# Limited login on multiple devices with one account && code verification

Yuqi Yang - May 26

## Introduction:

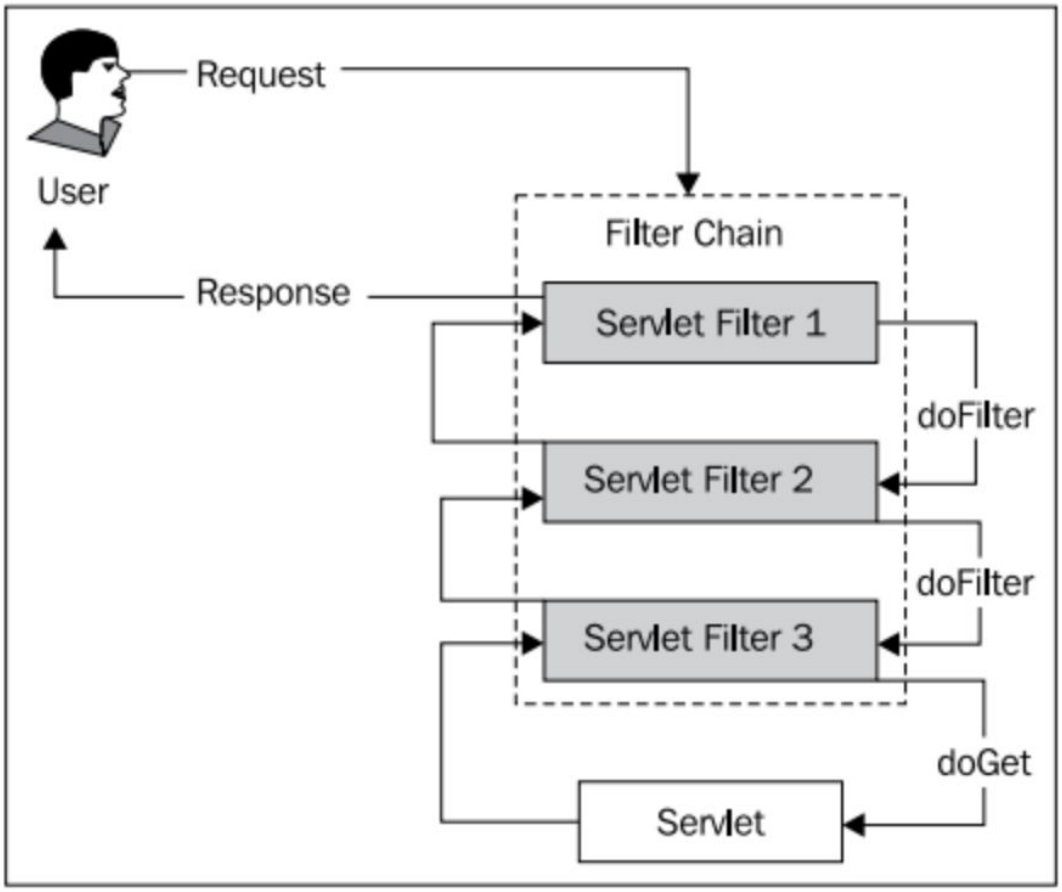
As we know, the feature of limiting to login on multiple devices with one account is a popular security mechanism to protect users from being involved in Identity theft, which is the reason we implement it in our project. Based on the framework -Spring security- we used as our server. We can easily take advantage of existing API to implement this feature, which is another reason we use spring-security as our basic framework.

Besides, we also think about adding features of code verification which protects accounts being logged by robots or attackers. We again make use of the spring security framework with authentication filters from the system. To customize the authentication features, we are supposed to construct one more filter dealing with code verification before user-password authentication.

## Overview：

1.Limited Multiple logins:
There is a onAuthentication method under package of org.springframework.security.web.authentication.session.ConcurrentSessionControlAuthenticationStrategy in springframework security which will be triggered whenever there is a login. Then according to the authentication provided by the login user, the server pulls out all sessions stored in the database, and counts how many sessions it has. Once the existing sessions are more than what we have set up. It takes advantage of a queue to edge out the first in session, and set that sesion be expire. Finally we just need to handle the function for the expired session.

2.code verification



Based on spring security, there is a systematic filterchain to authenticate the user identification. To add code verification, we first add one more filter before the user-password authentication filter on the filter chain(queue), only code being authenticated, will the identification being processed.

## API Overview:

1.Limited Multiple logins:

    1.sessionManagement()
    2.maximumSessions(1)
    3.expiredUrl("/")

2.Code verification :

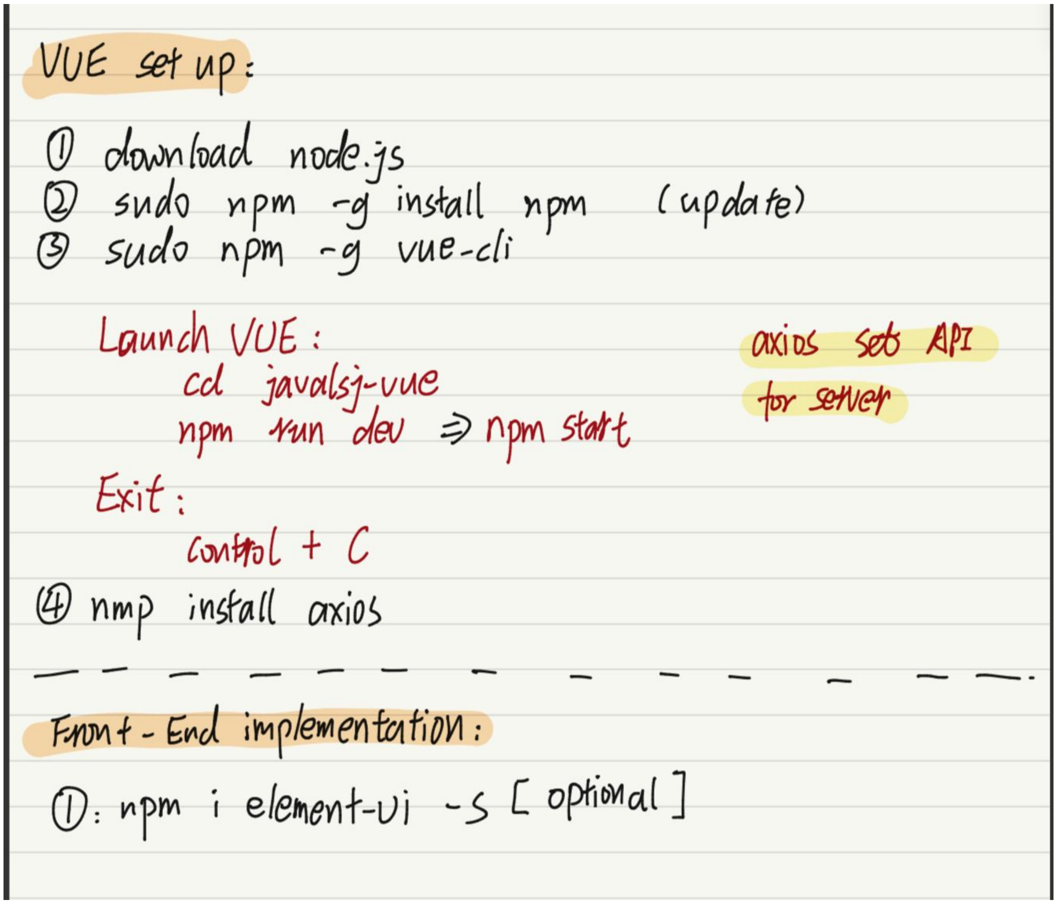
1.kaptcha to produce code
2.self define LoginAuthenticationFilter
3.loginUrlAuthenticationEntryPoint
4.authenticationFailureHandler()

## Reference:

1.cookie：
expire: check box(login free for 3 minutes, reset cookies)
2.LAN(get access to local server):https://blog.csdn.net/lamp_yang_3533/article/details/52154695
3.cdn: get location of ip
4. spring mvc - hibernate
5. websocket: long-distance login reminder(last login edge out the previous one, a dialog pop up on previous one and return to re- login) https://segmentfault.com/q/1010000007815114
6.username duplication check:

Vue skills:

1. how to dynamically add class to an element:
   https://vuejs.org/v2/guide/class-and-style.html
   https://michaelnthiessen.com/dynamically-add-class-name/
2. how to exchange information among vue components:
   https://vuex.vuejs.org/guide/
3. how to listen event change among vue components:
   eventbus:https://blog.logrocket.com/using-event-bus-in-vue-js-to-pass-data-between-components/



VUE set up:

① download node.js<br>
② sudo npm -g install npm (update)<br>
③ sudo npm -g vue-cli

Launch VUE:

cd javalsj-vue<br>
npm run dev ⇒ npm start

axios set API for server

Exit:

control + C

④ nmp install axios

Front-End implementation:

①: npm i element-ui -S [optional]

spring limit multiple devices get access:
1.https://www.cnblogs.com/zyly/p/12316099.html
2.https://sixdegree.github.io/2013/07/01/SpringSecurity.html
3.https://blog.csdn.net/elonpage/article/details/78955963

How to reset mysql when you are get denied because of forgetting password:
(I use homebrew to install mysql which has slightly different operation)

1. stop mysql service: brew services stop mysql
2. goes to related file: cd /usr/local/opt/mysql/bin
3. skip restriction: mysqld_safe --user=mysql --skip-grant-tables --skip-networking &
4. goes to mysql without password: mysql -u root mysql
5. change the password: ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
6. quirt : quit
7. Done