

Programování v Javě II

Frameworky

Michal Pobucký

michal.pobucky@fpf.slu.cz

Ústav informatiky – léto 2020

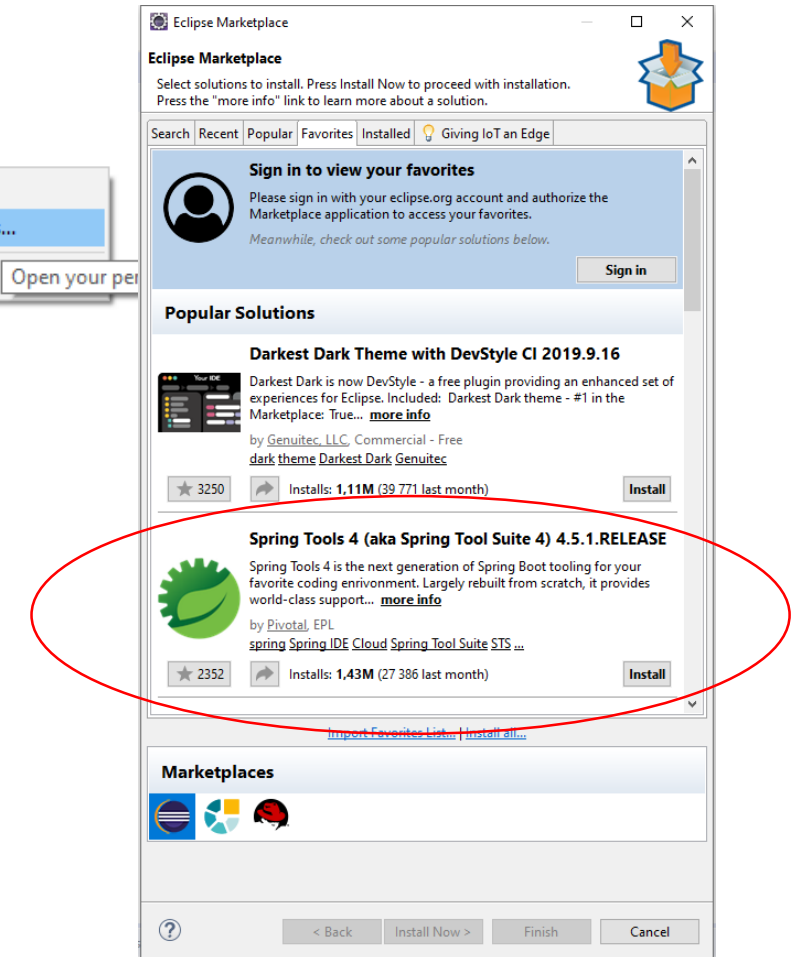
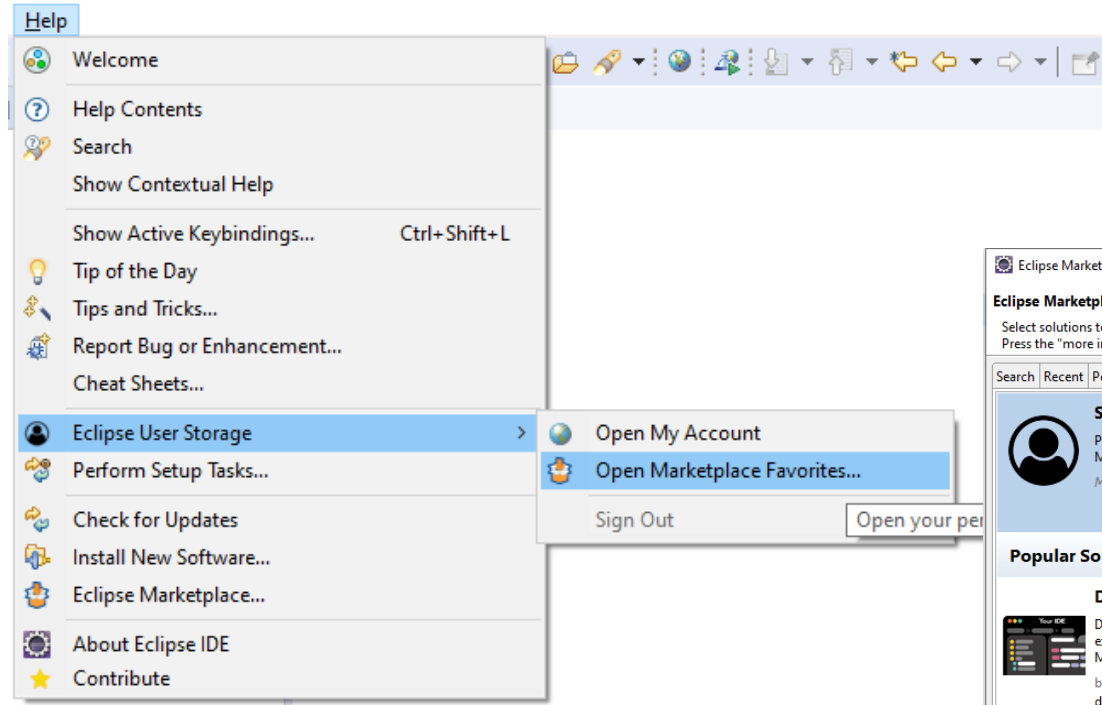


Framework

- Aplikační struktura, která pomáhá při programování a vývoji softwarových projektů
- Obsahuje podprogramy, knihovny, návrhové vzory, ...
- Například:
 - JavaServer Faces
 - Spring
 - Struts
 - Hibernate
 - Apache Wicket

Spring

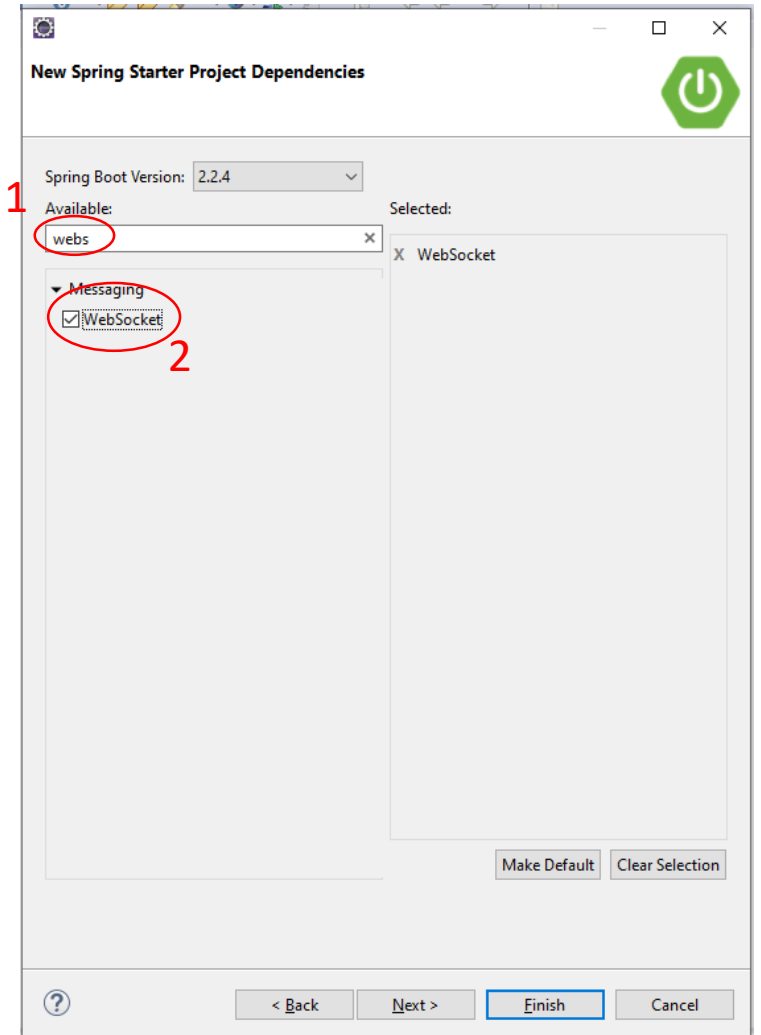
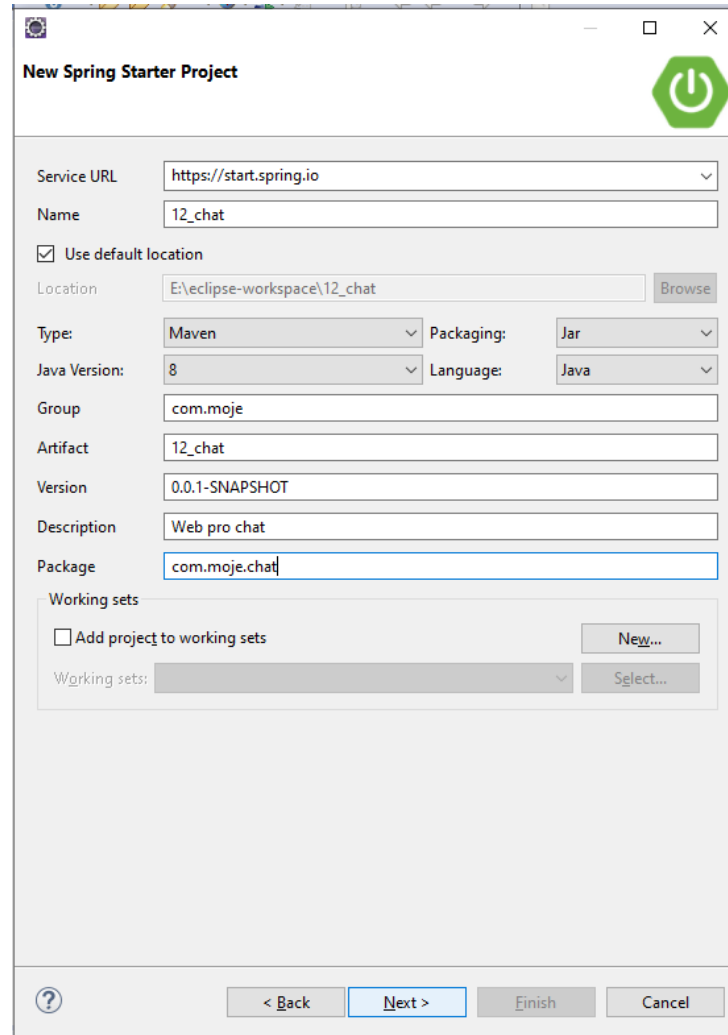
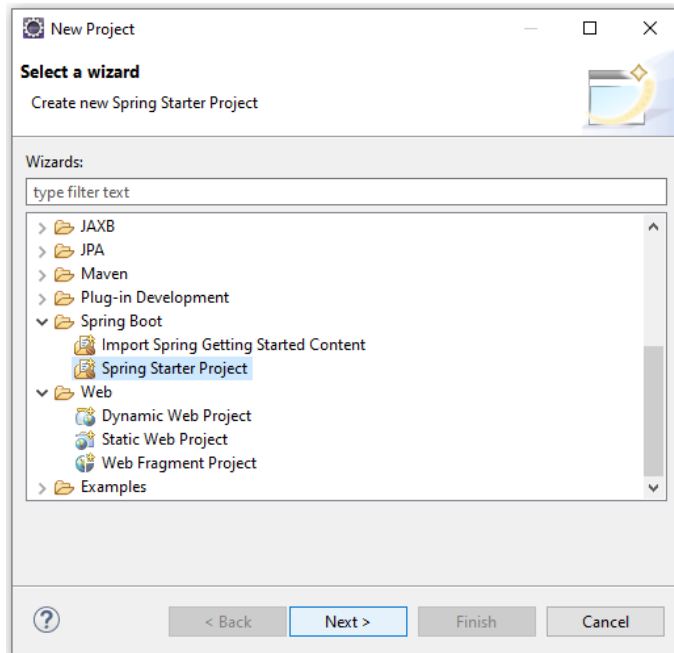
- Tools for Eclipse
- Spring Tools 4
- Spring Eclipse



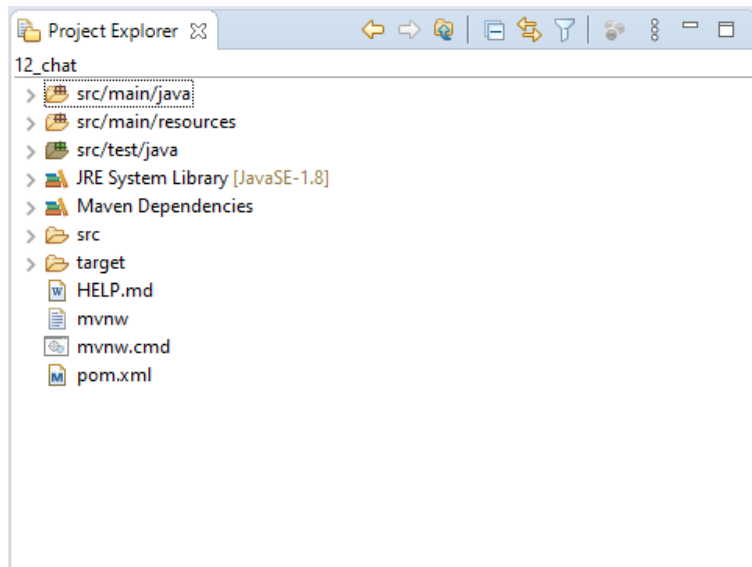
Spring

- Program
 - Studentský chat, každý klient se přihlásí do systému a může konverzovat s ostatními online v reálném čase

Spring



Spring

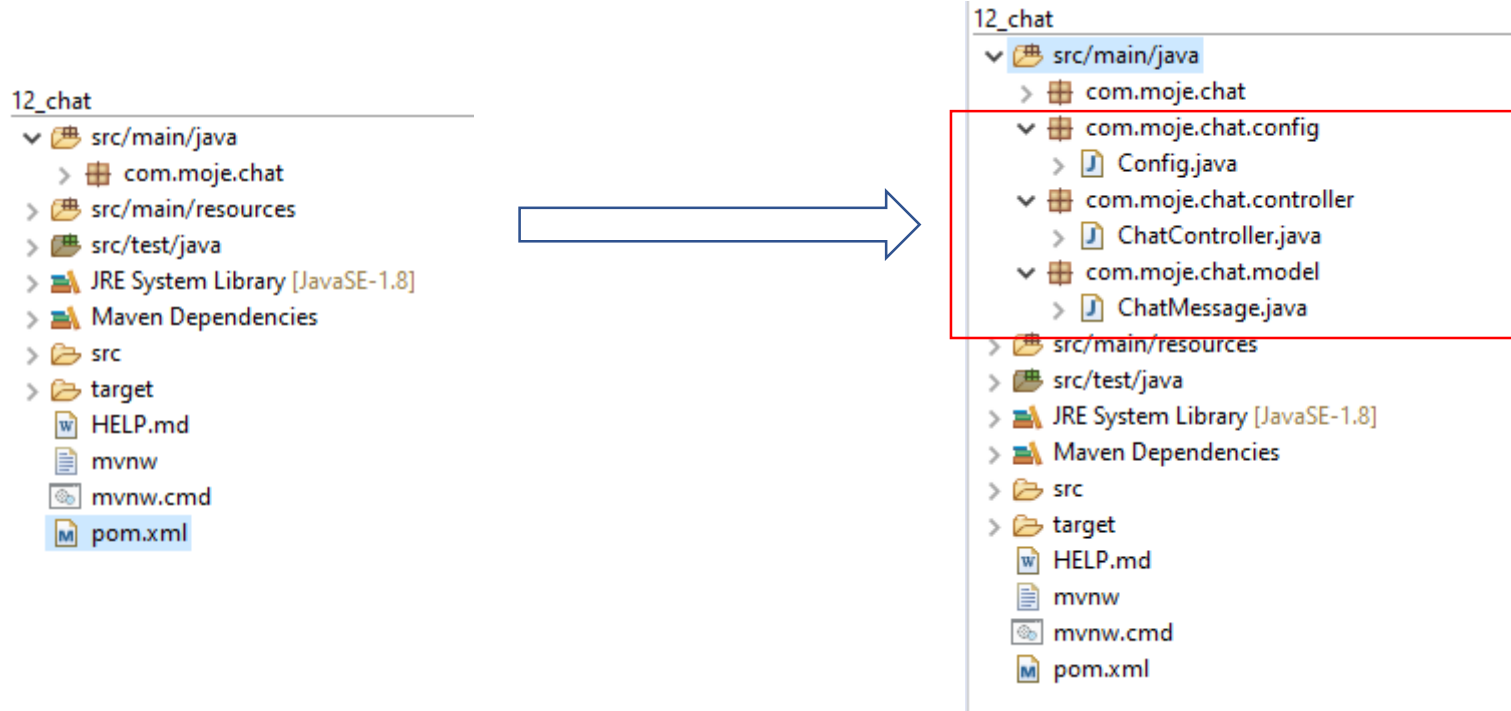


Spring

- Pom.xml

```
20
21- <dependencies>
22-   <dependency>
23     <groupId>org.springframework.boot</groupId>
24     <artifactId>spring-boot-starter-websocket</artifactId>
25   </dependency>
26
27-   <dependency>
28     <groupId>org.springframework.boot</groupId>
29     <artifactId>spring-boot-starter-reactor-netty</artifactId>
30   </dependency>
31
32-   <dependency>
33     <groupId>org.springframework.boot</groupId>
34     <artifactId>spring-boot-starter-test</artifactId>
35     <scope>test</scope>
36-   <exclusions>
37-     <exclusion>
```

Spring



Spring

- Config.java

```
1 package com.moje.chat.config;
2
3 import org.springframework.context.annotation.Configuration;
4 import org.springframework.messaging.simp.config.MessageBrokerRegistry;
5 import org.springframework.web.socket.config.annotation.EnableWebSocketMessageBroker;
6 import org.springframework.web.socket.config.annotation.StompEndpointRegistry;
7 import org.springframework.web.socket.config.annotation.WebSocketMessageBrokerConfigurer;
8
9 @Configuration
10 @EnableWebSocketMessageBroker
11 public class Config implements WebSocketMessageBrokerConfigurer {
12
13     @Override
14     public void registerStompEndpoints(StompEndpointRegistry registry) {
15         //WebSocketMessageBrokerConfigurer.super.registerStompEndpoints(registry);
16         registry.addEndpoint("/chat").withSockJS();
17     }
18
19     @Override
20     public void configureMessageBroker(MessageBrokerRegistry registry) {
21         //WebSocketMessageBrokerConfigurer.super.configureMessageBroker(registry);
22         registry.enableSimpleBroker("/topic");
23         registry.setApplicationDestinationPrefixes("/app");
24     }
25 }
```

Spring

- ChatMessage.java

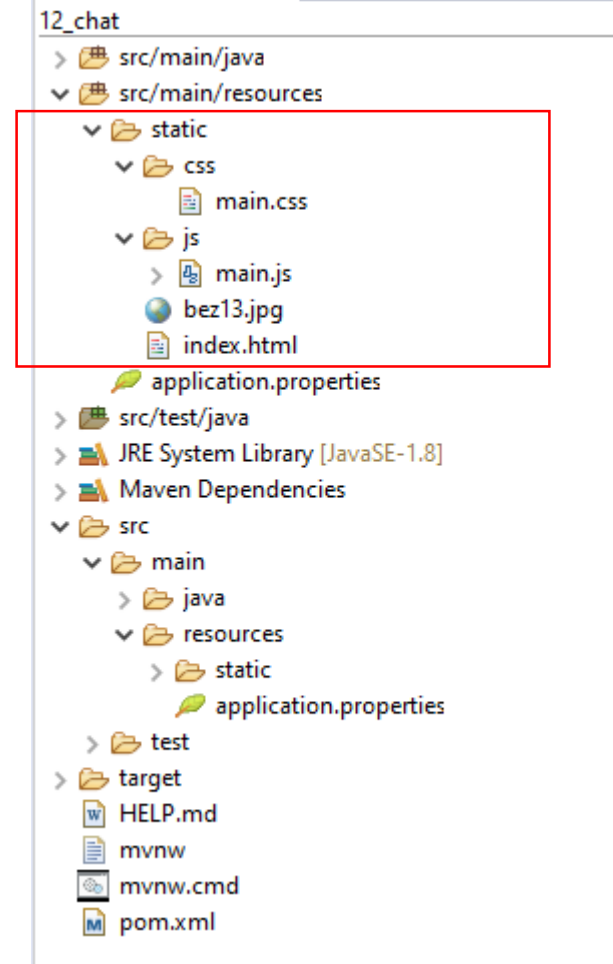
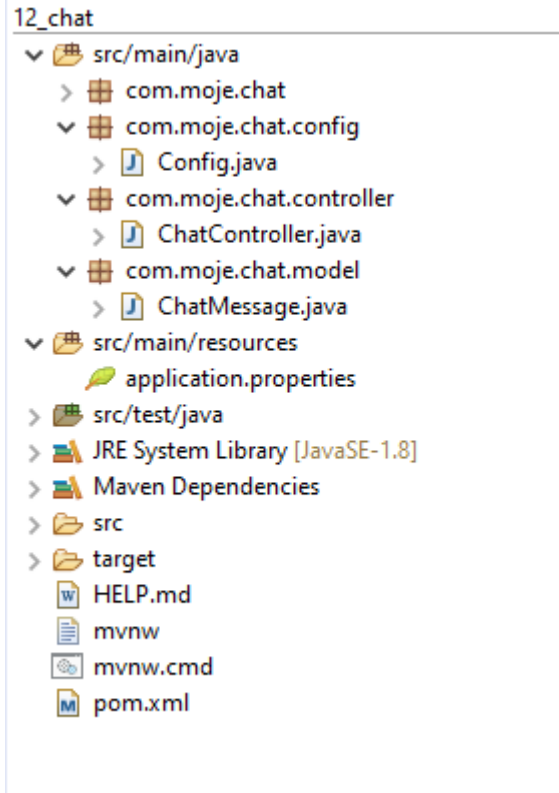
```
1 package com.moje.chat.model;
2
3 public class ChatMessage {
4     private String content;
5     private String sender;
6     private MessageType type;
7
8     public String getContent() {
9         return content;
10    }
11
12    public void setContent(String content) {
13        this.content = content;
14    }
15
16    public String getSender() {
17        return sender;
18    }
19
20    public void setSender(String sender) {
21        this.sender = sender;
22    }
23
24    public MessageType getType() {
25        return type;
26    }
27
28    public void setType(MessageType type) {
29        this.type = type;
30    }
31
32    public enum MessageType{
33        CHAT, LEAVE, JOIN
34    }
35 }
```

Spring

- ChatController.java

```
1 package com.moje.chat.controller;
2
3 import org.springframework.messaging.handler.annotation.MessageMapping;
4 import org.springframework.messaging.handler.annotation.Payload;
5 import org.springframework.messaging.handler.annotation.SendTo;
6 import org.springframework.messaging.simp.SimpMessageHeaderAccessor;
7 import org.springframework.stereotype.Controller;
8 import com.moje.chat.model.ChatMessage;
9
10 @Controller
11 public class ChatController {
12
13     @MessageMapping("/chat.register")
14     @SendTo("/topic/public")
15     public ChatMessage register(@Payload ChatMessage chatMessage, SimpMessageHeaderAccessor headerAccessor) {
16         headerAccessor.getSessionAttributes().put("Jméno", chatMessage.getSender());
17         return chatMessage;
18     }
19
20     @MessageMapping("/chat.send")
21     @SendTo("/topic/public")
22     public ChatMessage sendMessage(@Payload ChatMessage chatMessage) {
23         return chatMessage;
24     }
25 }
```

Spring



Spring

- Main.js

```
1 'use strict';
2
3 var usernamePage = document.querySelector('#username-page');
4 var chatPage = document.querySelector('#chat-page');
5 var usernameForm = document.querySelector('#usernameForm');
6 var messageForm = document.querySelector('#messageForm');
7 var messageInput = document.querySelector('#message');
8 var messageArea = document.querySelector('#messageArea');
9 var connectingElement = document.querySelector('.connecting');
10
11 var stompClient = null;
12 var username = null;
13
14 var colors = [
15     '#2196F3', '#32c787', '#00BCD4', '#ff5652',
16     '#ffc107', '#ff85af', '#FF9800', '#39bbb0'
17 ];
18
19 function connect(event) {
20     username = document.querySelector('#name').value.trim();
21
22     if(username) {
23         usernamePage.classList.add('hidden');
24         chatPage.classList.remove('hidden');
25
26         var socket = new SockJS('/chat');
27         stompClient = Stomp.over(socket);
28
29         stompClient.connect({}, onConnected, onError);
30     }
31     event.preventDefault();
32 }
33
34
35 function onConnected() {
36     // Subscribe to the Public Topic
37     stompClient.subscribe('/topic/public', onMessageReceived);
38
39     // Tell your username to the server
40     stompClient.send("/app/chat.register",
41         {},
42         JSON.stringify({sender: username, type: 'JOIN'})
43     )
44
45     connectingElement.classList.add('hidden');
46 }
47
48
49 function onError(error) {
50     connectingElement.textContent = 'Nemohu se připojit k WebSocket serveru. Obnov stránku.';
51     connectingElement.style.color = 'red';
52 }
```



Spring

- Main.js

```
55 function send(event) {
56     var messageContent = messageInput.value.trim();
57
58     if(messageContent && stompClient) {
59         var chatMessage = {
60             sender: username,
61             content: messageInput.value,
62             type: 'CHAT'
63         };
64
65         stompClient.send("/app/chat.send", {}, JSON.stringify(chatMessage));
66         messageInput.value = '';
67     }
68     event.preventDefault();
69 }
70
71
72 function onMessageReceived(payload) {
73     var message = JSON.parse(payload.body);
74
75     var messageElement = document.createElement('li');
76
77     if(message.type === 'JOIN') {
78         messageElement.classList.add('event-message');
79         message.content = message.sender + ' joined!';
80     } else if (message.type === 'LEAVE') {
81         messageElement.classList.add('event-message');
82         message.content = message.sender + ' left!';
83     } else {
84         messageElement.classList.add('chat-message');
85
86         var avatarElement = document.createElement('i');
87         var avatarText = document.createTextNode(message.sender[0]);
88         avatarElement.appendChild(avatarText);
89         avatarElement.style['background-color'] = getAvatarColor(message.sender);
90
91         messageElement.appendChild(avatarElement);
92
93         var usernameElement = document.createElement('span');
94         var usernameText = document.createTextNode(message.sender);
95         usernameElement.appendChild(usernameText);
96         messageElement.appendChild(usernameElement);
97     }
98
99     var textElement = document.createElement('p');
100     var messageText = document.createTextNode(message.content);
101     textElement.appendChild(messageText);
102
103     messageElement.appendChild(textElement);
104
105     messageArea.appendChild(messageElement);
106     messageArea.scrollTop = messageArea.scrollHeight;
107 }
108
109
110 function getAvatarColor(messageSender) {
111     var hash = 0;
112     for (var i = 0; i < messageSender.length; i++) {
113         hash = 31 * hash + messageSender.charCodeAt(i);
114     }
115
116     var index = Math.abs(hash % colors.length);
117     return colors[index];
118 }
119
120 usernameForm.addEventListener('submit', connect, true)
121 messageForm.addEventListener('submit', send, true)
```

Spring

- Index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0" charset="utf-8">
5 <title>Studentský chat</title>
6 <link rel="stylesheet" href="/css/main.css" />
7 </head>
8 <body background="bez13.jpg" style="background-position: center; background-repeat: no-repeat; background-size: 80%; ">
9 <noscript><h2>Tvůj prohlížeč nepodporuje javascript</h2></noscript>
10
11 <div id="username-page">
12 <div class="username-page-container">
13 <h1 class="title">Zadej svoji přezdívku </h1>
14 <form id="usernameForm" name="usernameForm">
15 <div class="form-group">
16 <input type="text" id="name" placeholder="Přezdívka" autocomplete="off" class="form-control" />
17 </div>
18 <div class="form-group">
19 <button type="submit" class="accent username-submit">Zahaj konverzaci</button>
20 </div>
21 </form>
22 </div>
23 </div>
24
25 <div id="chat-page" class="hidden">
26 <div class="chat-container">
27 <div class="chat-header">
28 <h2>Studentský chat</h2>
29 </div>
30 <div class="connecting">Připojování...</div>
31 <ul id="messageArea">
32
33 </ul>
34 <form id="messageForm" name="messageForm" nameForm="messageForm">
35 <div class="form-group">
36 <div class="input-group clearfix">
37 <input type="text" id="message" placeholder="Napiš zprávu..." autocomplete="off" class="form-control" />
38 <button type="submit" class="primary">Odeslat</button>
39 </div>
40 </div>
41 </form>
42 </div>
43 </div>
44
45 <script src="https://cdnjs.cloudflare.com/ajax/libs/sockjs-client/1.1.4/sockjs.min.js"></script>
46 <script src="https://cdnjs.cloudflare.com/ajax/libs/stomp.js/2.3.3/stomp.min.js"></script>
47 <script src="/js/main.js"></script>
48 </body>
49 </html>
```

Spring

- Main.css

```
1 * {
2   -webkit-box-sizing: border-box;
3   -moz-box-sizing: border-box;
4   box-sizing: border-box;
5 }
6
7 html,body {
8   height: 100%;
9   overflow: hidden;
10 }
11
12 body {
13   margin: 0;
14   padding: 0;
15   font-weight: 400;
16   font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
17   font-size: 1rem;
18   line-height: 1.58;
19   color: #333;
20   /* background-color: #f4f4f4; */
21   height: 100%;
22
23
24
25   /* Center and scale the image nicely */
26   background-position: center;
27   background-repeat: no-repeat;
28   background-size: cover;
29 }
30
31 body:before {
32   height: 50%;
33   width: 100%;
34   position: absolute;
35   top: 0;
36   left: 0;
37   background: 18.jpg;
38   content: "";
39   z-index: 0;
40 }
41
42 .clearfix:after {
43   display: block;
44   content: "";
45   clear: both;
46 }
47
48 .hidden {
49   display: none;
50 }
```

```
52 .form-control {
53   width: 100%;
54   min-height: 38px;
55   font-size: 15px;
56   border: 1px solid #c8c8c8;
57 }
58
59 .form-group {
60   margin-bottom: 15px;
61 }
62
63 input {
64   padding-left: 10px;
65   outline: none;
66 }
67
68 h1, h2, h3, h4, h5, h6 {
69   margin-top: 20px;
70   margin-bottom: 20px;
71 }
72
73 h1 {
74   font-size: 1.7em;
75 }
76
77 a {
78   color: #128ff2;
79 }
80
81 button {
82   box-shadow: none;
83   border: 1px solid transparent;
84   font-size: 14px;
85   outline: none;
86   line-height: 100%;
87   white-space: nowrap;
88   vertical-align: middle;
89   padding: 0.6rem 1rem;
90   border-radius: 2px;
91   transition: all 0.2s ease-in-out;
92   cursor: pointer;
93   min-height: 38px;
94 }
95
96 button.default {
97   background-color: #e8e8e8;
98   color: #333;
99   box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.12);
100 }
101 }
```


Spring

- Main.css

```
102 button.primary {
103     background-color: #25be38;
104     box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.12);
105     color: #fff;
106 }
107
108 button.accent {
109     background-color: #1778dd;
110     box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.12);
111     color: #fff;
112 }
113
114 #username-page {
115     text-align: center;
116 }
117
118 .username-page-container {
119     background: #fff;
120     box-shadow: 0 1px 11px rgba(0, 0, 0, 0.27);
121     border-radius: 2px;
122     width: 100%;
123     max-width: 500px;
124     display: inline-block;
125     margin-top: 42px;
126     vertical-align: middle;
127     position: relative;
128     padding: 35px 55px 35px;
129     min-height: 250px;
130     position: absolute;
131     top: 50%;
132     left: 0;
133     right: 0;
134     margin: 0 auto;
135     margin-top: -160px;
136 }
137
138 .username-page-container .username-submit {
139     margin-top: 10px;
140 }
141
142
143 #chat-page {
144     position: relative;
145     height: 100%;
146 }
147
148 .chat-container {
149     max-width: 700px;
150     margin-left: auto;
151     margin-right: auto;
152     background-color: #fff;
```

```

153     box-shadow: 0 1px 11px rgba(0, 0, 0, 0.27);
154     margin-top: 30px;
155     height: calc(100% - 60px);
156     max-height: 600px;
157     position: relative;
158 }
159
160 #chat-page ul {
161     list-style-type: none;
162     background-color: #FFF;
163     margin: 0;
164     overflow: auto;
165     overflow-y: scroll;
166     padding: 0 20px 0px 20px;
167     height: calc(100% - 150px);
168 }
169
170 #chat-page #messageForm {
171     padding: 20px;
172 }
173
174 #chat-page ul li {
175     line-height: 1.5rem;
176     padding: 10px 20px;
177     margin: 0;
178     border-bottom: 1px solid #f4f4f4;
179 }
180
181 #chat-page ul li p {
182     margin: 0;
183 }
184
185 #chat-page .event-message {
186     width: 100%;
187     text-align: center;
188     clear: both;
189 }
190
191 #chat-page .event-message p {
192     color: #777;
193     font-size: 14px;
194     word-wrap: break-word;
195 }
196
197 #chat-page .chat-message {
198     padding-left: 68px;
199     position: relative;
200 }
201
```

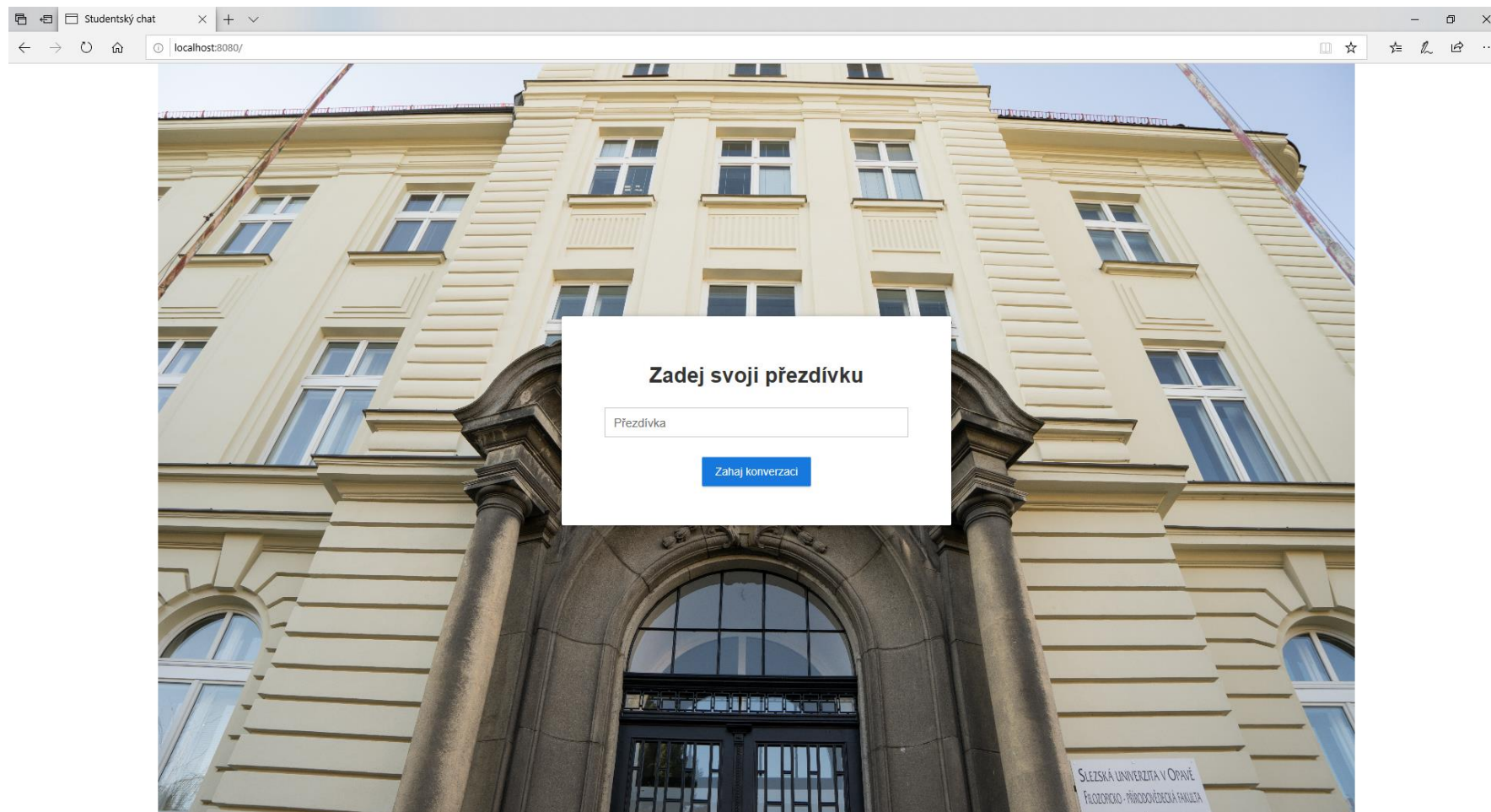
Spring

- Main.css

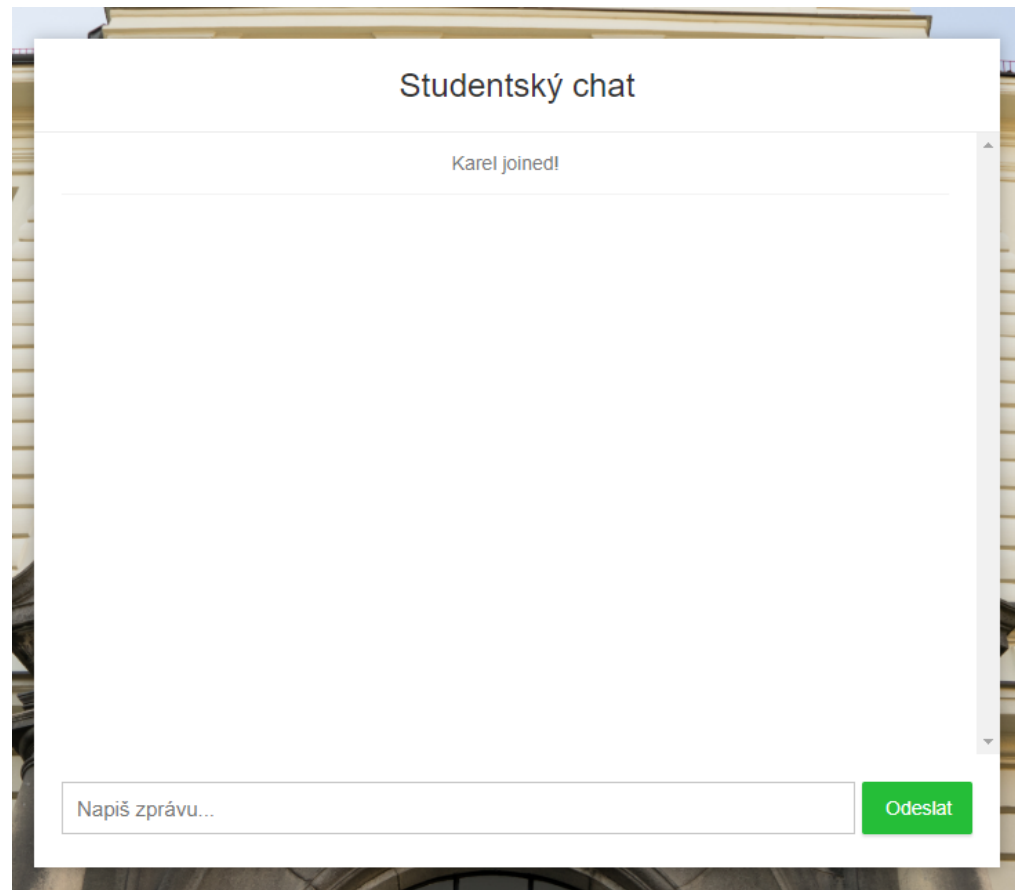
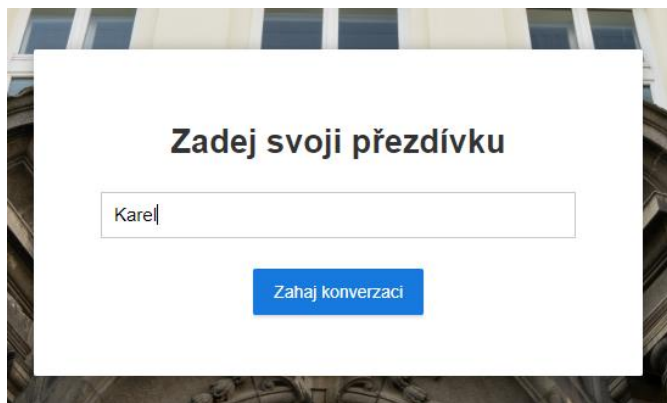
```
202⊖ #chat-page .chat-message i {
203   position: absolute;
204   width: 42px;
205   height: 42px;
206   overflow: hidden;
207   left: 10px;
208   display: inline-block;
209   vertical-align: middle;
210   font-size: 18px;
211   line-height: 42px;
212   color: #fff;
213   text-align: center;
214   border-radius: 50%;
215   font-style: normal;
216   text-transform: uppercase;
217 }
218
219⊖ #chat-page .chat-message span {
220   color: #333;
221   font-weight: 600;
222 }
223
224⊖ #chat-page .chat-message p {
225   color: #43464b;
226 }
227
228⊖ #messageForm .input-group input {
229   float: left;
230   width: calc(100% - 85px);
231 }
232
233⊖ #messageForm .input-group button {
234   float: left;
235   width: 80px;
236   height: 38px;
237   margin-left: 5px;
238 }
239
240⊖ .chat-header {
241   text-align: center;
242   padding: 15px;
243   border-bottom: 1px solid #ecec;
244 }
245
246⊖ .chat-header h2 {
247   margin: 0;
248   font-weight: 500;
249 }
250
251⊖ .connecting {
252   padding-top: 5px;
253   text-align: center;
254   color: #777;
```

```
255   position: absolute;
256   top: 65px;
257   width: 100%;
258 }
259
260
261⊖ @media screen and (max-width: 730px) {
262
263⊖   .chat-container {
264     margin-left: 10px;
265     margin-right: 10px;
266     margin-top: 10px;
267   }
268 }
269
270⊖ @media screen and (max-width: 480px) {
271⊖   .chat-container {
272     height: calc(100% - 30px);
273   }
274
275⊖   .username-page-container {
276     width: auto;
277     margin-left: 15px;
278     margin-right: 15px;
279     padding: 25px;
280   }
281
282⊖   #chat-page ul {
283     height: calc(100% - 120px);
284   }
285
286⊖   #messageForm .input-group button {
287     width: 65px;
288   }
289
290⊖   #messageForm .input-group input {
291     width: calc(100% - 70px);
292   }
293
294⊖   .chat-header {
295     padding: 10px;
296   }
297
298⊖   .connecting {
299     top: 60px;
300   }
301
302⊖   .chat-header h2 {
303     font-size: 1.1em;
304   }
305 }
```

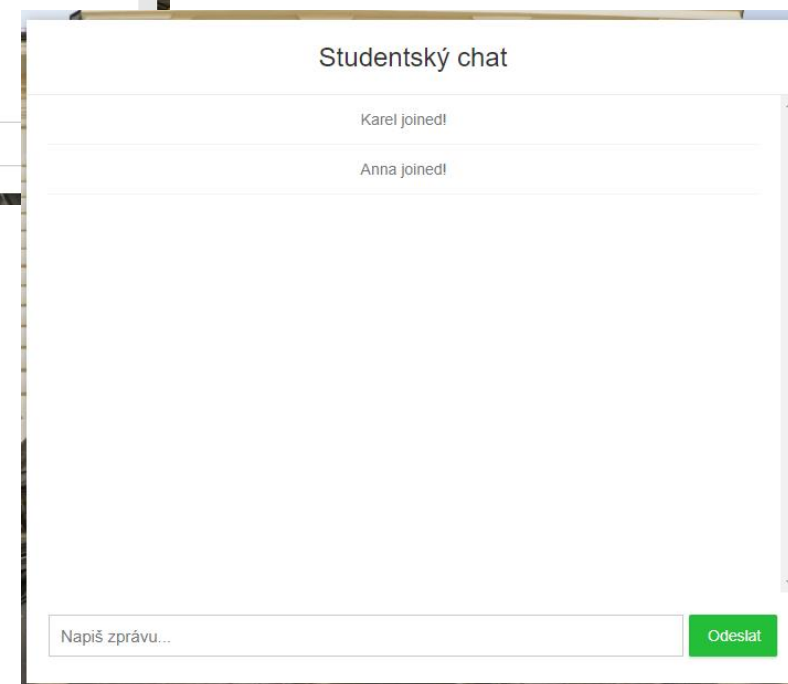
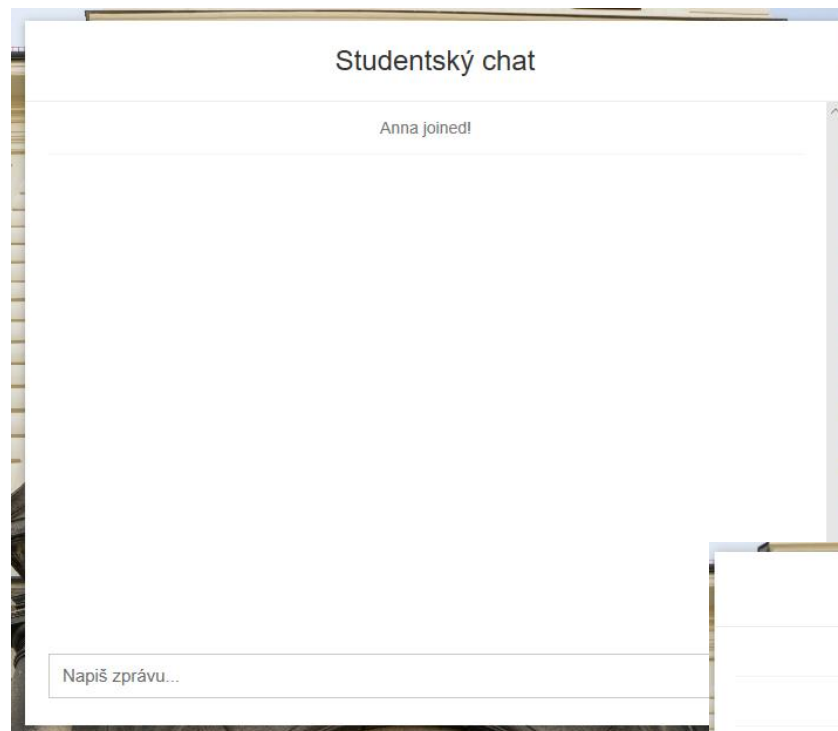
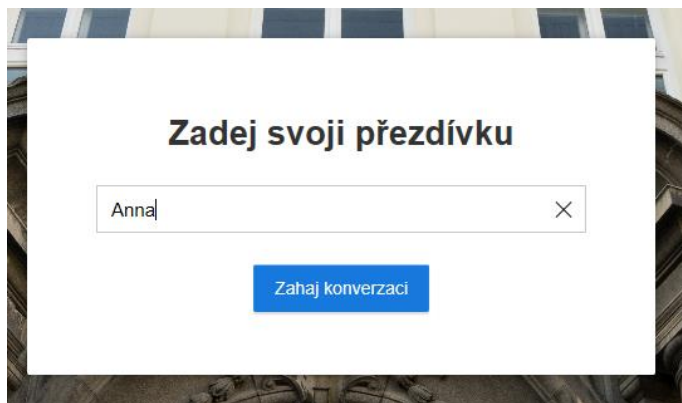
Spring



Spring



Spring



Spring

