**Research document**

Productivity Web Application

Prepared by: Lyutfi Ismail

01/11/2021

# Table of Contents

[Table of Contents 2](#_Toc87023075)

[1. Research approach 3](#_Toc87023076)

[2. Problem introduction 4](#_Toc87023077)

[3. Research questions 5](#_Toc87023078)

[3.1. Main question 5](#_Toc87023079)

[3.2. Sub-questions 5](#_Toc87023080)

[4. DOT framework research strategy 6](#_Toc87023081)

[4.1. What does a secure connection mean? 6](#_Toc87023082)

[4.2. What is Authentication and Authorisation? 8](#_Toc87023083)

[4.3. What are the different methods for setting up authentication and authorisation in SpringBoot? 9](#_Toc87023084)

[5. Results and conclusions 12](#_Toc87023085)

[5.1. What does a secure connection mean? 12](#_Toc87023086)

[5.2. What is Authentication and Authorisation? 12](#_Toc87023087)

[5.3. What are the different methods for setting up authentication and authorisation in SpringBoot? 12](#_Toc87023088)

[6. References 13](#_Toc87023089)

# Research approach

Research in ICT aims at creating an ICT product that fits the needs of the client, in this case this product will be a web application that help improve productivity. To help bring structure to the research we will use the DOT framework. The DOT framework has multiple research domains with research strategies that can be used to support the decisions made in a project.

Diagram

Description automatically generated

The DOT Framework is a tool that will be used in the research for this project. An overview will be made what method will be used in regard to the research questions.

# Problem introduction

In this document we will focus on the problem of Authentication and Authorisation, and how to implement it in the Spring Boot framework.

The overall goal of the project is to create a web application. Building such an application requires a front-end, which is the website a user will interact with, and a back-end (from here on referred as the “API”), which handles the data that the front-end generates and outputs on the page. Because of this separation of responsibility, a connection must be set-up between the two parts. That connection will use the HTTP protocol and it will run through the internet in the final product.

The last point is why a secure connection must be established between the two clients, so that only Authorised and Authenticated requests can be sent. There are multiple ways to implement this secure connection and we will research those methods in this document.

# Research questions

## Main question

How to set up a secure connection between the front-end and the API?

## Sub-questions

* + 1. What does a secure connection mean?
    2. What is Authentication and Authorisation?
    3. What are the different methods for setting up authentication and authorisation in SpringBoot?

# DOT framework research strategy

In this segment we will go over each sub question and try to answer it, while using the DOT framework’s research strategies.

## What does a secure connection mean?

We need to understand what a secure connection between the front-end and the API means.

* + 1. Library
* Literature study – We will look through different internet articles to find out what a secure connection means and how it works.

Here we are introduced to the SSL certificate –

“SSL stands for Secure Sockets Layer and, in short, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details. The two systems can be a server and a client (for example, a shopping website and browser) or server to server (for example, an application with personal identifiable information or with payroll information).” [1]

* Best, good, and bad practices – We can look at what type of connection most websites use.

We find out that the standard has become such that every major website implements SSL connection using the HTTPS protocol.

* Available product analysis– After we have learned that SSL is an important feature for setting up a secure connection, we need to find out how it can be implemented in our use case.

“In this spring boot example, learn to configure web application to run on SSL (HTTPS) with self-signed certificate. Also learn to create SSL cert, as well.

Table of Contents

Terminology

Create your own self signed SSL certificate

Create Spring-boot application and configure SSL

Redirect to HTTPS from HTTP “ [2]

Using this tutorial an easy solution can be implemented to use spring boot with a secure SSL connection.

## What is Authentication and Authorisation?

We need to understand the difference between the two and what they mean in the software domain.

* + 1. Library
* Literature study - We will look through different internet articles to find out what Authentication and Authorisation is.

“In the digital world, authentication and authorization accomplish these same goals. Authentication is used to verify that users really are who they represent themselves to be. Once this has been confirmed, authorization is then used to grant the user permission to access different levels of information and perform specific functions, depending on the rules established for different types of users.” [3]

## What are the different methods for setting up authentication and authorisation in SpringBoot?

To use our finding form the last two sub question, we need to learn how to implement a secure connection between the front-end of our application and our API. And specifically, since our API uses the SpringBoot framework, we need to find the specific solution for that framework.

* + 1. Library –

We need to first understand how such a connection works in SpringBoot in theory.

* Literature study – We will look through different internet article to find an explanation on how authentication and authorisation works in SpringBoot.

“Application security boils down to two more or less independent problems: authentication (who are you?) and authorization (what are you allowed to do?). Sometimes people say, “access control” instead of "authorization", which can get confusing, but it can be helpful to think of it that way because “authorization” is overloaded in other places. Spring Security has an architecture that is designed to separate authentication from authorization and has strategies and extension points for both.” [4]

From this article we can find out the SpringBoot has an internal integration that allows users to implement auth/auth form within SpringBoot.

How that we know this, we can find that there are different solutions for implementation this feature. Some example solutions are:

1. JWT
2. Oauth2

After reading what they are we have decided to further research JWT based security.

Let’s see what JWT really is:

“JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA" [5]

* Available product analysis – now we need to understand if JWT security can be implemented in SpringBoot.

Since we know that SpringBoot already supports JWT, we can look at existing project and tutorials to understand how it can be implanted in our use case.

“In this tutorial we will be developing a Spring Boot Application that makes use of JWT authentication for securing an exposed REST API. In this example we will be making use of hard coded user values for User Authentication. In next tutorial we will be implementing Spring Boot + JWT + MYSQL JPA for storing and fetching user credentials. Any user will be able to consume this API only if it has a valid JSON Web Token (JWT). In a previous tutorial we have seen what is JWT, when and how to   
use it.” [5]

* + 1. Prototyping – After reading the article, next step will be to follow the turorial and create a prototype to test its functionality.

After testing the simple implementation of our problem, we can go into adding it to our main project and securing our application.

# Results and conclusions

After researching the sub question, we had, we can go over the results and conclusion we reach from each one of them, and understand how that helped with our understanding of the main research question.

## What does a secure connection mean?

* + 1. Result – With the research done on this question, we found out that, for the connection itself to be secured, we need to encrypt the data that is being transferred using the SSL certificate. To implement that feature we need to upgrade the protocol SpringBoot provides from the basic “HTTP” to the secure version “HTTPS”, that implements SSL by default. We looked at how this can be done in SpringBoot and found that the implementation is simple for the feature that it provides.
    2. Conclusion – Now we know why securing the data is important and we know how to implement it using SpringBoot. We reach the conclusion that this feature should be implemented in our project.

## What is Authentication and Authorisation?

* + 1. Result – With the research done on this question, we found the difference between Authentication and Authorisation, and how it really functions. This will help us when we need to secure our application to know the fundamentals security features for a connection between a client and server.
    2. Conclusion – Now that we know what Authentication and Authorisation is, when implementing security on our project, we will know why securing the access to our API is important and how it fundamentally functions.

## What are the different methods for setting up authentication and authorisation in SpringBoot?

* + 1. Result – With the research done on this question, we now know the different methods of implanting authentication and authorisation in our project. We learned that SpringBoot has internal tools for implementing basic auth/auth functionality. We can improve this basic service by using JWT security, which is also supported for implementation by the framework.
    2. Conclusion – Now that we know that Spring boot supports internally auth/auth services, we know that implementing will not be difficult. To build on top of that, to add extra security, we can implement JWT to use together with the SpringBoot services.

# References

1. *What is an SSL certificate? (n.d.). www.websecurity.digicert.com. Retrieved November 5, 2021, from* [*https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https*](https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https)
2. *Spring Boot SSL [https] Example. (2020, December 26). howtodoinjava.com. Retrieved November 5, 2021, from* [*https://howtodoinjava.com/spring-boot/spring-boot-ssl-https-example/*](https://howtodoinjava.com/spring-boot/spring-boot-ssl-https-example/)
3. *What is the difference between authentication and authorization? (2021, July 19). www.sailpoint.com. Retrieved November 5, 2021, from* [*https://www.sailpoint.com/identity-library/difference-between-authentication-and-authorization/*](https://www.sailpoint.com/identity-library/difference-between-authentication-and-authorization/)
4. *Spring security architecture. (n.d.). spring.io. Retrieved November 5, 2021, from* [*https://spring.io/guides/topicals/spring-security-architecture*](https://spring.io/guides/topicals/spring-security-architecture)
5. *Spring boot security + JWT hello world example. (n.d.). www.javainuse.com. Retrieved November 5, 2021, from* [*https://www.javainuse.com/spring/boot-jwt*](https://www.javainuse.com/spring/boot-jwt)