



**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ (İ.Ö)**

AĞ PROGRAMLAMA DERSİ PROJE ÖDEVİ

DERSİ VEREN ÖĞRETİM ÜYESİ :

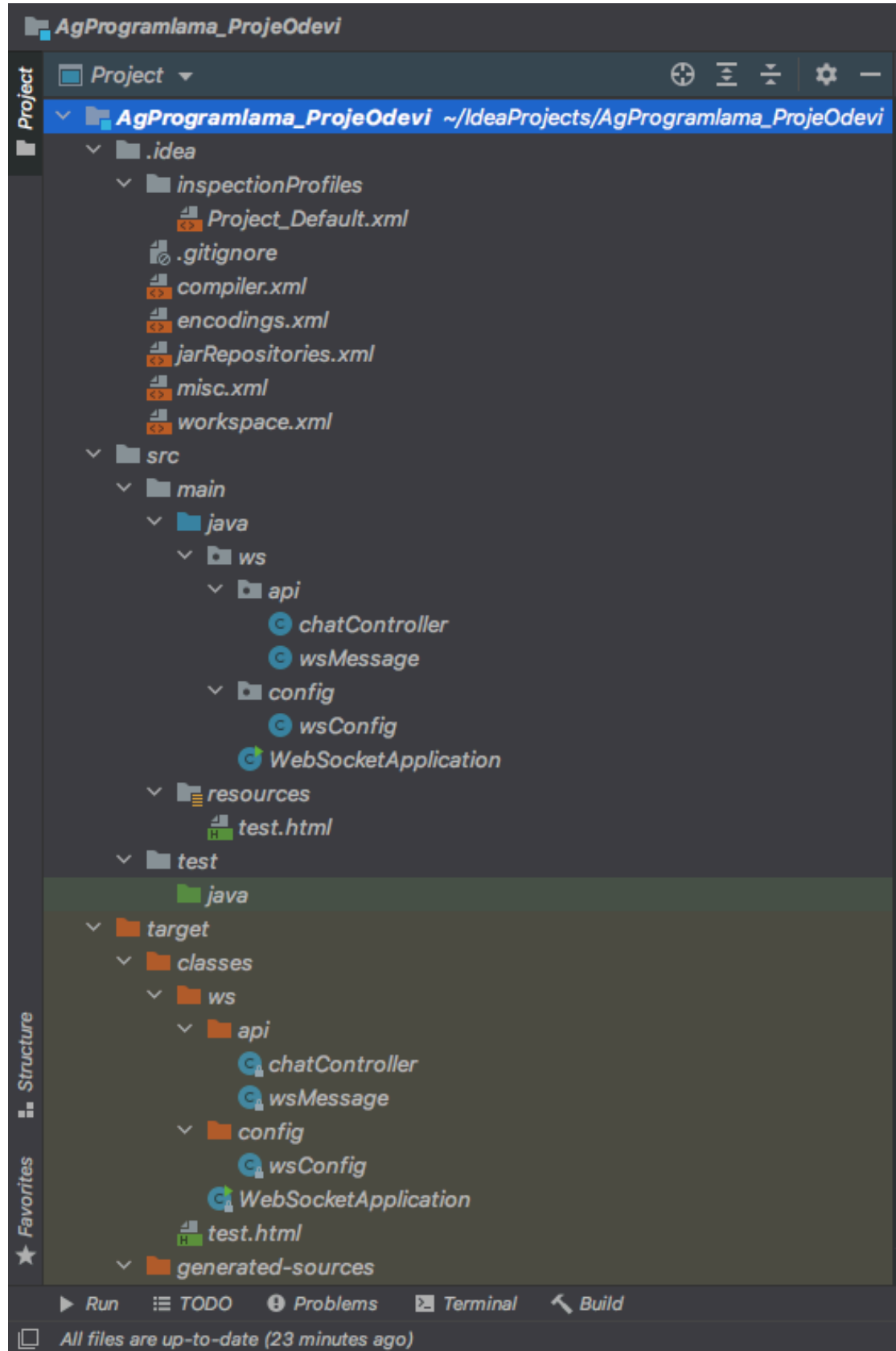
Dr. Öğr. Üyesi MUSA BALTA

ÖĞRENCİ : TUĞBA ULUSOY G171210017

SAKARYA (2021)

Projeimin Konusu: Websoket Programlama

Proje klasörümün içeriği ise:



AgProgramlama_ProjeOdevi

Project

- generated-sources
- annotations
- AgProgramlama_ProjeOdevi.iml
- pom.xml
- External Libraries
 - < 11 > /Library/Java/JavaVirtualMachines/jdk-11.0.11.jdk/Contents/Home
 - Maven: ch.qos.logback:logback-classic:1.2.3
 - Maven: ch.qos.logback:logback-core:1.2.3
 - Maven: com.fasterxml.jackson.core:jackson-annotations:2.10.1
 - Maven: com.fasterxml.jackson.core:jackson-core:2.10.1
 - Maven: com.fasterxml.jackson.core:jackson-databind:2.10.1
 - Maven: com.fasterxml.jackson.datatype:jackson-datatype-jdk8:2.10.1
 - Maven: com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.10.1
 - Maven: com.fasterxml.jackson.module:jackson-module-parameter-names:2.10.1
 - Maven: com.fasterxml:classmate:1.5.1
 - Maven: jakarta.annotation:jakarta.annotation-api:1.3.5
 - Maven: jakarta.validation:jakarta.validation-api:2.0.1
 - Maven: org.apache.logging.log4j:log4j-api:2.12.1
 - Maven: org.apache.logging.log4j:log4j-to-slf4j:2.12.1
 - Maven: org.apache.tomcat.embed:tomcat-embed-core:9.0.29
 - Maven: org.apache.tomcat.embed:tomcat-embed-el:9.0.29
 - Maven: org.apache.tomcat.embed:tomcat-embed-websocket:9.0.29
 - Maven: org.hibernate.validator:hibernate-validator:6.0.18.Final
 - Maven: org.jboss.logging:jboss-logging:3.4.1.Final
 - Maven: org.slf4j:jul-to-slf4j:1.7.29
 - Maven: org.slf4j:slf4j-api:1.7.29
 - Maven: org.springframework.boot:spring-boot:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-autoconfigure:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter-json:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter-logging:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter-tomcat:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter-validation:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter-web:2.2.2.RELEASE
 - Maven: org.springframework.boot:spring-boot-starter-websocket:2.2.2.RELEASE

Structure

Favorites

Run TODO Problems Terminal Build

All files are up-to-date (24 minutes ago)

AgProgramlama_ProjeOdevi

Project

- > Maven: com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.10.1
- > Maven: com.fasterxml.jackson.module:jackson-module-parameter-names:2.10.1
- > Maven: com.fasterxml:classmate:1.5.1
- > Maven: jakarta.annotation:jakarta.annotation-api:1.3.5
- > Maven: jakarta.validation:jakarta.validation-api:2.0.1
- > Maven: org.apache.logging.log4j:log4j-api:2.12.1
- > Maven: org.apache.logging.log4j:log4j-to-slf4j:2.12.1
- > Maven: org.apache.tomcat.embed:tomcat-embed-core:9.0.29
- > Maven: org.apache.tomcat.embed:tomcat-embed-el:9.0.29
- > Maven: org.apache.tomcat.embed:tomcat-embed-websocket:9.0.29
- > Maven: org.hibernate.validator:hibernate-validator:6.0.18.Final
- > Maven: org.jboss.logging:jboss-logging:3.4.1.Final
- > Maven: org.slf4j:jul-to-slf4j:1.7.29
- > Maven: org.slf4j:slf4j-api:1.7.29
- > Maven: org.springframework.boot:spring-boot:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-autoconfigure:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter-json:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter-logging:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter-tomcat:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter-validation:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter-web:2.2.2.RELEASE
- > Maven: org.springframework.boot:spring-boot-starter-websocket:2.2.2.RELEASE
- > Maven: org.springframework:spring-aop:5.2.2.RELEASE
- > Maven: org.springframework:spring-beans:5.2.2.RELEASE
- > Maven: org.springframework:spring-context:5.2.2.RELEASE
- > Maven: org.springframework:spring-core:5.2.2.RELEASE
- > Maven: org.springframework:spring-expression:5.2.2.RELEASE
- > Maven: org.springframework:spring-jcl:5.2.2.RELEASE
- > Maven: org.springframework:spring-messaging:5.2.2.RELEASE
- > Maven: org.springframework:spring-web:5.2.2.RELEASE
- > Maven: org.springframework:spring-webmvc:5.2.2.RELEASE
- > Maven: org.springframework:spring-websocket:5.2.2.RELEASE
- > Maven: org.yaml:snakeyaml:1.25

Scratches and Consoles

Run TODO Problems Terminal Build

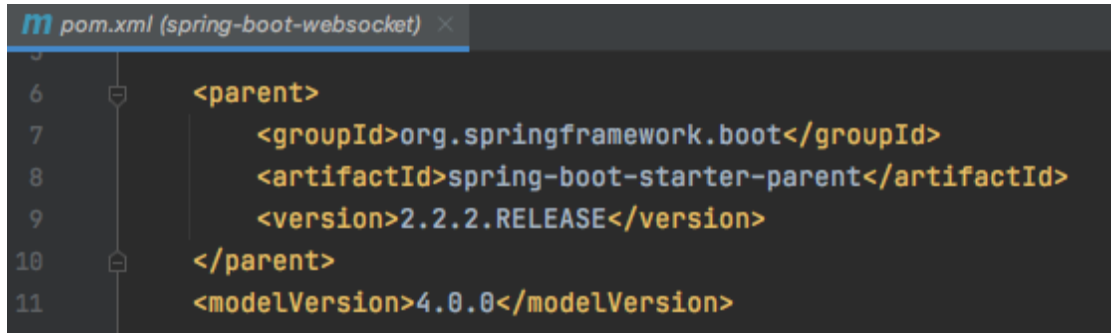
All files are up-to-date (31 minutes ago)

Websoket programlama için “ws” kütüphanesini seçtim. Package ws.api ve package ws; kullandım. Spring Boot yapısını kullanabilmek için springframework’leri import ettim.

Kullandığım Teknoloji ve eklentiler;

- Java JDK 11
- IntelliJ IDEA CE
- Maven
- “ws” websoket kütüphanesi
- Spring Boot
- SockJS
- Stomp

Spring-boot altyapısı bize birtakım çalıştırılabilir kolaylıklar sunuyor.

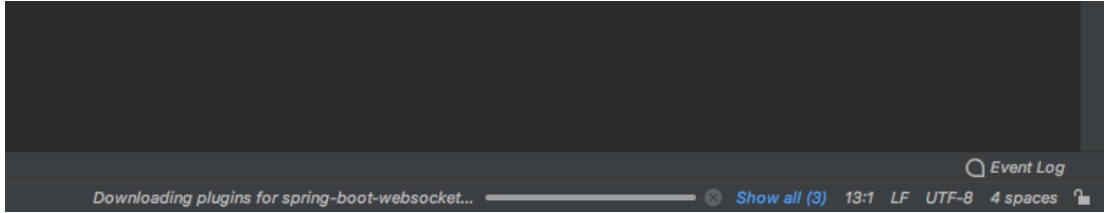


Şekil: “Paketleme” ve “Version Yönetimi” için projemin pom.xml ‘inde yazdığım kod bloğum.



Şekil: pom.xml’de Jackson’ın üzerinde birtakım değişiklikler yapabilmek için yazdığım kod bloğum

Proje ödevimi yaparken ara sıra hatalar alıyordum bu sebeple gerekli tüm pluginleri indirdim. Böylelikle sorunlarım çözüldü.



“WebSocketApplication.java” class ımda aşağıdaki kod bloğunu yazdım.

```
public static void main(String[] args){  
    SpringApplication.run(WebSocketApplication.class, args);  
}
```

Şekil: Spring boot uygulamamı başlatmak için kullandığım kod bloğum

Konfigürasyonlarımı yazdığım “wsCongig.java” class’ ı oluşturdum. Bunun içinde iki fonksiyonu override ettim.

1.Fonksiyonumun Açıklaması:

Client ‘lar tarafında Endpoint(“/chat”) üzerinden konuşuyorum. Mesajı bu Endpoint’e yolluyorum ve yine mesajı buradan dinliyorum. setAllowedOrigins(“*”) klasör içinde nereden bu Endpoint çağırılırsa çağırılsın cevap verilebilsin diye “*” koydum. Buna bağlanırken “withSockJS()” kullandım.

```
public class wsConfig implements WebSocketMessageBrokerConfigurer {  
    @Override  
    public void registerStompEndpoints(StompEndpointRegistry registry) {  
        registry.addEndpoint( ...strings: "/chat").setAllowedOrigins("*").withSockJS();  
    }  
}
```

Şekil: (Server) Sunucumun dinlemesi için yazdığım kod bloğum.

2.Fonksiyonumun Açıklaması:

```
@Override
public void configureMessageBroker(MessageBrokerRegistry registry){
    registry.enableSimpleBroker( ...destinationPrefixes: "/topic");
}
```

Şekil: (Client) İstemcimin dinlemesi için yazdığım kod bloğum. “Broker” distribution ‘dan sorumlu.

```
chatController.java x
1 package ws.api;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.messaging.handler.annotation.MessageMapping;
5 import org.springframework.messaging.handler.annotation.Payload;
6 import org.springframework.messaging.simp.SimpMessagingTemplate;
7 import org.springframework.stereotype.Controller;
8 import org.springframework.web.bind.annotation.CrossOrigin;
9
10 @Controller
11 @CrossOrigin
12 public class chatController {
```

Şekil: Gelen isteklerin tümünü izinlendirmek için @CrossOrigin kullandım.

Bu mesajı arayüz bize göndericek biz de bu mesajı ya herkese ya da ilgili bir kişiye yollayacağız. Toplu chat uygulaması ise herkese göndermemiz gerekiyor. Kullanıcı özel mesaj gönderiyorsa sadece ilgili kullanıcıya mesajı göndermemiz gerekiyor.

Gelen mesajı herkese göndermek istiyorsak: “@SendTo” kullanıyoruz.

```
@MessageMapping("/chat")
@SendTo("/topic")
public void chatEndpoint(@Payload wsMessage ws_message){

}
```

Şekil: Gelen bütün mesajlar tüm kullanıcılara gidiyor.

Eğer mesajı spesifik bir kullanıcıya gönderiyorsak “@SendToUser” kullanıyoruz.


```

pu String[] value() default {}
String[] destinations() default {}
boolean broadcast() default true
//@SendTo("/topic")
@SendToUser()
public void chatEndpoint(@Payload wsMessage ws_message){
}

```

Şekil: Mesajı spesifik bir kullanıcıya göndermek istiyorsak

- 1) Value 'yi vereceğiz
- 2) Nereye gideceğini yazacağız. Yani kullanıcıyı vereceğiz burada.
- 3) Broadcast' i seçersek otomatik mesaj herkese gider.

Arayüzden biri Endpoint'e mesaj gönderdiğinde onu önce konsola yazıyorum. Sonra **client'** lara yolluyorum.

```

@Autowired
private SimpMessagingTemplate messagingTemplate;

@MessageMapping("/chat")
//@SendTo("/topic")
//@SendToUser()
public void chatEndpoint(@Payload wsMessage ws_message){
    System.out.println(ws_message);
    messagingTemplate.convertAndSend( destination: "/topic", ws_message);
}

```

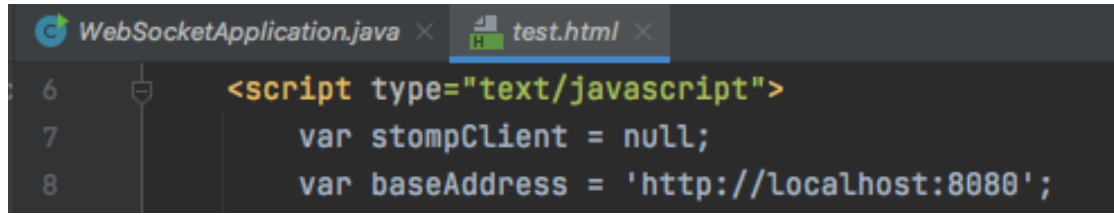
Şekil: Bir nevi herkesin herkese mesaj gönderebildiği bir grup kanalı implementasyonu.

```

WebSocketApplication.java x test.html x
1 <html>
2 <head>
3 <title>Chat WebSocket</title>
4 <script src="https://cdnjs.cloudflare.com/ajax/libs/sockjs-client/1.4.0/sockjs.js"></script>
5 <script src="https://cdnjs.cloudflare.com/ajax/libs/stomp.js/2.3.3/stomp.js"></script>


```

Şekil: Sockjs ve Stomp protokollerini kullandım.



```
WebSocketApplication.java x test.html x
6 <script type="text/javascript">
7   var stompClient = null;
8   var baseAddress = 'http://localhost:8080';
```

Şekil: Bizim Endpoint'imizin hangi path'ten yayınlandığını yazdım. http üzerinden register oluyorum chat' e.



```
test.html x
17 function connect() {
18   var socket = new SockJS(baseAddress + '/chat');
19   stompClient = Stomp.over(socket);
20   stompClient.connect({}, function(frame) {
21     setConnected(true);
22     console.log('Connected: ' + frame);
23     stompClient.subscribe('/topic', function (message) {
24       handleReceivedMessage(JSON.parse(message.body));
25     });
26   });
27 }
```

Şekil: Bağlanma fonksiyonum

Kod bloğumun açıklaması

“/chat” Endpoint'imize bir socket oluşturdum. stompClient üzerinden mesajları gönderdim.

stompClient kullanarak “/topic” kanalına da üye olmuş oldum.