콘텐츠 테이블

시스템 설계 및 개발

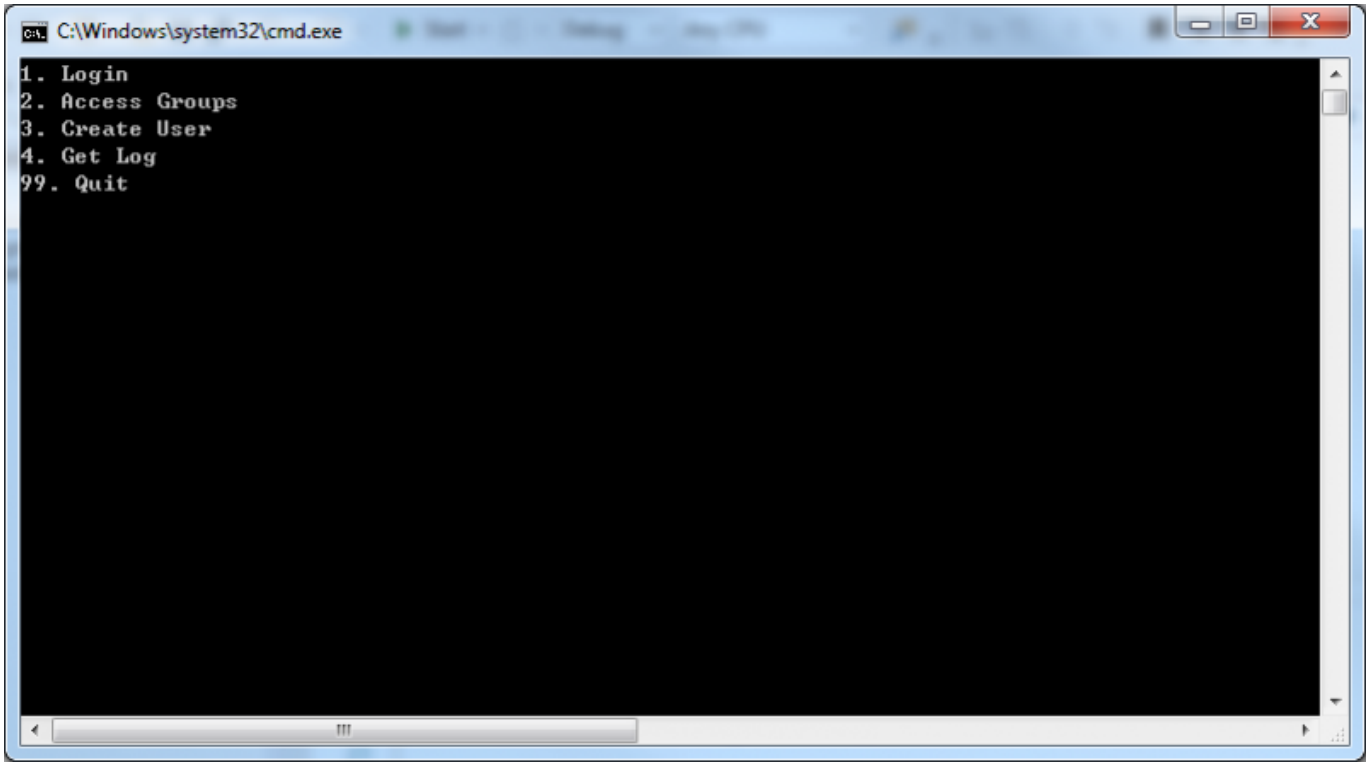
BioStar 2 API 퀵 가이드

소개

공식 BioStar API 개발자 웹 사이트는 https://api.biostar2.com입니다. 공식 웹 사이트를 방문하여 BioStar API 사용 방법에 대한 자세한 내용을 확인할 수 있습니다. 이 문서에서는 C# 또는 독립 실행형 Windows 애플리케이션에 익숙한 사용자를 위해 만든 샘플 애플리케이션에 대해 자세히 살펴보겠습니다. BioStar API는 RESTful API이므로 RESTful API에 익숙하지 않은 경우 BioStar API로 자체 애플리케이션을 구현하는 데 어려움을 겪을 수 있습니다. 이에 따라 본 문서에서 샘플 애플리케이션을 통해 개발을 시작할 수 있는 방법을 안내하고자 합니다.

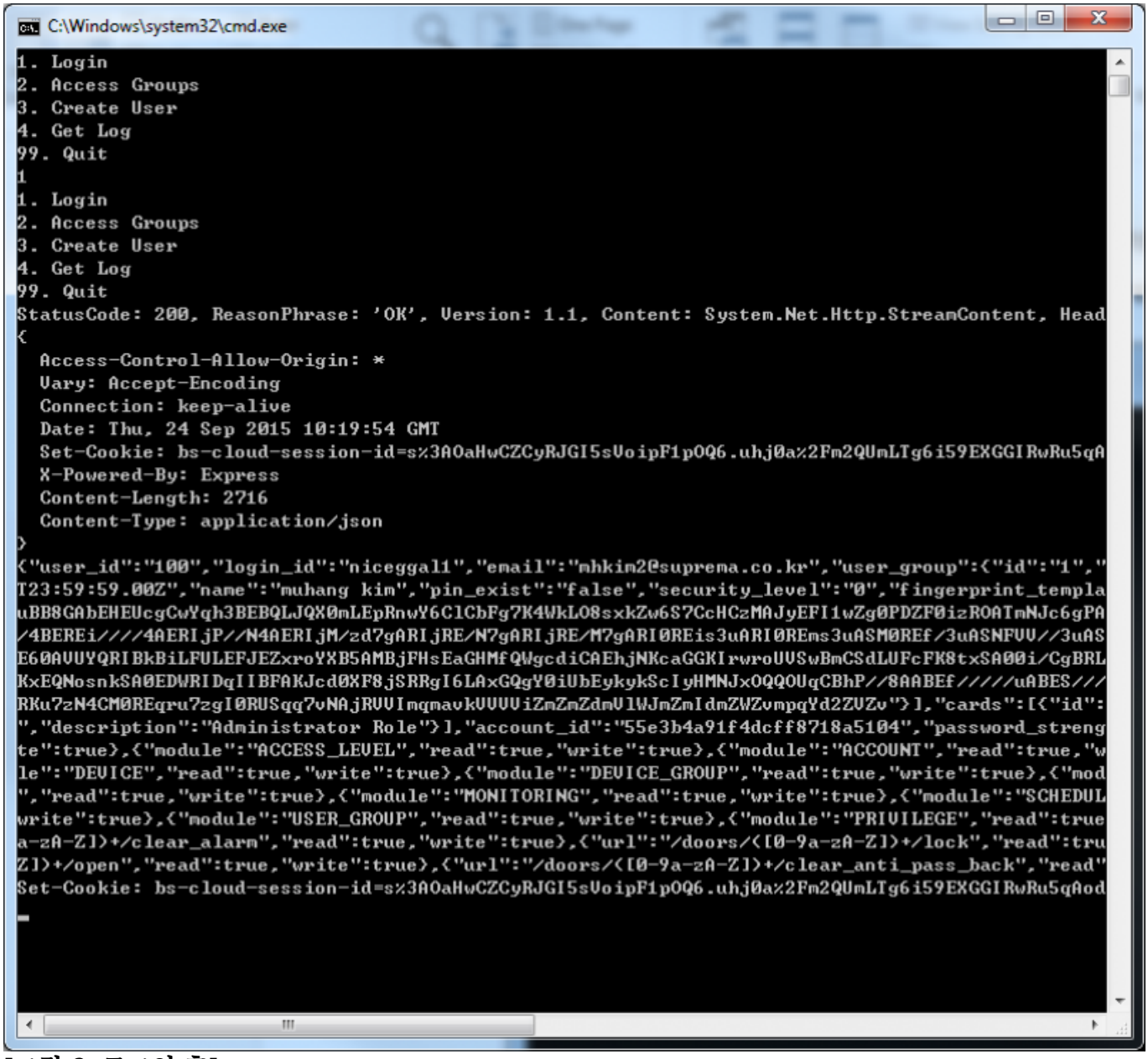
기능

이 샘플 애플리케이션은 Visual C# 콘솔 애플리케이션이며 로그인, 사용자 생성, 출입 그룹 가져오기, 로그 가져오기, 이렇게 네 가지 기본 기능을 포함합니다.



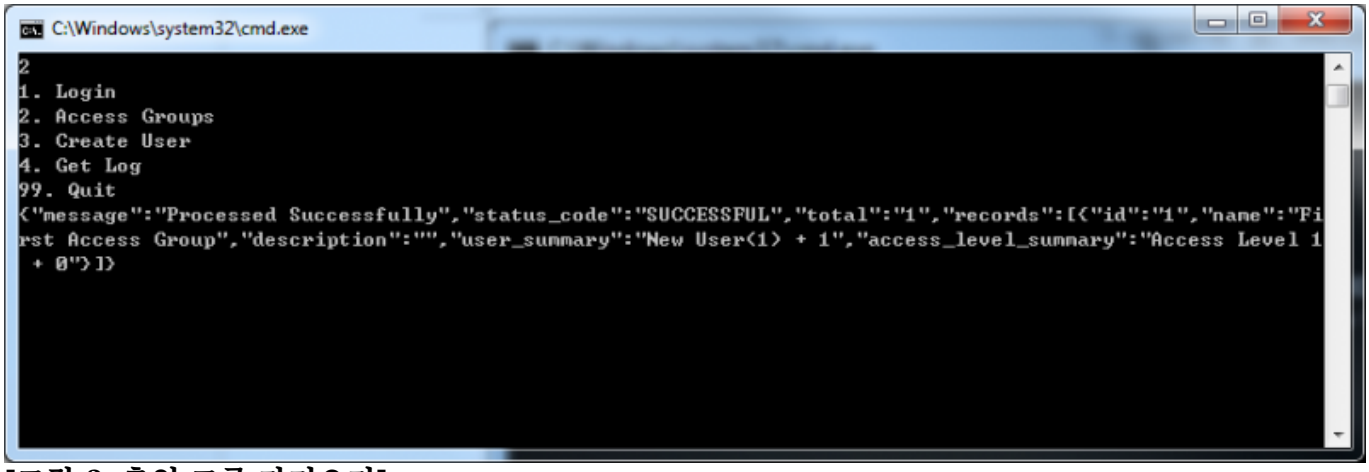
[그림 1. 샘플 애플리케이션을 시작한 화면]

우선 다른 기능을 사용하기 전에 로그인부터 해야 합니다. '1'을 입력하고 <Enter>를 누르면 샘플 애플리케이션이 BioStar 클라우드를 통해 로컬 BioStar 서버에 로그인됩니다. BioStar API는 BioStar 클라우드에서만 작동하며, BioStar 클라우드를 사용하지 않고 로컬 BioStar 서버에서 바로 BioStar API를 호출할 수 없기 때문에 일단 BioStar API를 사용하도록 클라우드 설정을 구성해야 합니다. 가까운 시일 내에 로컬 BioStar API 계층을 출시할 예정입니다.



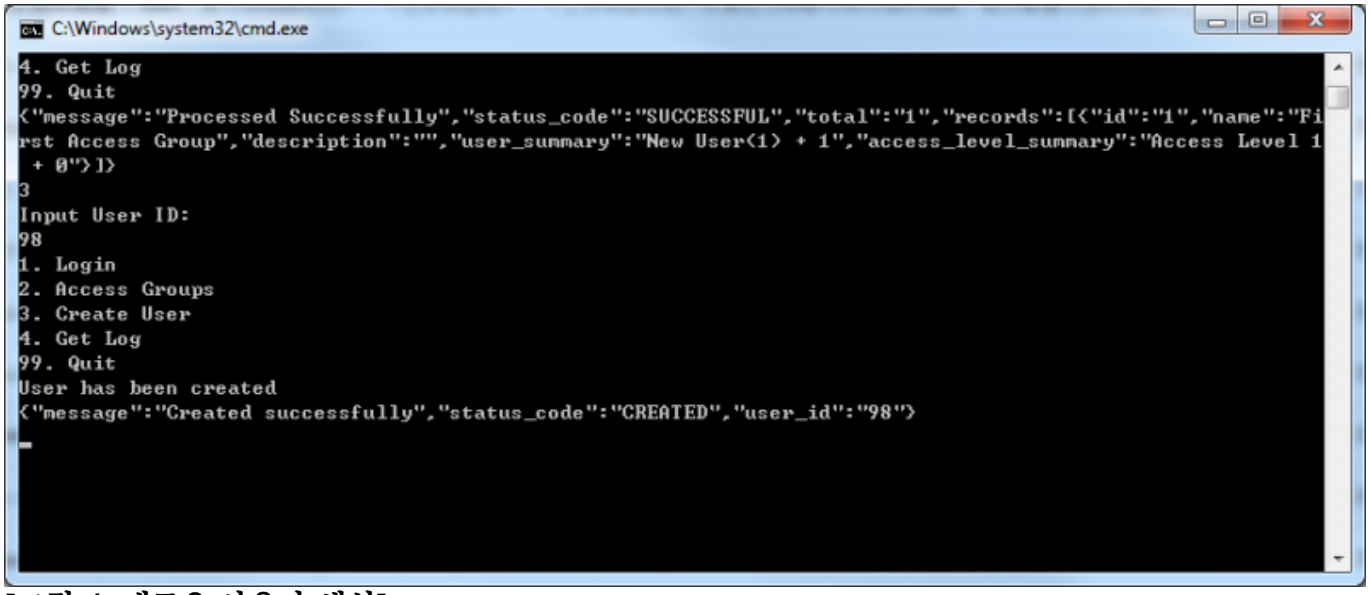
[그림 2. 로그인 후]

성공적으로 로그인하면 로그인하는 데 사용한 사용자 정보를 가져옵니다. 사용자 데이터는 JSON 형식이며, 그림 2에서 볼 수 있듯이 지문 템플릿 등, 매우 상세한 데이터가 포함되어 있습니다. 로그인되었기 때문에 다른 기능을 사용할 수 있습니다. 출입 그룹을 가져오려면 “2”를 입력하고 <Enter>를 누릅니다.



[그림 3. 출입 그룹 가져오기]

스크린샷에서 볼 수 있듯이 BioStar 2 서버에는 출입 그룹이 한 개이며 출입 그룹의 이름은 “First Access Group”입니다. 그리고 “user_summary” 속성을 보면 두 명의 사용자가 있는 것을 알 수 있습니다. 이제 “3”을 입력하고 <Enter>를 눌러 새로운 사용자를 생성하십시오. 사용자 ID를 입력하라는 메시지가 나타날 것입니다. 저는 사용자 ID로 “98”을 입력했고 서버에서 완료 메시지가 표시되었습니다.



[그림 4. 새로운 사용자 생성]

마지막으로 서버에서 로그 이벤트를 가져오겠습니다. “4”를 입력하고 Enter를 누르면 아래 스크린샷에 표시된 대로 서버에서 로그 데이터를 가져옵니다.

[그림 5. 로그 목록 가져오기]

소스 코드 분석

1. 로그인

샘플 애플리케이션에서 가장 중요한 부분은 로그인입니다. 아래에서 함수를 살펴보겠습니다.

```
24          static async void LoginTask()
25          {
26              string resourceAddress = "https://api.biostar2.com/v1/login";
27
28              HttpClient httpClient = new HttpClient();
29
30              JavaScriptSerializer serializer = new JavaScriptSerializer();
31
32              Dictionary<string, string> dicLoginUser = new Dictionary<string, string>();
33              dicLoginUser.Add("name", "ts22");
34              dicLoginUser.Add("password", "rlaangkd!1");
35              dicLoginUser.Add("user_id", "niceggal1");
36
37              string jsonLoginUser = serializer.Serialize(dicLoginUser);
38
39              StringContent sc = new StringContent(jsonLoginUser, Encoding.UTF8, "application/json");
40              HttpResponseMessage httpResponse = await httpClient.PostAsync(resourceAddress, sc);
41
42
43              if(httpResponse.IsSuccessStatusCode == true)
44              {
45                  Console.WriteLine(httpResponse.ToString());
46                  string httpResponseBody = await httpResponse.Content.ReadAsStringAsync();
47                  Console.WriteLine(httpResponseBody);
48
49
50                  MemoryStream responseMemoryStream = new MemoryStream();
51                  StreamWriter sw = new StreamWriter(responseMemoryStream);
52                  sw.Write(httpResponse.ToString());
53                  sw.Flush();
54
55                  bool isSessionIDContained = httpResponse.Headers.Contains("Set-Cookie");
56                  if (isSessionIDContained == true)
57                  {
58                      IEnumerable<string> sessionEnum = httpResponse.Headers.GetValues("Set-Cookie");
59                      foreach(string element in sessionEnum)
60                      {
61                          Console.WriteLine("Set-Cookie: " + element);
62                          string[] strCookieArr = element.Split(new string[] { "bs-cloud-session-id=" }, StringSplitOptions.None);
63                          string[] strCookieArr2 = strCookieArr[1].Split(new string[] { ";" }, StringSplitOptions.None);
64                          sessionID = strCookieArr2[0];
65                      }
66                  }
67                  else
68                  {
69                      Console.WriteLine("Session ID not found");
70                  }
71              }
72              else
73              {
74                  Console.WriteLine("Failed to log in");
75                  Console.WriteLine(httpResponse.ToString());
76              }
77          }
```

- 26행: 로컬 BioStar 서버에 로그인하는 데 사용할 URL입니다. HTTPS 프로토콜만 허용하며 “api.biostar2.com/v1/“은 접두사입니다. 접두사 다음의 “login”은 취할 동작 또는 작업을 나타냅니다.
- 27행: 이 샘플 코드에서는 요청을 전송하고 BioStar 클라우드에서 응답을 받는 데 HttpClient 클래스를 사용합니다.
- 30행: 데이터를 JSON 형식의 데이터로 변환하거나 JSON 형식의 데이터를 원하는 형식으로 구문 분석하는 데 JavaScriptSerializer 클래스가 필요합니다.
- 32~35행: 이 행은 문자열 키 및 문자열 값으로 구성된 사전을 생성합니다. 하위 도메인 이름, ID, 암호, 이렇게 세 가지 매개 변수가 반드시 필요합니다. “name” 필드가 하위 도메인 이름에 해당하므로 이 필드를 하위 도메인 이름으로 설정하십시오.
- 37행: 이 행은 사전을 JSON 형식의 문자열로 변환합니다.
- 39행: 이 행은 JSON 형식의 문자열을 HTTP 요청 콘텐츠로, UTF8을 인코딩 옵션으로, JSON을 미디어 유형으로 설정합니다.
- 40행: 로그인을 위한 HTTP 요청을 만드는 데 HTTP POST 메서드를 사용합니다.
- 45~53행: 디버깅 목적으로 HTTP 응답 콘텐츠를 출력합니다.
- 55~65행: 로그인 정보가 올바르면 서버에서 세션 정보가 수신됩니다. API 호출을 할 때마다 HTTP 헤더에 세션 정보를 넣어야 합니다. 즉, 55~65행은 HTTP 응답 헤더에서 나중에 사용할 수 있도록 세션 정보를 추출합니다.

2. 출입 그룹 가져오기

```
203         static async void AccessGroupsTask()
204         {
205             if (sessionID == null)
206             {
207                 Console.WriteLine("You must log in first!");
208                 return;
209             }
210
211             CookieContainer cookieContainer = new CookieContainer();
212
213             HttpClientHandler handler = new HttpClientHandler();
214             handler.CookieContainer = cookieContainer;
215
216             HttpClient client = new HttpClient(handler);
217
218
219             cookieContainer.Add(new Uri("https://api.biostar2.com"), new Cookie("bs-cloud-session-id", sessionID));
220             HttpResponseMessage httpResponse = await client.GetAsync("https://api.biostar2.com/v1/access_groups");
221
222             if (httpResponse.IsSuccessStatusCode == true)
223             {
224                 string httpResponseBody = await httpResponse.Content.ReadAsStringAsync();
225                 Console.WriteLine(httpResponseBody);
226             }
227             else
228             {
229                 Console.WriteLine("Retrieving Access Groups Failed");
230                 Console.WriteLine(httpResponse.ToString());
231             }
232         }
```

- 205~209행: 우선 성공적으로 로그인되었고 세션 ID가 저장되었는지 확인해야 합니다.
- 211행: BioStar 서버로 세션 ID 정보를 전송하는 데 CookieContainer 클래스를 사용합니다.
- 219행: 쿠키에 세션 ID를 입력할 때 URI를 지정해야 합니다.
- 220행: HTTP GET 메서드를 통해 출입 그룹 가져오기가 완료됩니다.

3. 이벤트 가져오기

```
125         static async void GetLogTask()
126         {
127             if(sessionID == null)
128             {
129                 Console.WriteLine("You must log in first!");
130                 return;
131             }
132 
133             CookieContainer cookieContainer = new CookieContainer();
134 
135             HttpClientHandler handler = new HttpClientHandler();
136             handler.CookieContainer = cookieContainer;
137 
138             HttpClient httpClient = new HttpClient(handler);
139 
140             HttpClient client = new HttpClient(handler);
141             cookieContainer.Add(new Uri("https://api.biostar2.com"), new Cookie("bs-cloud-session-id", sessionID));
142 
143             string resourceAddress = "https://api.biostar2.com/v1/monitoring/event_log/search";
144 
145             string startTime = "1970-01-01T00:00:00Z";
146             string endTime = DateTime.UtcNow.ToString("yyyy-MM-ddTHH:mm:ssZ");
147 
148             DateTime dtLatestLogTime = new DateTime(1970, 1, 1);
149 
150             JavaScriptSerializer serializer = new JavaScriptSerializer();
151 
152             for (int logCallIndex = 0; logCallIndex < 1000; logCallIndex++)
153             {
154                 endTime = DateTime.UtcNow.ToString("yyyy-MM-ddTHH:mm:ssZ");
155 
156                 string payload = "{ \"datetime\": [\"" + startTime + "\", \"" + endTime + "\"] }";
157 
158 
159                 StringContent sc = new StringContent(payload, Encoding.UTF8, "application/json");
160                 HttpResponseMessage httpResponse = await httpClient.PostAsync(resourceAddress, sc);
162                 if (httpResponse.IsSuccessStatusCode == true)
163                 {
164                     Console.WriteLine("Succeeded to retrieve log from " + startTime + " to " + endTime);
165                     string httpResponseBody = await httpResponse.Content.ReadAsStringAsync();
166                     Console.WriteLine(httpResponseBody);
167 
168                     endTime = startTime;
169 
170                     Dictionary<string, dynamic> logValues = serializer.Deserialize<Dictionary<string, dynamic>>(httpResponseBody);
171                     foreach(KeyValuePair<string, dynamic> logElement in logValues)
172                     {
173                         if (logElement.Key == "records")
174                         {
175                             foreach (Dictionary<string, dynamic> recordElement in logElement.Value)
176                             {
177                                 if(recordElement.ContainsKey("datetime"))
178                                 {
179                                     Console.WriteLine(recordElement["datetime"]);
180                                     DateTime dtLogTime = DateTime.Parse(recordElement["datetime"]);
181 
182                                     if(dtLogTime > dtLatestLogTime)
183                                     {
184                                         dtLatestLogTime = dtLogTime;
185                                         startTime = dtLatestLogTime.ToUniversalTime().AddSeconds(1).ToString("yyyy-MM-ddTHH:mm:ssZ");
186                                     }
187                                 }
188                             }
189                         }
190                     }
191 
192                     System.Threading.Thread.Sleep(1000);
193                 }
194                 else
195                 {
196                     Console.WriteLine("Log Retrieval Failed from " + startTime + " to " + endTime);
197                     Console.WriteLine(httpResponse.ToString());
198                     break;
199                 }
```

- 152행: 서버에서 간격을 두고 반복적으로 이벤트를 가져오기 위해 For 루프를 사용합니다.
- 154~156행: 서버에서 이벤트를 가져올 때, 시작 시간과 종료 시간을 지정해야 합니다. 이번에는 Dictionary 클래스를 사용하지 않고 데모용으로 직접 JSON 형식의 문자열을 구축합니다.
- 170~188행: JSON 형식의 데이터를 사전 데이터 구조로 구문 분석하기 위해 문자열 키 및 동적 값과 함께 Dictionary 클래스를 사용합니다. 이벤트는 배열 형식이기 때문에 사전에서 동적 유형 값을

사용해야 합니다.

4. 사용자 생성

```
79          static async void CreateUserTask()
80          {
81              if (sessionID == null)
82              {
83                  Console.WriteLine("You must log in first!");
84                  return;
85              }
86
87              CookieContainer cookieContainer = new CookieContainer();
88
89              HttpClientHandler handler = new HttpClientHandler();
90              handler.CookieContainer = cookieContainer;
91
92              HttpClient httpClient = new HttpClient(handler);
93
94              HttpClient client = new HttpClient(handler);
95              cookieContainer.Add(new Uri("https://api.biostar2.com"), new Cookie("bs-cloud-session-id", sessionID));
96
97              string resourceAddress = "https://api.biostar2.com/v1/users";
98
99              Console.WriteLine("Input User ID: ");
100             string userInputID = Console.ReadLine();
101
102             JavaScriptSerializer serializer = new JavaScriptSerializer();
103
104             Dictionary<string, string> dicNewUser = new Dictionary<string, string>();
105             dicNewUser.Add("user_id", userInputID);
106
107             string payload = serializer.Serialize(dicNewUser);
108
109             StringContent sc = new StringContent(payload, Encoding.UTF8, "application/json");
110             HttpResponseMessage httpResponse = await httpClient.PostAsync(resourceAddress, sc);
```

- 99~100행: 새로운 사용자 ID에 대한 사용자 입력을 수신합니다.
- 104~105행: 새로운 사용자를 생성할 때 제공해야 하는 필수 속성은 사용자 ID 밖에 없습니다.

결론

지금까지 BioStar API를 활용하는 방법에 대해 간략하게 알아보았습니다. 이 문서의 대부분에서는 BioStar API에 대한 구체적인 내용보다 HttpClient 클래스의 기본적인 사용 방법과 웹 API를 사용하기 위해 해야 할 작업에 대해 살펴봤습니다. Web API에 익숙하지 않아도 이 문서를 통해 BioStar API의 사용 방법을 알아갈 수 있을 것입니다. BioStar API에 대한 자세한 내용은 BioStar API 웹 사이트를 참조하십시오.
https://api.biostar2.com