**Spring Security**

***Introduction***

What is Spring Security?

* Spring Security helps us to achieve application level security.

Like:

1. What is the kind of functionality you want to allow and to whom?
2. What are the things that your user can do in the context of your application?
3. What is the kind of data that you want to expose and to whom?

This is Application Level Security!!

Now the main threat is that Hackers trying to hack your secured applications every day.   
Basically what you need is not to handle security yourself but to hire a security guard to secure your building. The security guard intercepts every visitor ask them who they are and what do they want?

Spring Security is like that security guard who sits and asks these two questions to each and every request, who is this request from and what do they want?

Why Spring Security?

* Spring Security is an Application Framework that lets us do application level security.
* By default, it gives us:

1. Login and logout functionality.
2. Allow/block access to URLs to logged in users.
3. Allow/block access to URLs to logged in users and with certain Roles.

What can Spring Security do?

* User name/ password authentication.
* Application level authorization.
* Intra Application authorization.
* Micro service security (using Web Tokens, JWT)
* Method level security.

***FIVE CORE CONCEPTS OF SPRING SECURITY:***

* Authentication
* Authorization
* Principal
* Granted Authority
* Roles

1. **Authentication:**

Let us assume that the security guard asks you that who you are. So, it is your responsibility to answer who you are and not only that you answering with your name is enough, but you need to produce some kind of unique Id which only you are in possession with and no other impersonator can produce.  
  
In technical terms, you could have an Username(who you are) and a password (your unique Id).  
  
This is Authentication.  
  
This type of authentication is called:

***Knowledge Based Authentication***.

The authentication is based on the knowledge you have. Like:

🡪 Password

🡪Pin code

🡪Answer to a secret question/personal question

Advantage of knowledge based authentication:

1. Easy to implement and use.

Disadvantage of knowledge based authentication:

1. Not fully safe 🡪 What if someone steals your password, they can impersonate you easily. This could be overcome with the help of a different type of authentication. That is Possession Based Authentication.

***Possession Based Authentication:***

1) Phone/text messages

2) Key cards.

***Multi-Factor Authentication:***

Combination of Knowledge based authentication + Possession based authentication.  
Like:

Enter the password and the provide the text message sent to the user’s mobile number.

1. **Authorization:**

Authorization is that which user is allowed to access which functionality. To achieve Authorization, we need to get the Authentication done first. Before we give the user permission to access what the user wants to access we need to know who the user is first.

***Fundamental difference between Authentication and Authorization:***

**Authentication:**   
Who is the user?

**Authorization:**

Are they allowed to do what they are asking for?

1. **Principal:**

Principal is the person we have identified through the process of authentication.   
In other words, it is the currently logged in user.

Once you authenticate with any application by giving the user name and the password, Spring security establishes a principal which remembers it.

This is the reason why you need to authenticate with an application once and you don’t need to enter your user name and password with every request/page load. This is because principal remembers the currently logged in user details.

1. **Granted Authority:**

Access assigned to users. That means which user is allowed to access which tasks.   
Let us take an example, Assume that there is a store where we have clerks, department manager and store manager.

1. Clerks – would have the access to ***do checkout***, ***make store announcements.***
2. Department manager – would have the access to ***do checkout***, ***make store announcements, view department financials, view department inventory.***
3. Store manager – would have access to ***do checkout***, ***make store announcements, view department financials, view department inventory, view store financials.***

***\*\*But there is a disadvantage with granted authority--- Let’s say the retail store hires new employees (clerks), then someone has to assign all the correct authority every time a new employee is hired.   
  
This can be overcome with the help of the concept of Roles. \*\****

1. **ROLES:**

Group of authorities that are usually assigned together. So for the previous example, we could create a role\_store\_clerk and assign all the authorities (do checkout, make store announcements).  
Now whenever a new Store clerk is hired we can just assign them to “role\_store\_clerk” and automatically they get all the authorities assigned.