
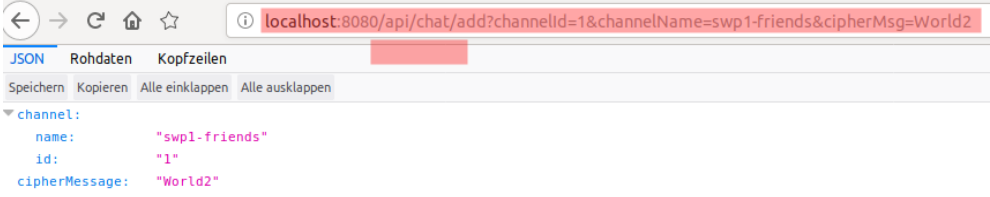
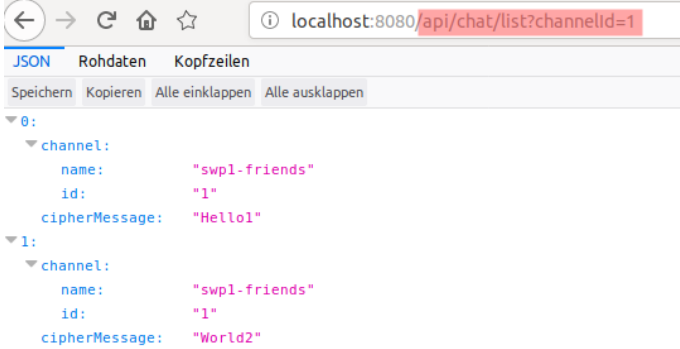


Fach/Klasse	SPW1	5JG
Themenbereich	6. Webtechnologien	
Bsp08 Webchat	<p>Erstelle eine SpringBoot Applikation für einen einfachen Chat. Die Applikation umfasst</p> <ul style="list-style-type: none"> • ein RESTful API, • einen Ajax/AngularJS Client (welcher das RESTful API verwendet) • Verschlüsselung der Nachrichten auf dem Server • Speicherung der Daten in einer Mongo-DB • und eine WebSocket-Komponente für aktive Benachrichtigungen an den Client/Browser. <p>Im Ordner bsp08_webchat befindet sich bereits die initiale Konfiguration für die Applikation.</p>	
I) RESTful-API	<p>Erzeuge ein RESTful Api Controller mit folgenden Schnittstellen:</p> <ul style="list-style-type: none"> • Neue Chat-Nachricht erstellen  <ul style="list-style-type: none"> • Liste aller Chatnachrichten für eine Channel-Id:  <p>Als Hilfestellung kann das Tutorial unter https://spring.io/guides/gs/rest-service/ verwendet werden.</p> <p>Speichere die Nachrichten als java.util.Map ab:</p> <pre>Map<String, List<Message>> channelMessages = new HashMap<>();</pre> <p>Verwende als Key für die Map die ChannelId.</p>	
II) Client	Programmiere einen Ajax oder AngularJS Client, welcher auf die RESTful-API zugreift.	
III) Verschlüsselung	Die Chatnachrichten sollen nur verschlüsselt abgelegt werden. Verwende dazu den AES-Algorithmus (mit der Übergabe des Keys vom Client zum Server) damit die Nachrichten verschlüsselt werden.	

IV) DB-Speicherung	Speichere die Daten in einer Mongo-DB ab.	
V) Websocket	<p>Erstelle einen Websocket-Client, mit welchem neue Nachrichten direkt vom Server an den Browser übermittelt werden.</p> <p>Verwende dazu die zwei Tutorials:</p> <p>https://spring.io/guides/gs/messaging-stomp-websocket/</p> <p>https://www.baeldung.com/websockets-spring</p>	