for certain resources.



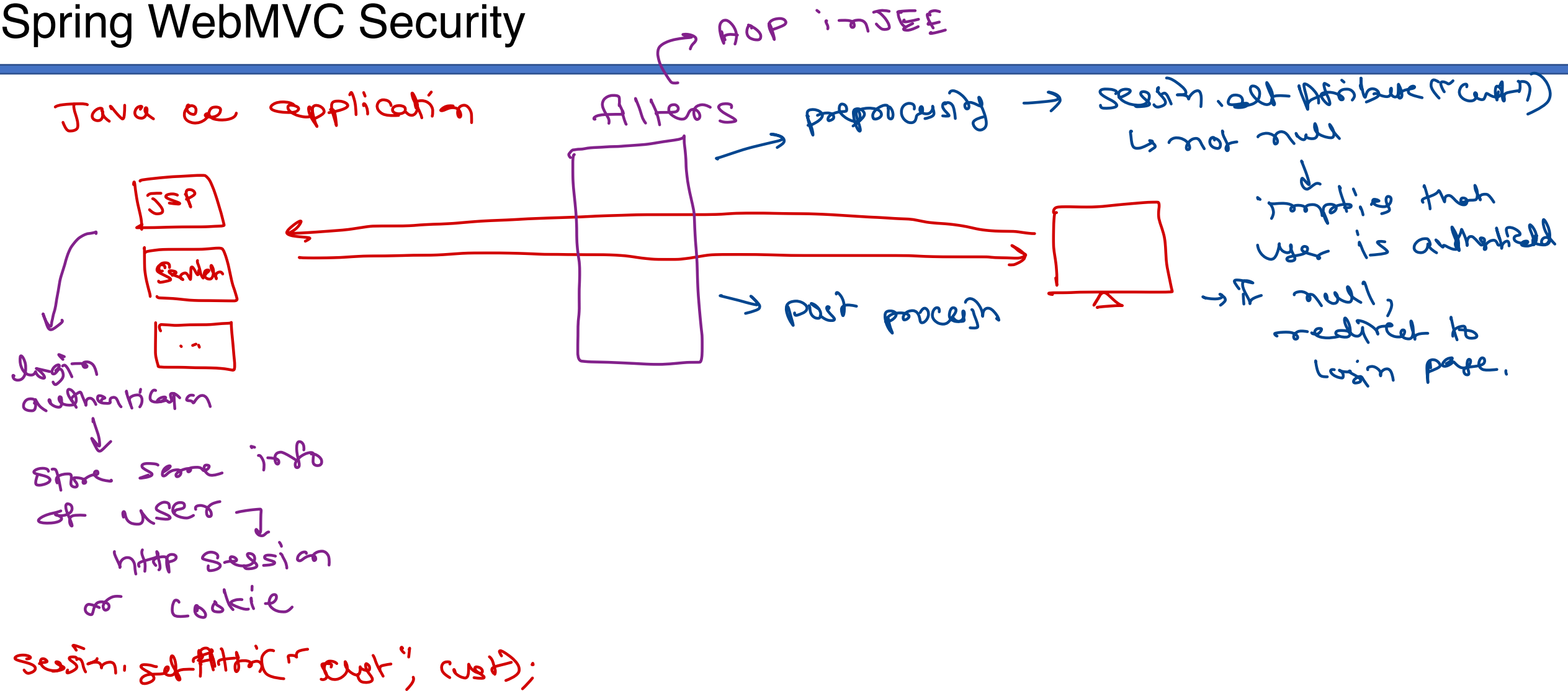
Hand-drawn stick figures representing users. Below them is the word 'developer' written in a wavy line.



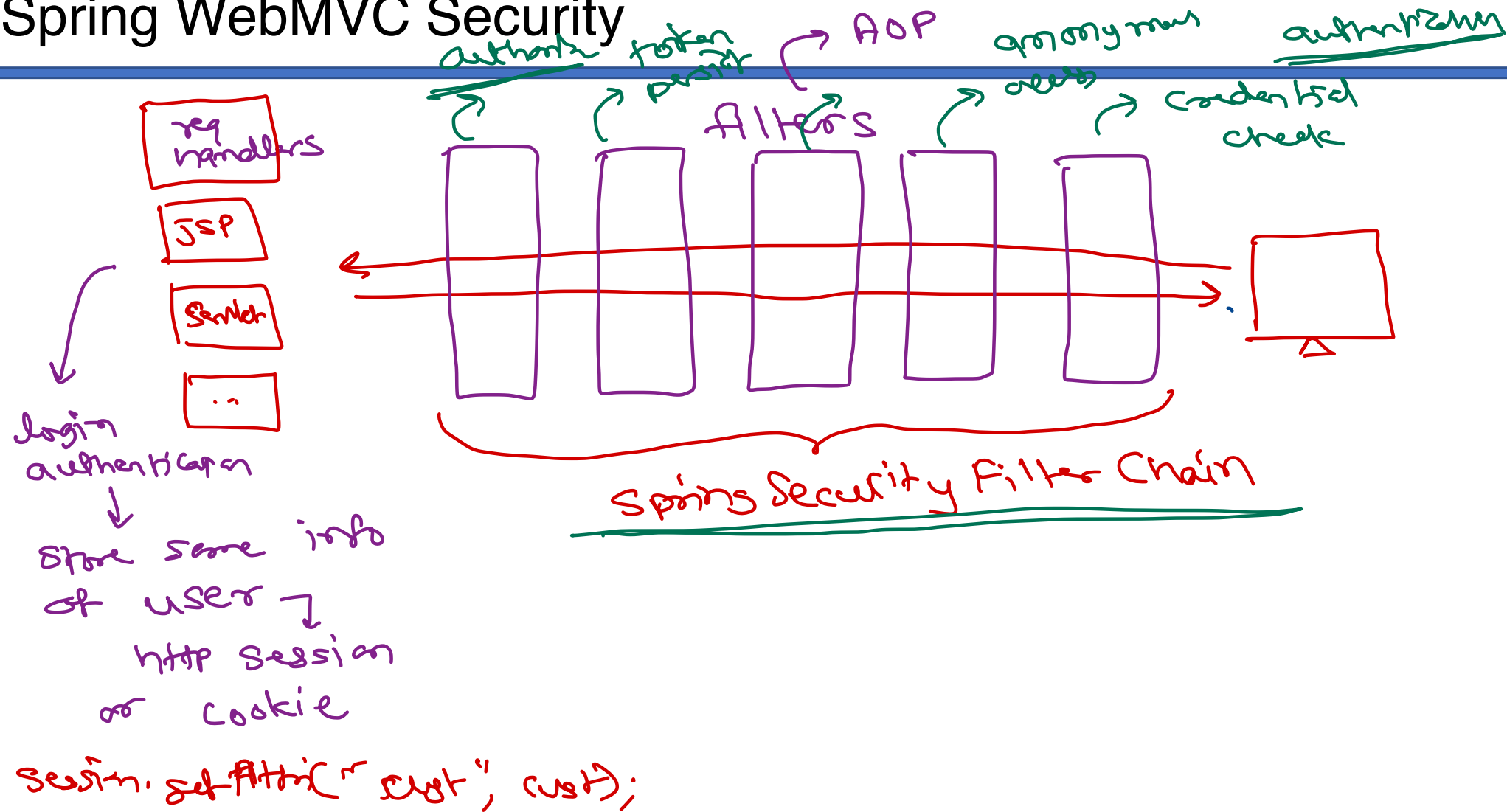
Hand-drawn stick figures representing users. Below them is the text 'Sys admin' written in a wavy line.



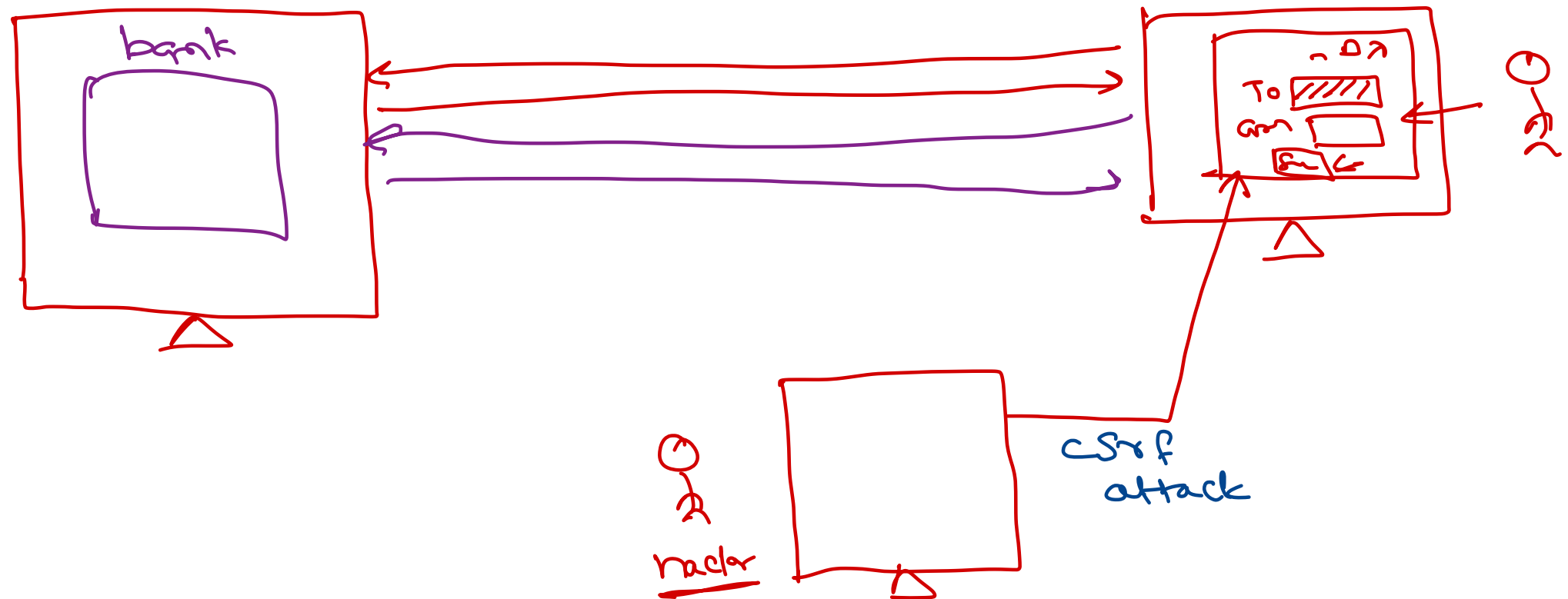
# Spring WebMVC Security

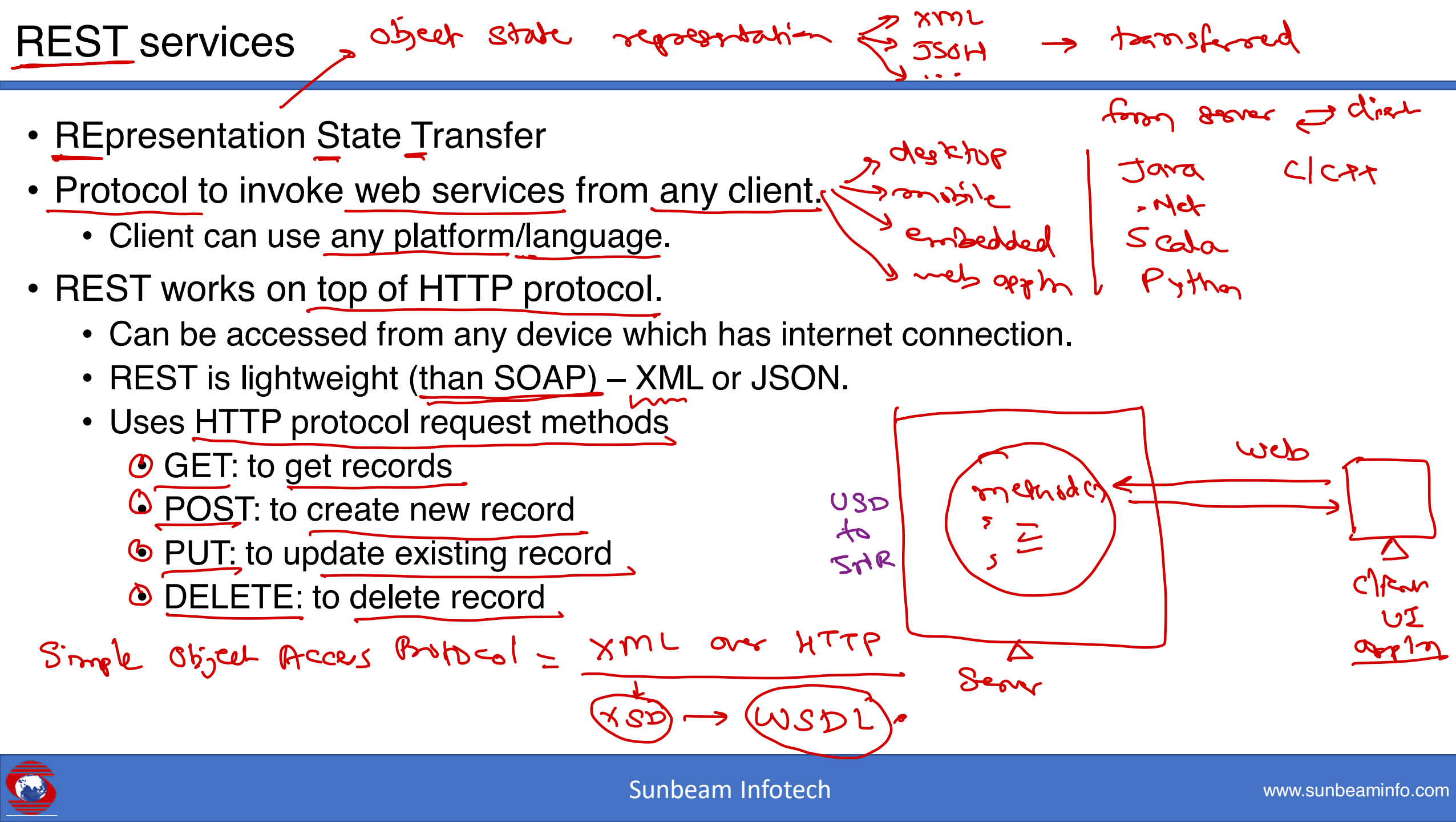


# Spring WebMVC Security



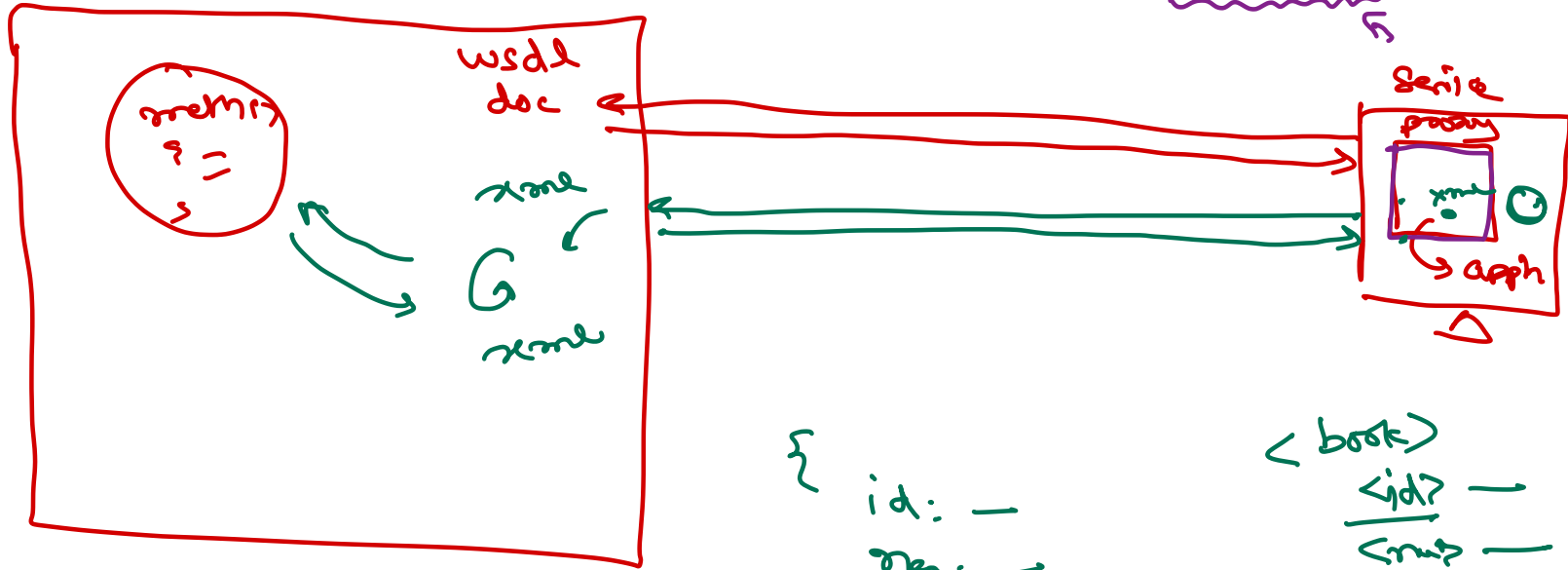
# Spring WebMVC Security







# SOAP Services



```
{
  id: -
  name: -
  author: -
}
```

↳  
↳

```
<book>
  <id> -
  <name> -
  <author> -
</book>
```

- ① Grammar check
- ② bytes,

GET

/rest/books → fetch all books.

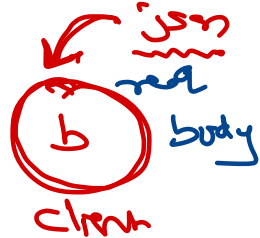
/rest/books/{id} → fetch book at given id

/rest/{subject}/books → fetch book at given subject

POST

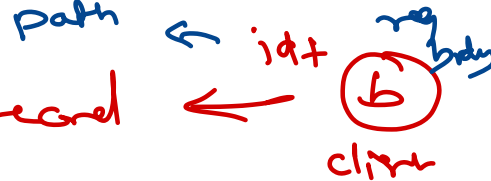
json b → client

/rest/books → insert new books



PUT

/rest/books/{id} → update given record



DELETE

/rest/books/{id} → delete



# Spring REST services

- Based on top of Spring Web MVC. *↑ JSON Conversion*
- Maven dependency: jackson-databind and jackson-databind-xml (if need XML)
- HttpMessageConverter beans:
  - MappingJackson2HttpMessageConverter, MappingJackson2XmlHttpMessageConverter
  - @Override configureMessageConverters()
- Using @Controller *method = G P P D*
  - @GetMapping, @PostMapping, @PutMapping, @DeleteMapping or @RequestMapping
  - @RequestBody, @ResponseBody / ResponseEntity <>
  - @PathVariable, @RequestParam
- Using @RestController
  - @GetMapping, @PostMapping, @PutMapping, @DeleteMapping or @RequestMapping
  - @RequestBody, ResponseEntity <>
  - @PathVariable, @RequestParam
- To manipulate JSON response use @JsonProperty or @JsonIgnore.





*Thank you!*

Nilesh Ghule <nilesh@sunbeaminfo.com>

