HttpClients.custom().setDefaultCredentialsProvider().setDefaultRequestConfig().build();

1. **public** **class** Client {

2.

3. **public** **void** test(){

4.

5. RequestConfig requestConfig = RequestConfig.custom()

.setConnectionRequestTimeout(10000)

.setConnectTimeout(10000)

.setSocketTimeout(10000).build();

6. CredentialsProvider credentialsProvider = **new** BasicCredentialsProvider();

7. credentialsProvider.setCredentials(AuthScope.ANY,**new** UsernamePasswordCredentials("xxx","xxx"));

8. CloseableHttpClient httpClient = HttpClients.custom().setDefaultCredentialsProvider(credentialsProvider).setDefaultRequestConfig(requestConfig).build();

9. **try** {

10. URI uri = **new** URIBuilder("http://127.0.0.1:9200").setParameter("name","yyh").build();

11. HttpPost httpRequest = **new** HttpPost(uri);

12. httpRequest.setHeader("Basic Authorization","xxx");

13. httpRequest.setEntity(**new** StringEntity("", ContentType.APPLICATION\_JSON));

14. httpClient.execute(httpRequest);

15. } **catch** (Exception e) {

16. e.printStackTrace();

17. }

18.

19. }

20. }

Functioin

public class FuncSearch {

public void searchRelative() throws Exception {

searchFunction(request -> search(request), getRequest(SearchRequest.class));

}

public <Request, Response> Response searchFunction(Function<Request, Optional<Response>> fn, Request request) {

return fn.apply(request).get();

}

public Optional<SearchResponse> search(SearchRequest searchRequest) {

return Optional.of(new SearchResponse());

}

public <T> T getRequest(Class<T> searchRequest) throws Exception {

return (T) searchRequest.getClass().newInstance();

}

}

IO

Scanner in = new Scanner(System.in);

System.out.printLn("what is your name");

String name = in.nextLine();

System.out.printLn("how old are you");

int age = in.extInt();

System.out.printLn("name: " + name + " age: " + age);

OutputStreamWriter与BufferedWriter区别

BufferedWriter需手动换行，bw.newLine();

OutputStreamWriter底层使用的是StreamEncoder，也就是对charset进行了编码，BufferedWriter不能指定编码

OutputStreamWriter和BufferedWriter的缺省buffer size,OutputStreamWriter是8k bytes，BufferedWriter是8k chars。

public void test(){  
 try {  
 File file = new File("D:/test.text");  
 OutputStream outputStream = new FileOutputStream(file);  
 OutputStreamWriter outputStreamWriter = new OutputStreamWriter(outputStream);  
 BufferedWriter bufferedWriter = new BufferedWriter(outputStreamWriter);  
 bufferedWriter.write("hello world");  
  
 InputStream inputStream = new FileInputStream(file);  
 InputStreamReader inputStreamReader = new InputStreamReader(inputStream);  
 BufferedReader bufferedReader = new BufferedReader(inputStreamReader);  
   
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
}

fff