

# Easy Connect

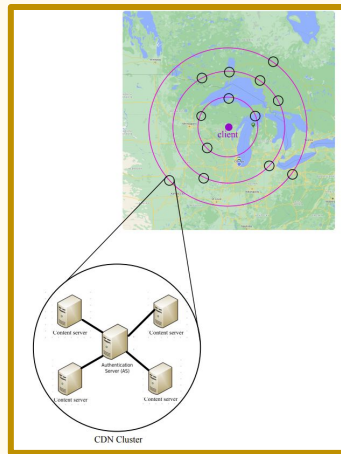
---

By: Linfeng Li, Claudio Jimenez, Jaoudat Karime

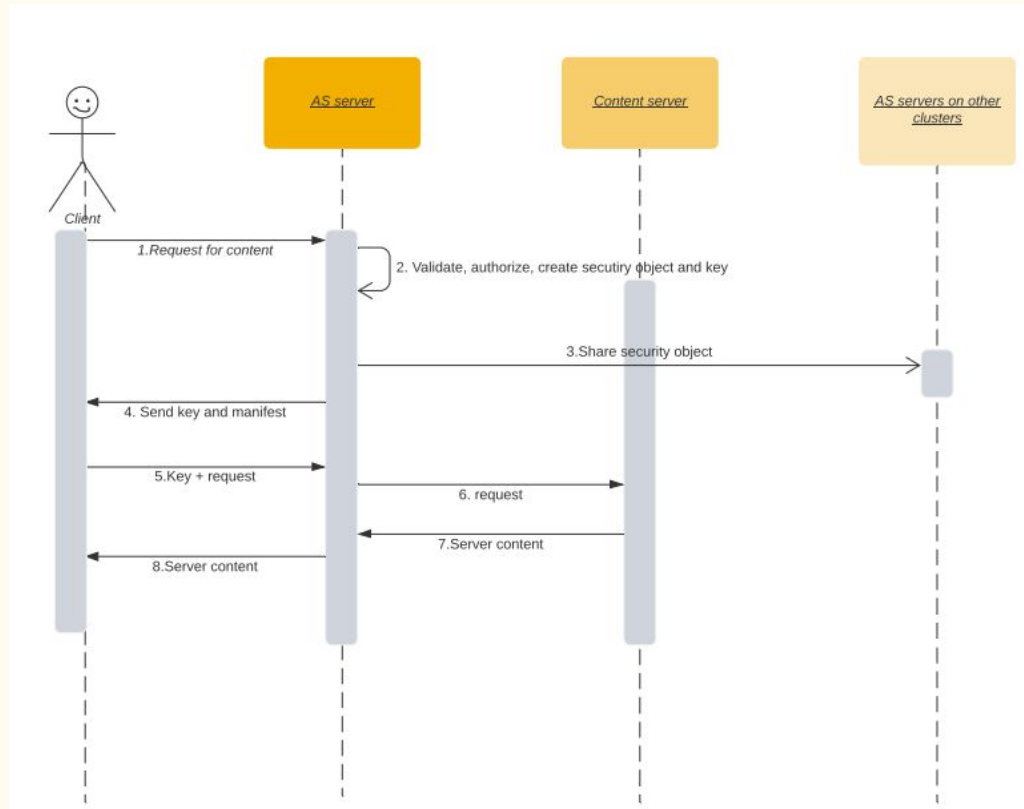


# General Overview

- Our purpose for this project is to share a security object amongst servers within a certain radius of a client to ease connection
  - We emulate this is by having three folders with each holding its own portion of the project
    - Server: The starting authentication server
    - Server2: Authentication server that client switches to
    - Client: Client that will change between those two servers



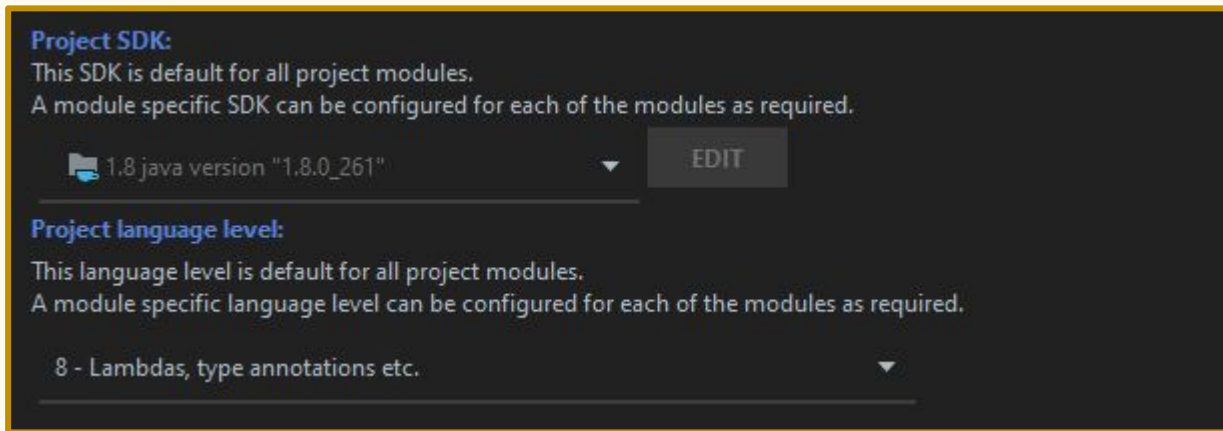
# Sequence Diagram for Connecting Client to Server





# Pre-Compilation Setup

1. To ensure the code runs efficiently please ensure that you have the following settings installed/inserted into your IntelliJ environment.
  - a. Ensure that your SDK is Java 8 as well as the project language level of Java 8
    - i. Below is a screenshot of how IntelliJ environment should be configured



# How to run project on IntelliJ Overview

1. Start by entering the folder called Server and run the Main within the folder.
2. Now in a separate IntelliJ window open the server2 folder and run the Main class within there.
3. Lastly, open the client folder in a final IntelliJ window and run the Main class.
  - a. After running the client class enter the following username and password
    - i. Username: username
    - ii. Password: password
4. The two servers will now be connected together and the client will be connected to server. (Which is the first server that was ran)

# Step 1: Running Server #1

- Once Server 1 is running it will output this statement to the terminal

- ```
hello from auth server 1  
  
initiating auth servers and populate its content servers  
  
waiting to connect nearby auth servers
```

- This statement proclaims that we successfully booted authentication server #1 populating its content server and now is waiting to connect to other authentication servers.



## Step 2: Running Server #2

- Once authentication server #2 is running it shall connect to server #1 and server #1 will display this output on the left while server #2 will display the output on the right.

○

```
hello from auth server 1  
initiating auth servers and populate its content servers  
waiting to connect nearby auth servers  
Near by auth server connected!  
Server started waiting for client to connect
```

```
hello from auth server 2  
initiating auth servers and populate its content servers  
connected with 127.0.0.1
```

- This output confirms that we have interconnected the two servers and are now ready to receive a client

## Step 3: Connecting a Client

- Once the Main class in the client is ran, output from the first server is on the left and output in the client is on the right.

○

```
hello from auth server 1  
initiating auth servers and populate its content servers  
waiting to connect nearby auth servers  
Near by auth server connected!  
Server started waiting for client to connect  
Client connected!
```

```
Enter username |
```

## Step 3: Connecting a Client cont..

- Once username and password are entered the client console will output on the right and the server will display that the client has connected on the left.

○

```
hello from auth server 1  
initiating auth servers and populate its content servers  
waiting to connect nearby auth servers  
Near by auth server connected!  
Server started waiting for client to connect  
Client connected!  
client connected to auth server at: 127.0.0.1
```

```
Enter username username  
Enter password password  
login success  
1: get content  
2: switch AS and get content  
3: logout  
Enter your input:
```

# Step 4: Command Explanation

## 1. Get Content

- a. Simulates a content stream being received from the content Server

## 2. Switch Authentication Server and Get Content:

- a. When 1 is entered it will display the **server number** as well as the **content identifier**

- b.

```
Enter your input: 1
```

```
serving content from content server 1 with identifier 0
```

- c. Until switched will server number remain content server 1
- d. Identifier will then increment by one even if switched to a different server

```
Enter username username
Enter password password
login success
1: get content
2: switch AS and get content
3: logout
Enter your input:
```

## Step 4: Command Explanation cont...

- When the client inserts 2 into the command line it will then update the content server it is being served from while the identifier stays the same and increments by one.
  - This is the equivalent of changing servers mid stream of content.
  - **Green** shows the content server changing
  - **Red** shows the identifier not resetting to 0

```
Enter your input: 1
serving content from content server 1 with identifier 2
1: get content
2: switch AS and get content
3: logout

Enter your input: 2
serving content from content server 2 with identifier 3
1: get content
2: switch AS and get content
3: logout

Enter your input: |
```

# Exiting Client Login

- When Client wants to logout of the server, user can enter 3 to logout and the following output is shown in the client.

- 

```
1: get content
2: switch AS and get content
3: logout
Enter your input: 3

Process finished with exit code 0
```

# Login Expiring Output

- If the login security object times out, the client console will display the following message indicating that the time session is over.
  - This will be displayed when the client either asks for content or to switch authentication servers

- ```
Enter your input: 2
login expired
1: get content
2: switch AS and get content
3: logout
```





# Testing Overview

- Testing includes initialization of the following classes:
  - Authentication Server
  - Body
  - ContentServer
  - Security Object
  - Server
- Running the tests for each section is done by right clicking and selecting the Run Tests Play Button