How to call an API from the Java Code

Steps:

1. Create an instance of Closeablehttpclient.
2. Create an instance of HttpPost, giving the URL as a parameter to the constructor.
3. Build the request body as a String.
4. Create an instance of StringEntity by passing the String as parameter to the constructor.
5. Set the entity to the HttpPost instance.
6. Set the headers to the HttpPost instance.
7. Set the authorization headers also.
8. Execute the request and hold the response in HttpResponse object
9. Now the response can be sent back to the calling method or response can be converted into String and sent back to the calling method.

Example:

This method is in the file WishListService.java.

```
/*
     * Given the user ID, list of item IDs are retrieved from the wishlist
table.
     * Product details for the list of item IDS
     * are retrieved from an API and the response is converted to String
     * and the String is returned.
     */
    public String getWishLists(int userId) throws IOException {

        List<WishListModel> wishes = new ArrayList<>();
        StringBuilder items = new StringBuilder();

        //Fetching all the wishlisted items for the given user and adding it
to a list
        wishes = wishListRepository.findByWishListIdUserId(userId);
```

```
        //If there is no wishlisted item for that user , it returns null
        if (wishes.size() == 0)
           return null;

        // If there are some items ,
        // those items are added to a String builder wrapping it with double
quotes and separating by comma
        for( WishListModel wish : wishes){
           String itemNumber = wish.getWishListId().getItemNumber();
           items.append("\"");
           items.append(itemNumber);
           items.append("\",");
        }
        items.deleteCharAt(items.length()-1);

        // Forming a String to hold the request body of HttpPost method
        // appending it with the String of items built.
        String requestBody = "{\"catalog\": {"
        +"\"@xmlns\":
\"http://xmlns.oracle.com/apps/per/rest/catalog/GET_CATALOG_ITEMS\","
        +"\"RESTHeader\": {"
        +"\"xmlns\":
\"http://xmlns.oracle.com/apps/per/rest/catalog/header\","
        +"\"Responsibility\":\"US_HRMS_MANAGER\","
        +"\"RespApplication\":\"PER\","
        +"\"SecurityGroup\":\"STANDARD\","
        +"\"NLSLanguage\":\"AMERICAN\","
        +"\"Org_Id\" :\"204\""
        +"},"
        +"\"InputParameters\": {"
        +"\"P_ITEMS\": {"
```

```
        +"\"ITEM_LIST\":["+items
        +"]} } } }";

        // Basic Authorization credentials required to call the API, and it is encoded for security.
        String usernameColonPassword = "sysadmin:sysadmin";
        String basicAuthPayload = "Basic " + Base64.getEncoder().encodeToString(usernameColonPassword.getBytes());

        // try with resources statement. A resource is an object that must be closed after the program is finished with it
        try (CloseableHttpClient client = HttpClientBuilder.create().build()) {

            // Build HttpPostRequest with URL, RequestBody, headers and Basic Authorization
            HttpPost request = new HttpPost("http://e3a9ff5.online-server.cloud:8007/webservices/rest/price/GET_CATALOG_ITEMS/");
            request.setEntity(new StringEntity(requestBody));
            request.setHeader("Content-Type","application/json");
            request.setHeader("Authorization",basicAuthPayload);

            //Execution of the request
            HttpResponse response = client.execute(request);

            //Below lines of code receives the response and change the response object to String
            BufferedReader bufReader = new BufferedReader(
                new InputStreamReader(response.getEntity().getContent()));
            StringBuilder builder = new StringBuilder();
            String line;
```

```
            while ((line = bufReader.readLine()) != null) {
                builder.append(line);
                builder.append(System.lineSeparator());
            }
            return builder.toString();
        }
        catch(Exception e){
             return null;
        }
    }
```