

차세대 챗봇 Lab I

Lecture-6 SpringBoot Chatbot Basic

Sungja Choi,
School of Computing, Gachon University
2023-1



학습내용

RESTful 지원을 위한 JPA,에 대해 학습합니다.

WebSocket

STOMP(Simple Text Oriented Messaging Protocol)

SockJS

Lecture-6. SpringBoot Chatbot Basic

Building a chat application with Spring Boot and WebSocket

Practice

>> **SockJS** : 웹소켓을 지원하지 않는 브라우저를 위한 자바스크립트 라이브러리

>> **STOMP** :
Publisher(송신자)-Subscriber(수신자)를 지정하고, 메시지 브로커를 통해 특정 사용자에게만 메시지를 전송하는 기능이 가능하다. 메시지 브로커는 Publisher로부터 전달받은 메시지를 Subscriber로 전달해주는 중간 역할을 수행한다.

[REF.] WebSocket

스프링 4.x부터는 WebSocket 을 아래와 같이 지원한다.

- 메시지 전송과 수신을 위한 하위 레벨 API
- 스프링 MVC 컨트롤러에서의 메시지 처리를 위한 상위 레벨 API
- 메시지 전송을 위한 메시징 템플릿
- 브라우저, 서버, 프록시에서 웹 소켓 지원 부족을 지원하기 위한 SockJS

웹 소켓만으로 메시지의 수신대상과 메시지에 대한 정보를 표현할 수 없음



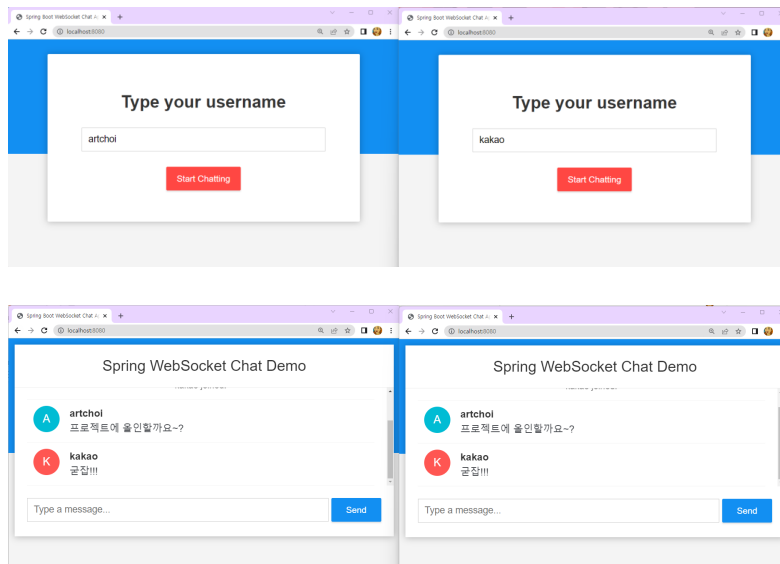
“STOMP 프로토콜로 해결”

Practice

실습-1 챗봇 만들기

스프링 부트 기반의 웹 소켓을 활용하여 챗 서비스를 제공합니다.

[실행화면]



Practice

>> start.spring.io

>> WebSocket

의존성 라이브러리 추가

>> WebSocket-demo.zip을
다운로드 및 압축해제

프로젝트 생성

The screenshot shows the Spring Initializr web application interface. The browser address bar displays 'start.spring.io'. The page features a sidebar with a hamburger menu and a settings icon. The main content area is divided into several sections:

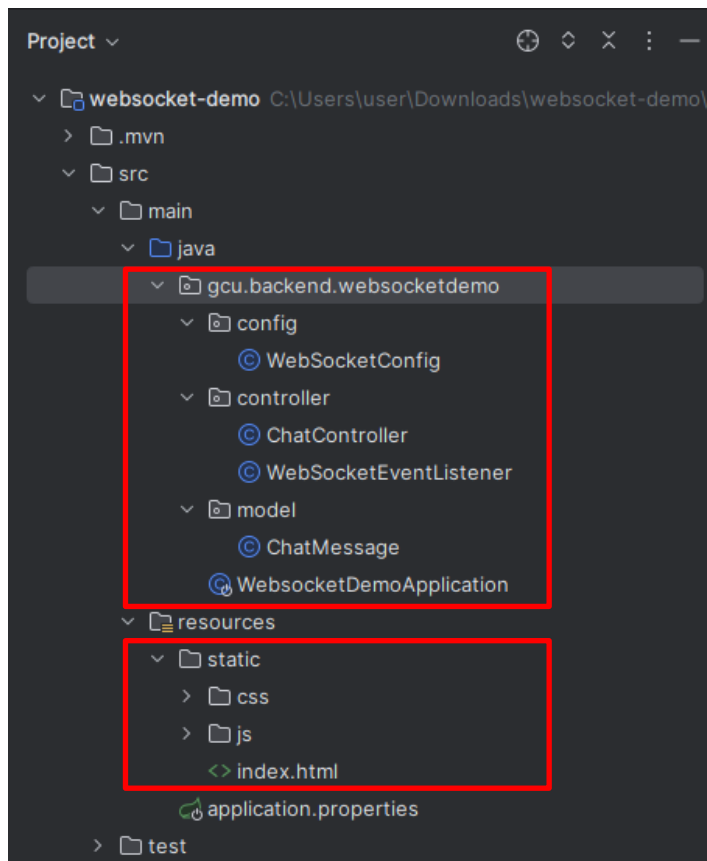
- Project:** Includes radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', and 'Maven' (selected).
- Language:** Includes radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'.
- Spring Boot:** Includes radio buttons for '3.1.0 (SNAPSHOT)', '3.1.0 (M2)', '3.0.6 (SNAPSHOT)', '3.0.5' (selected), '2.7.11 (SNAPSHOT)', and '2.7.10'.
- Project Metadata:** Includes input fields for 'Group' (gcu.backend), 'Artifact' (websocket-demo), 'Name' (websocket-demo), 'Description' (Demo project for Spring Boot), and 'Package name' (gcu.backend.websocket-demo). There are also radio buttons for 'Packaging' (Jar selected, War unselected).
- Dependencies:** Includes a button 'ADD DEPENDENCIES... CTRL + B' and a section for 'WebSocket MESSAGING' with the description 'Build Servlet-based WebSocket applications with SockJS and STOMP.' This section is highlighted with a red box.

At the bottom of the page, there are three buttons: 'GENERATE CTRL + G' (highlighted with a red box), 'EXPLORE CTRL + SPACE', and 'SHARE...'. Social media icons for GitHub and Twitter are visible in the bottom left corner.

Practice

>> **static** 폴더 부분은
프론트 엔드 부분의 코드

프로젝트 구조



Practice

챗 메시지 모델정의

Model/ChatMessage.java

```
@Data
public class ChatMessage {
    private MessageType type;
    private String content;
    private String sender;

    public enum MessageType {
        CHAT,
        JOIN,
        LEAVE
    }
}
```


Practice

❶

@EnableWebSocketMessageBroker

웹 소켓 기능을 활성화

❷ sockJs클라이언트가

웹소켓 연결을 위한

endpoint 지정

❸ 메시지 보낼 때 관련

경로 설정

클라이언트가 메시지를

보낼 때 경로 맨 앞에

"/app"이 붙어있으면

Broker로 보내짐

❹ 메시지 받을 때 관련

경로설정

웹 소켓 리스너 정의

config/WebSocketConfig.java

@Configuration

@EnableWebSocketMessageBroker ❶

public class WebSocketConfig implements
WebSocketMessageBrokerConfigurer {

@Override

public void registerStompEndpoints(StompEndpointRegistry registry) {
 registry.addEndpoint("/ws").withSockJS(); ❷
}

@Override

public void configureMessageBroker(MessageBrokerRegistry registry) {
 registry.setApplicationDestinationPrefixes("/app"); ❸
 registry.enableSimpleBroker("/topic"); ❹
}

Practice

- ❶ 목적지가 path와 일치하는 메시지를 수신했을 경우 해당 메서드(sendMessage)를 호출
- ❷ Payload는 chatMessage객체에 바인딩
- ❸ 세션정보에 사용자 이름을 넣어서 보냄

챗 컨트롤러 정의

@Controller

controller/ChatController.java

```
public class ChatController {  
    @RequestMapping("/chat.sendMessage") ❶  
    @SendTo("/topic/public") ❷  
    public ChatMessage sendMessage(@Payload ChatMessage chatMessage)  
    { return chatMessage; }  
  
    @RequestMapping("/chat.addUser")  
    @SendTo("/topic/public")  
    public ChatMessage addUser(@Payload ChatMessage chatMessage,  
        SimpMessageHeaderAccessor headerAccessor) {  
        // Add username in web socket session  
        headerAccessor.getSessionAttributes().put("username",  
            chatMessage.getSender()); ❸  
  
        return chatMessage;  
    }  
}
```

Practice

- ① 해당 클래스에 대한 로그정보를 남기도록 한다
- ② 웹 소켓 컨넥션 이벤트를 기다리는 리스너를 정의

웹 소켓 리스너 정의

Controller/WebSocketEventListener.java

@Component

```
public class WebSocketEventListener {
```

```
    ① private static final Logger logger =  
    LoggerFactory.getLogger(WebSocketEventListener.class);
```

@Autowired

```
private SimpMessageSendingOperations messagingTemplate;
```

@EventListener ②

```
public void handleWebSocketConnectListener(SessionConnectedEvent  
event) {  
    logger.info("Received a new web socket connection");  
}
```

Practice

- ① 소켓 접속 종료를 기다리는 리스너
- ② 메시지 타입을 LEAVE로 설정하고 /topic/public 경로로 보냄

웹 소켓 리스너 정의

Controller/WebSocketEventListener.java

```
@EventListener ①
public void handleWebSocketDisconnectListener(SessionDisconnectEvent
event) {
    StompHeaderAccessor headerAccessor =
    StompHeaderAccessor.wrap(event.getMessage());

    String username = (String)
    headerAccessor.getSessionAttributes().get("username");
    if(username != null) {
        logger.info("User Disconnected : " + username);

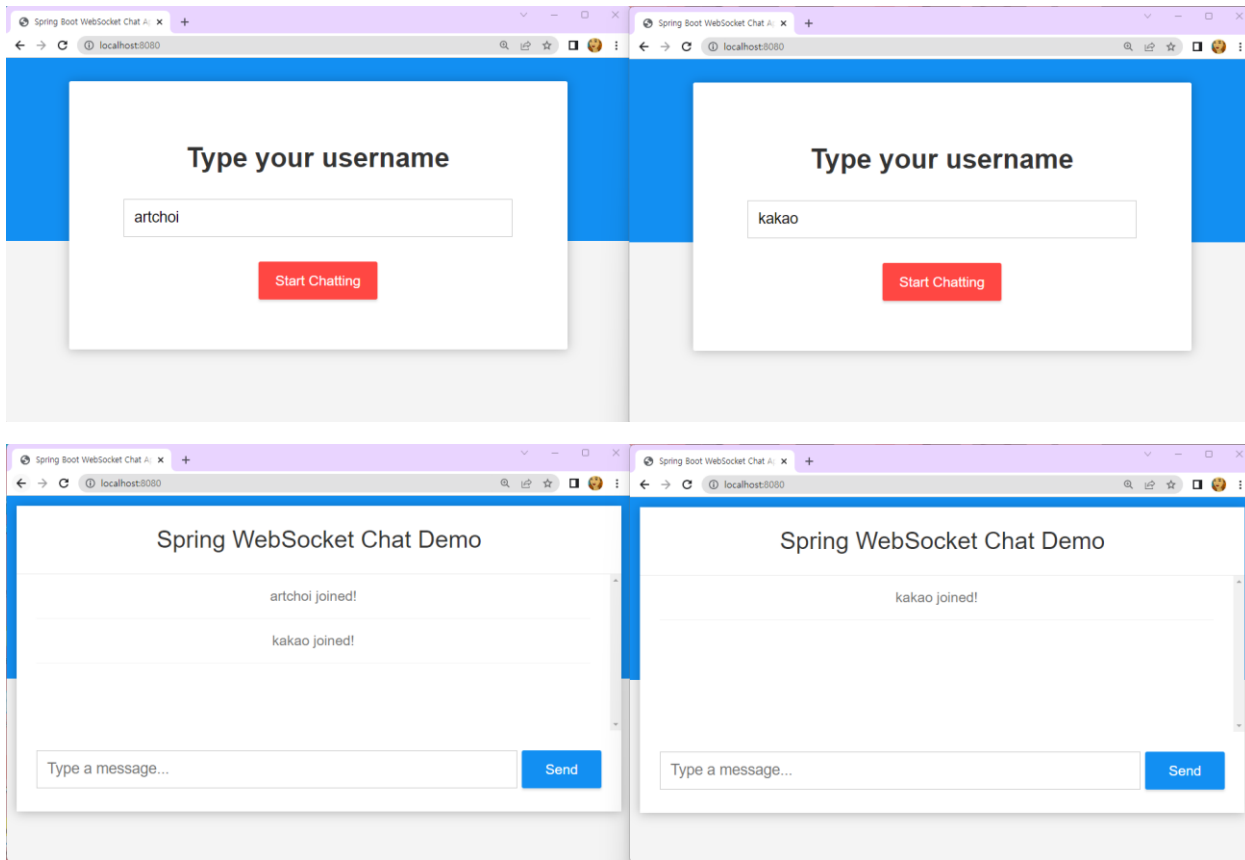
        ChatMessage chatMessage = new ChatMessage();
        chatMessage.setType(ChatMessage.MessageType.LEAVE);
        chatMessage.setSender(username);

        ② messagingTemplate.convertAndSend("/topic/public", chatMessage);
    } }
```

Practice

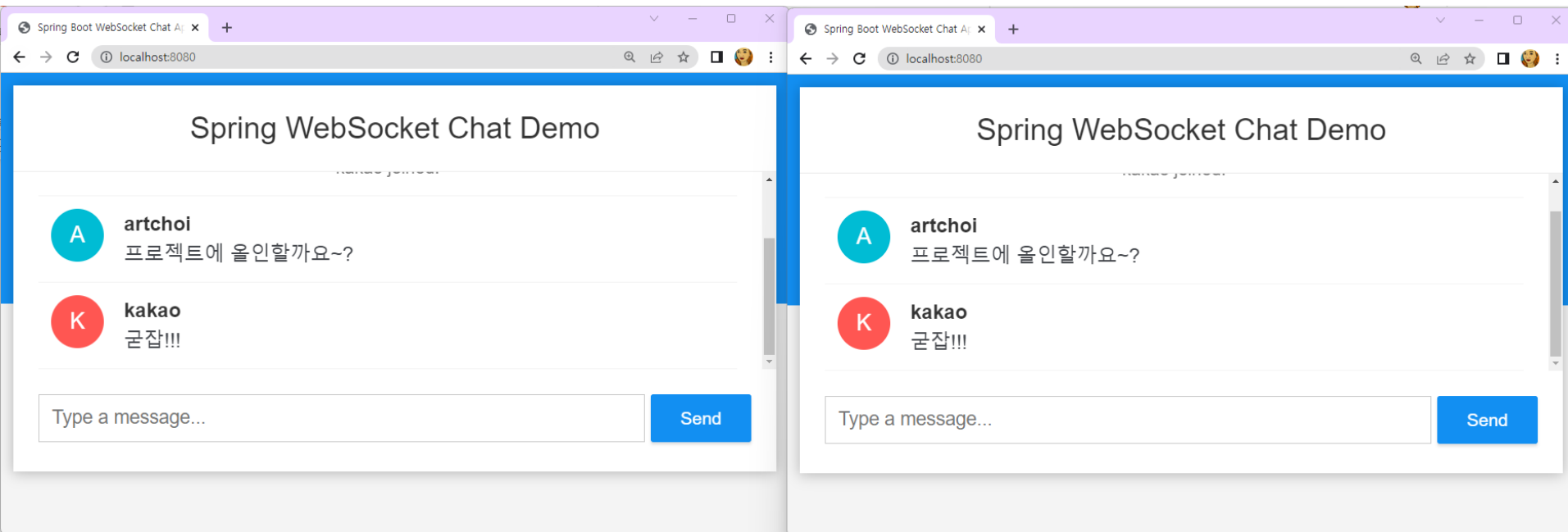
- 프론트 엔드는
사이버캠퍼스의 static.zip
파일을 압축해제해서
사용합니다.

실행화면



Practice

실행화면

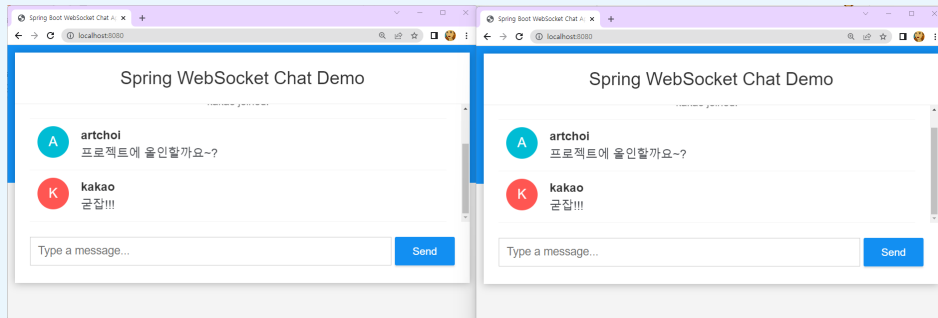


수업시간 내 진행하시기 바랍니다.

챗봇 프로토타입 제작

“요구분석엔지니어링을 기반으로 제작해봅니다”

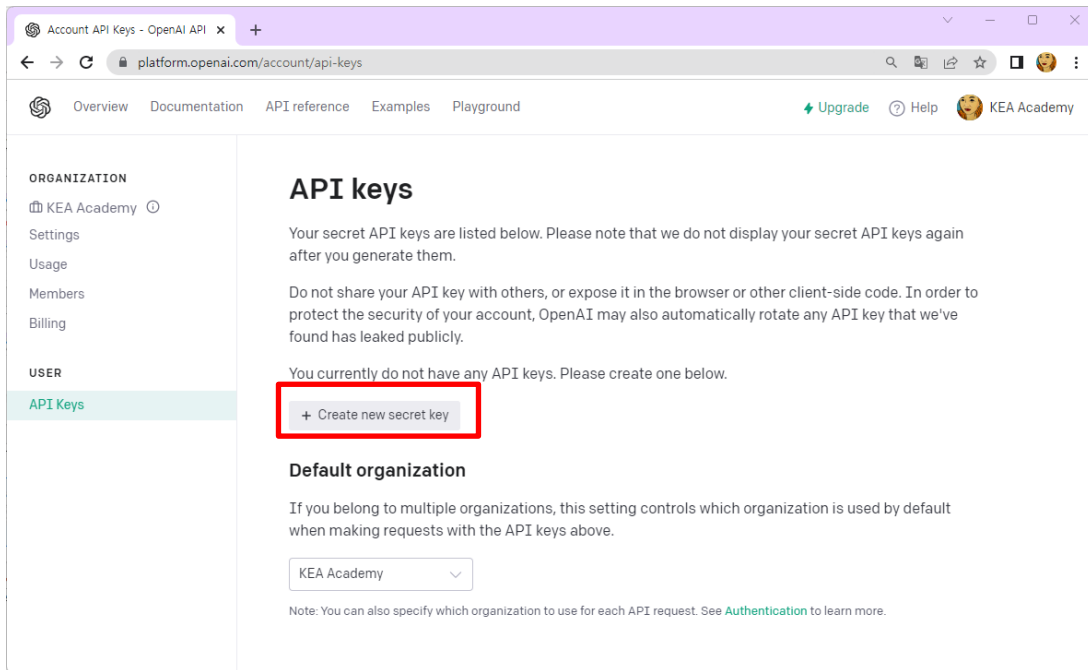
- 최종 프로젝트에서 활용할 챗봇 구현
- 언어는 무관 (파이썬도 가능)
- 7주차까지 진행
- 팀별 제출
- 평가(10점)



Practice

[REF .] ChatGPT API를 스프링프레임워크 사용

1. API Key 발급받기



Account API Keys - OpenAI API

platform.openai.com/account/api-keys

Overview Documentation API reference Examples Playground Upgrade Help KEA Academy

ORGANIZATION

- KEA Academy
- Settings
- Usage
- Members
- Billing

USER

- API Keys**

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

You currently do not have any API keys. Please create one below.

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

KEA Academy

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

Practice

[REF .] Spring Boot와 ChatGPT

2. ChatGptConfig

```
public class ChatGptConfig {  
    public static final String AUTHORIZATION = "Authorization";  
    public static final String BEARER = "Bearer ";  
    public static final String API_KEY = "Your-API-Key";  
    public static final String MODEL = "text-davinci-003";  
    public static final Integer MAX_TOKEN = 300;  
    public static final Double TEMPERATURE = 0.0;  
    public static final Double TOP_P = 1.0;  
    public static final String MEDIA_TYPE = "application/json; charset=UTF-8";  
    public static final String URL = "https://api.openai.com/v1/completions";  
}
```

CONNECT.

SOLVE.

CREATE. 