**47.1. Spring Security – Overview**

**Here we will learn**:

* Secure Spring MVC Web Apps
* Develop login pages (default and custom)
* Define users and roles with simple authentication
* Protect URLs based on role
* Use JSP tags to hide/show content based on role
* Store users, passwords and roles in DB (plain-text -> encrypted)

**Practical Results**:

* Cover the most common Spring Security tasks that you will need on daily projects
* Not an A to Z reference … for that you can see **Official** **Spring Security Reference Manual**

**Link:**

<https://docs.spring.io/spring-security/site/docs/4.2.3.RELEASE/reference/htmlsingle/>

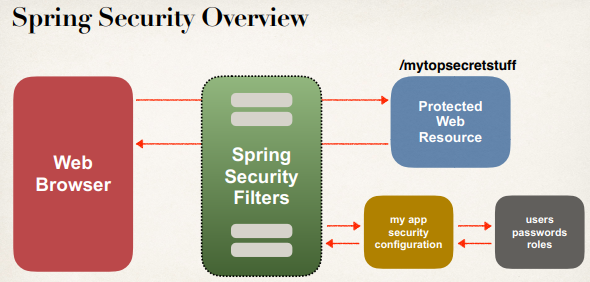
**Spring Security Model**:

* Spring Security defines a framework for security
* Implemented using Servlet filters in the background
* Two methods of securing a Web app: declarative and programmatic

**Spring Security with Servlet Filters**:

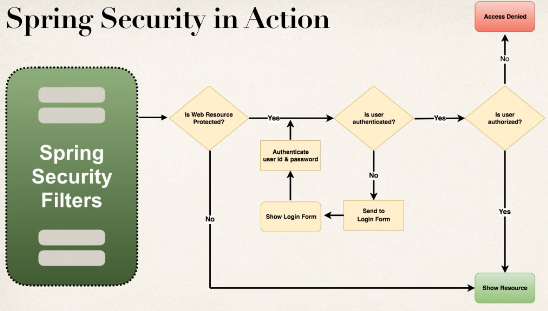
* Servlet Filters are used to pre-process / post-process web requests
* Servlet Filters can route web requests based on security logic
* Spring provides a bulk of security functionality with servlet filters

**Spring Security Overview**:



**Spring Security in Action**:

Spring Security in action with the flow-chart.



**Security Concepts**:

* Authentication
  + Check user id and password with credentials stored in app / db
* Authorization
  + Check to see if user has an authorized role

**Declarative Security**:

* Define application’s security constraints in configuration
  + All Java config (@Configuration, no xml)
  + or Spring XML config
* Provides separation of concerns between application code and security

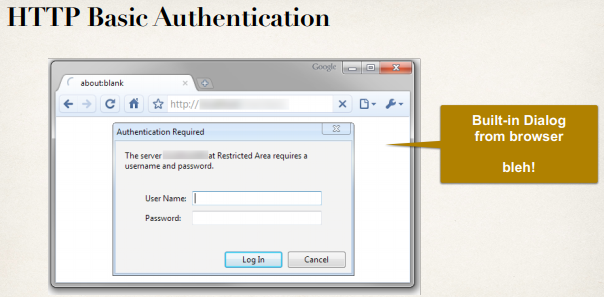
**Programmatic Security**:

* Spring Security provides an API for custom application coding
* Provides greater customization for specific app requirements

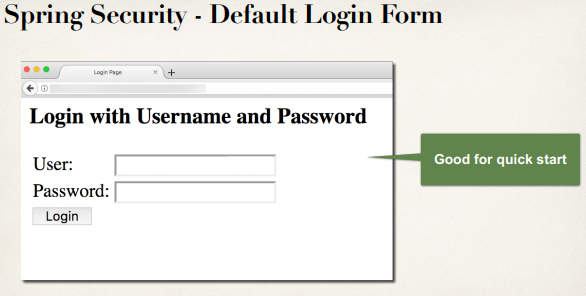
**Different Login Methods**:

* HTTP Basic Authentication
* Default login form
  + Spring Security provides a default login form
* Custom login form
  + Our own look-and-feel, HTML + CSS

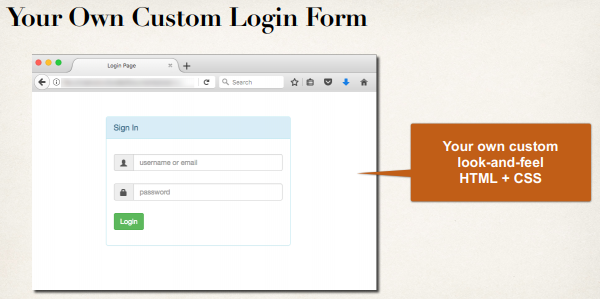
**HTTP Basic Authentication**:



**Spring Security - Default Login Form**:



**Your Own Custom Login Form**:



**Authentication and Authorization (Users Password and Roles)**:

* In-memory
* JDBC
* LDAP
* Custom / Pluggable
* others …

We will cover password storage in DB as plain-text AND encrypted

47.1. Spring Security - Overview