Spring security:

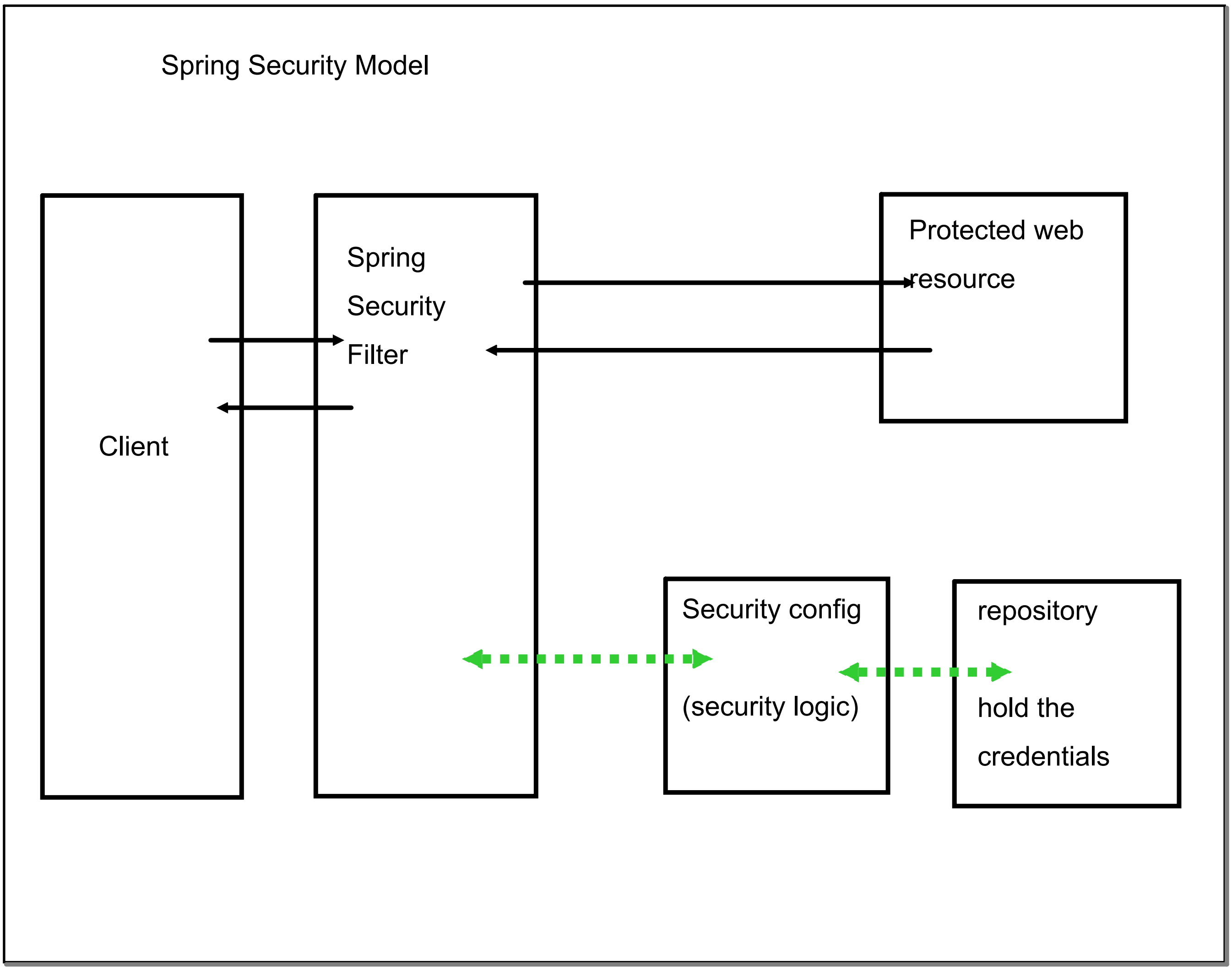
Secure Spring MVC WebApp

Security Model:

Implemented using Servlet filter (background)

Servlet filter : intercepts request, pre-process/post-process

can route web request based on security logic (custom defined)



Spring Security Model

Client

Spring Security Filter

Protected web resource

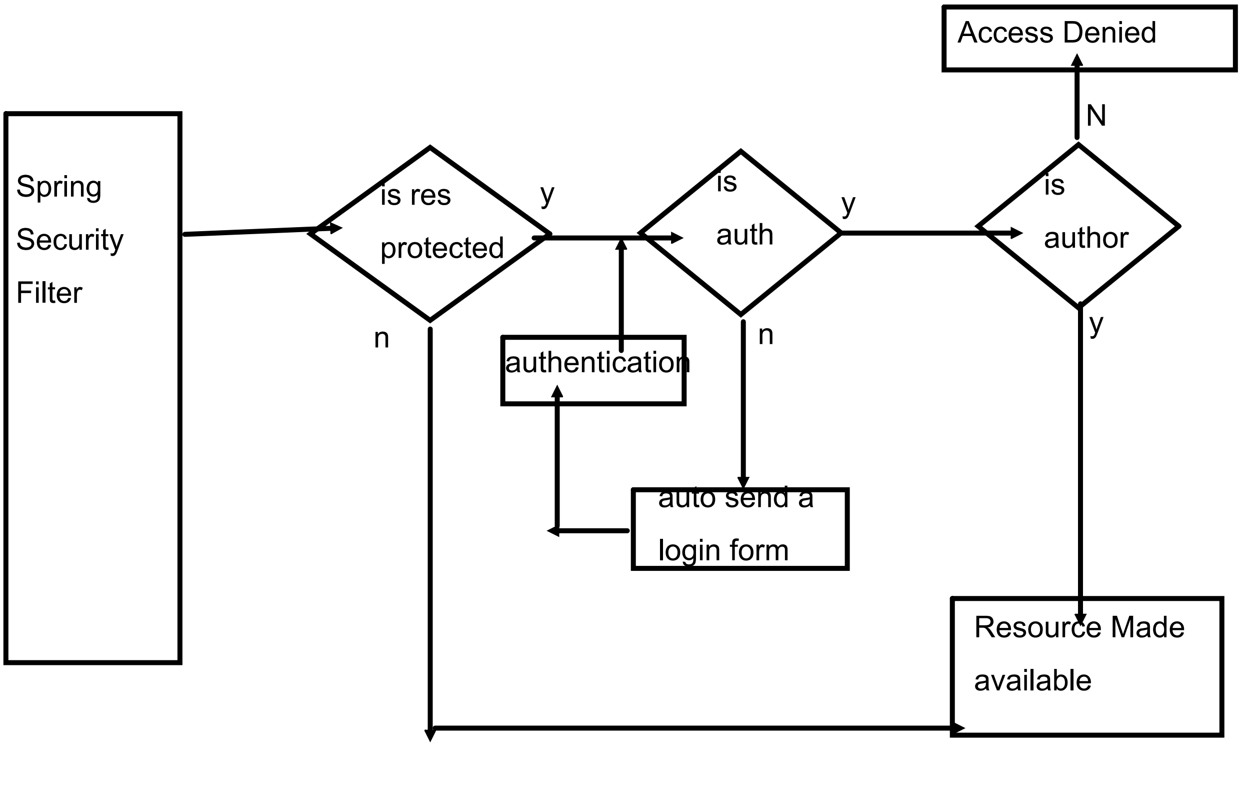
Security config

(security logic)

repository

hold the credentials

Mode of Action



Config:

XML config

JAVA config (class)

Spring Security : works independent of application (declarative security)

Login Approaches

1. HTTP Basic Authentication : Browser based login

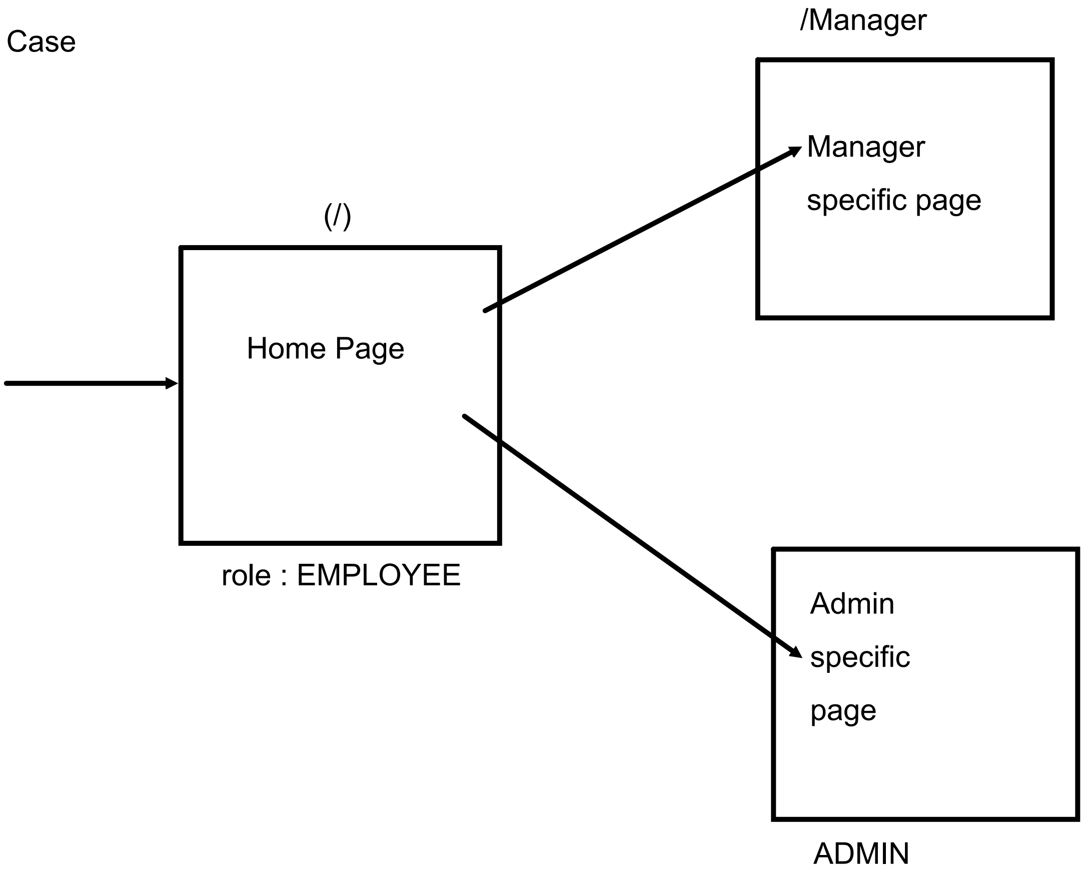
2. Default login form : inbuilt login form provided with security

3. Custom Login form

Credential Repo :

- # In-memory (hard-coded credential directly defined in config file)
- # JDBC
- # LDAP
- # Pluggable

Use Case



/Manager

Manager specific page

(/)

Home Page

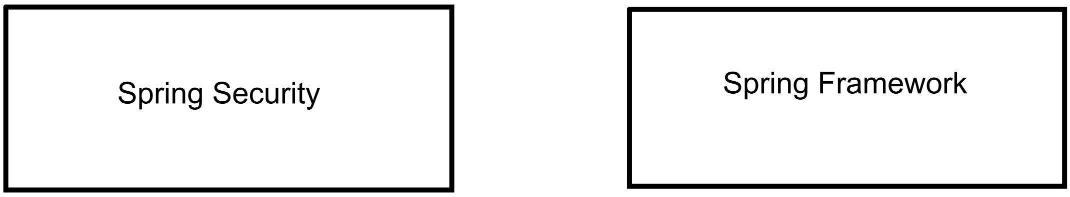
role : EMPLOYEE

Admin specific page

ADMIN

Setup for security

- Dependency
  - spring-security-web
  - spring-security-config



Spring Security

Spring Framework

Two separate projects : not in sync...

Development Process

1. Initialize the Spring Security Filter (Create Spring Security Initializer)
2. Create the Security config
3. add users, pass and roles

# special inbuilt class to register Spring Security filter

# Security Config : class to be inherited : WebSecurityConfigurerAdapter : make available the configure method : auto used by Security Filter

# Credentials : in-memory auth (hard-coded)

==>By default all request would be authenticated (Basic auth)

Custom Login Form

Process

1. Modify the security config file to ref to our custom login form
2. Develop controller/view page

Custom Login Page:

1. form : spring forms (additional security) : submit method : POST
2. field names must be username/password
3. Action URL need not to be backed by any implementation (will be provided by security framework for free)

When login is failed:

# user is provided with login page again : appended with param ?error

Modify the custom login form, check for ?error param an show error msg

# need to add support for jstl tags

Logout:

1. Add logout support to Spring security config
2. Add Logout button in jsp (as submit button in a form)
3. update login form to show "logout" message

Logout form

1. spring form tags
2. method : POST
3. action : /logout (in-built pre-defined url for logout) : implementation is provided for free

Logout Process :

# invalidates users HTTP session....

# send user back to login page

# append the logout param : ?logout

CSRF security concern:

Cross-Site Request Forgery

to protect additional auth token is added to Forms, which on submission is verified

This is by default enabled in Spring Security

Spring forms manages token

in traditional HTML form support for authentication token has to be explicitly added

Access the details about logged in person

Special Spring Security JSP Tag Library

all info of logged in user is assembled in an object : principal

roles : authorities

==> Restrict the web paths access based on user roles

Steps :

1. Create appropriate controller and view pages
2. update roles
3. Config the access over web path

To restrict the web path:

```
antMatchers("<path/url>").hasRole("<role name>")
antMatchers("<path/url>").hasAnyRole("<list of roles>")
```

Add Custom Access Denied Page :

create controller method and view

config the access denied page

User Credentials stored in DB

# need to provide connection (DataSource) and all activities will be managed by Spring Security

# need to provide pre-defined table schemas

Table Schemas

2 tables

1. users : (username and password)

username varchar (PK)

password varchar

enabled TINYINT(1)

2. authorities : (roles of users)

username varchar (FK) : refering the username PK of users table

Unique Key

authority varchar

Spring Security 5, password are stored in specific format

{id}encodedPassword

id : type of encoding

eg : Plain text password : {noop}abc

eg: Bcrypt password hashing : {bcrypt}-----------------

Xampp/ Lamp / WAMP

Configure the data source

- #dependency
  - : mysql-connector
  - : Connection-pool datasource

# Configure the Connection Pool DataSource and make it available to Spring Security

# Configure Spring Security to use DataSource

$2a : Spring Security Support

$2b

$2c

# working on custom table schema

# need to explicitly mention query