**SPRING SECURITY**

Spring security is a mechanism that helps to secure our application from being used by some unauthorized person.

A diagram of a person's face

Description automatically generated

Spring security is like a watchman sitting in the front of your application and keeping I safe and asking everyone who comes some set who questions:

Who are you and what do you want.

We have access control specific to our application

A screen shot of a security framework

Description automatically generated

**What spring security can do?  
A screen shot of a computer

Description automatically generated**

using spring security we can achieve the username and password authentication,and

also can do app level authorizations an dmany more things.

**Basic spring security terms**

**1.Authentication:** who is this person

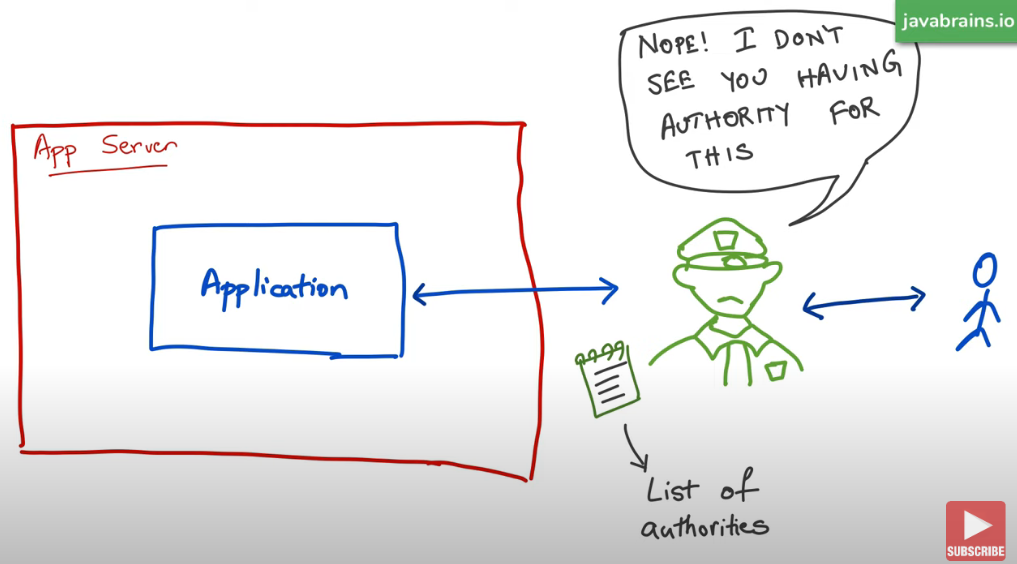
**2.Authorization:** what is he allowed to do it.what is he authorized to do.

**3.Principal:**currently logged in user.application remembers a person once logged in as a principal

**4.Roles:** group of authoritiez.a user is assigned a role in which all the authorities are mentioned which he is allowed to perform.

**How does authorization happen?**

If this is the logged in user allow him to do tis.if it is some other person allow him to do anything else.in spring security context it is known as **granted authority**.



Spring security already has a list of all these granted authorities so that it can tell if a person is allowed to perform that action or not.

**Security types**

**1.knowledge based**

In knowledge based authentication we need to have knowldeg of our username and password

**2.posseeion based**

In possession based security we need to have some device with us through which we can tell that we are authenticated user of that thing.

So generally the person can have a mobile phone for an otp or a swipe card

**MULTI FACTOR AUTHENTICATION**

In multifactpr authentication we use both knowledge based and possesiion based security.

**SPRING SECURITY KEY TERMS:**

**1.AUTHENTAICATION:**means a person is authentaicated or not

**2.AUTHORIZATION:** what an authenticated person is authorized to do

**3.PRINCIPAL:** the current loggen in user is the principal.We do not have to pass the credentials every time we make a call as the spring will remember us as a principal.

**4.GRANTED AUTHORITY**

**5.ROLE:** a group of people is given a role which makes them acessbile to some services and things.

**HOWTO ADD SPRING SECURITY TO YOUR SPRING BOOT APP**

In order to add the first level spring security to our spring boot app we simple need to add the spring starter security dependency to our pom.xml.

Dependency

Spring-booot-starter-security

A screenshot of a login box

Description automatically generated

Just by adding the above dependency we can get the above page in our application.

**HOW DOES SPRING SECURITY WORK**

**To**  understand this first we will understand servelts concept.

First of all when a request comes its url is checked by the servlet container and then the container handles the request to be handled by a particular servlet.

**So now how spring security works:**

1.there are servlets on which are spring application is working on

2.Spring security will act as a filter

3.this filter will now check ever request that is coming in.

4.If the request has acceptable headers then it allows otherwise denies.

5.It does not checks for the eroor pages.

Diagram

Description automatically generated

**DEFAULT BEHAVIOUR OF SPRING SECUIRTY**

**A screen shot of a computer security

Description automatically generated**

Spring adds a mandatory authentication for urls and adds a login form and also if login not successful it has an error handling mechanism of giving an error page .By default spring will create a user and sets a default password.

**INTERNAL WORKING OF SPRING SECURITY**

When we add the spring security dependency.then in our application it will create a login page and a default password and username being the default user which can be seen in the startup console.

Every time we restart the application a new password is generated.

Now if the credentials are not accurate then it will return a bad credentials message.

Graphical user interface, application

Description automatically generated

**SETTING UP OUR CUSTOM USEERNAME AN DPASSWORD**

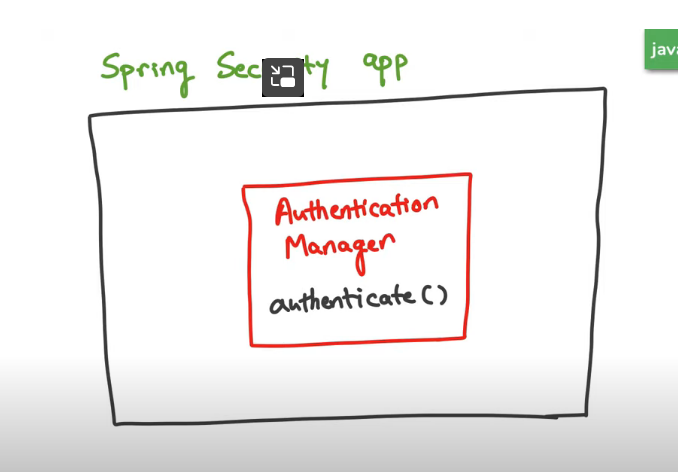
For this we need to go to our apliaction.properties and add the username and password there.

Graphical user interface, text

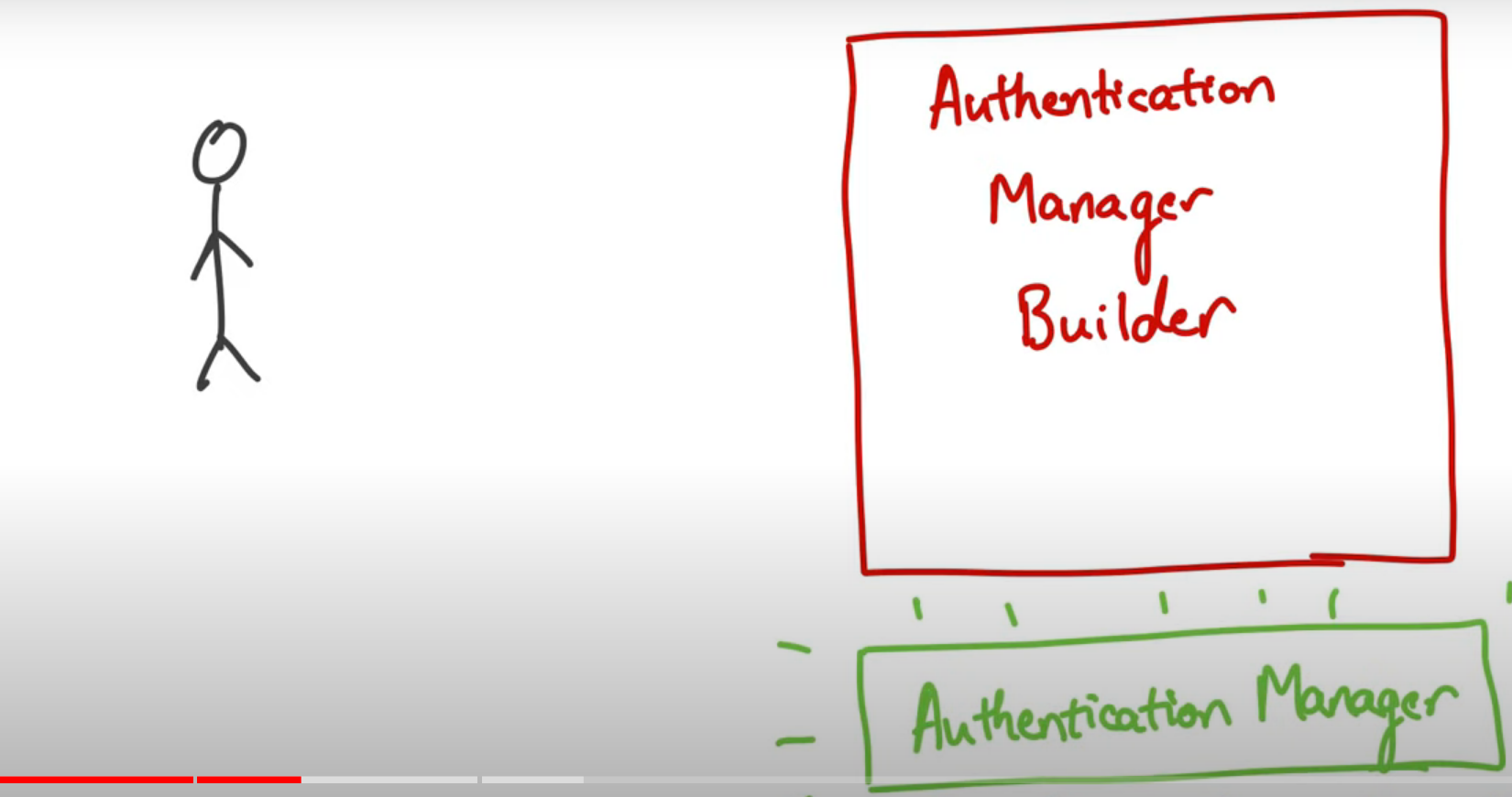
Description automatically generated

**ADDING OUR OWN USERS TO GET THE SPRING SECURITY**

We need to add our own users to authenticate the spring so we will be making use of an authentication manager to do so.This class has a method named as authenticate which help us do the task.



There will be an authentication manager builder which will build an authentication manager for us which will have details of all the users and their password whom we ant to authenticate.



**SPRING SECURITY WITH JDBC**

First we make a class and extend the WebsecurityConfigurerAdapter

A screen shot of a computer

Description automatically generated

Now we want to fetch the username and password from the database

We will use application.properties

A screenshot of a computer

Description automatically generated

Now we will be making a schema.sql

A computer screen with white and blue text

Description automatically generated

After this to enter values in this schema we will make a data.sql

A computer screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated

**SPRING SECURITY WITH JPA**

First of all we will make a table in the database and enter the username and password in tahtw hich needs to be accessed while logging in into the system.

Table

Description automatically generated

2.create a spring project

(spring web,mysql,jpa,security)

Generate the project and open in your ide.

3.WE make a controller which can be accessed by all the application developers

Text

Description automatically generated

4.now setup the authorization

Here first line says that authorize all the requests and next it tells if the url is/normal then allow people who have a normal role

Text

Description automatically generated

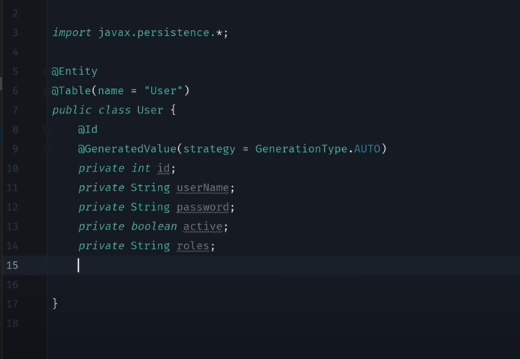
5.NOW SETUP THE AUTHENTICATION

Diagram

Description automatically generated

**STEPS OF SPRING SECURITY WITH JPA**

First of all make an entity which will represent a table in your database.

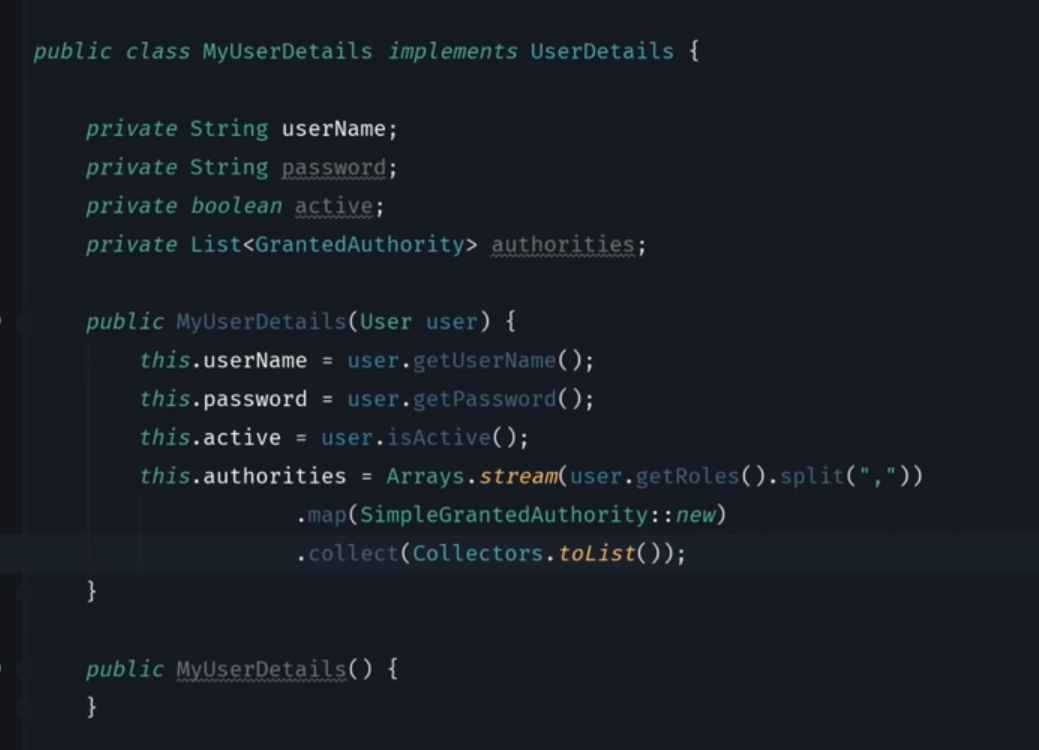


Now make an intefcae or a repository



Now in userdetails service we will do as below.From the repository we will find out wat is the user to be found from the database.





**LDAP**

**A computer screen shot of a computer code

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

Add the above to your dependencies

A screen shot of a computer

Description automatically generated

Application.properties