**Java Rest template Sample in FDC**

@RequestMapping("/current-user-auth-and-firm-associations")

**public** ResponseEntity<?> getCurrentUserAuthAndFirmAssociations(String userId, @RequestHeader Map<String, String> requestHeaders) {

Long start = System.currentTimeMillis();

LOGGER.info("getCurrentUserAuthAndFirmAssociations Time Start {}", Utility.whiteList(LocalTime.now()));

LOGGER.info("User Id {}", Utility.whiteList((userId)));

MultiValueMap<String, String> attributeMap = **new** LinkedMultiValueMap<>();

// here we use MultiValueMap to store the apigee headers.

**try** {

attributeMap.add("userId", URLDecoder.decode(userId, "UTF-8"));

// here the data is decoded with UTF-8 encoding

} **catch** (UnsupportedEncodingException e) {

LOGGER.error("UnsupportedEncodingException in currentUserAuthAndFirmAssociations api");

}

HttpHeaders headers = **new** HttpHeaders();

headers.setAll(getApigeeHeaders(requestHeaders));

headers.setContentType(MediaType.APPLICATION\_JSON);

HttpEntity<Object> entity = **new** HttpEntity<>(headers);

StringBuffer url = **new** StringBuffer();

url.append(env.getProperty(protocol)).append(env.getProperty(DCP\_SCS\_USER\_MGMT\_ENDPOINT)).append(env.getProperty(suffix));

// we are building a url here based on the property file

UriComponentsBuilder builder = UriComponentsBuilder

.fromUriString(url.toString())

.path(fdcScsUserMgmtPathUserStatusAndFirmAssociations)

.queryParams(attributeMap);

LOGGER.info("User Mgmt URL : {}", builder.toUriString());

ResponseEntity<?> statusEntity = restTemplate.exchange(builder.toUriString(), HttpMethod.GET, entity, Object.**class**);

// builder.toUriString() – url created from the property files

// entity – header values stored as entity header.

// Object.class – return / response type of the data from the operation

List<UserAssociatedFirms> firmUserList = (List<UserAssociatedFirms>) statusEntity.getBody();

// this line let us to get the response body from the payload

LOGGER.info("userStatusAndFirmAssociations List Size : {}", firmUserList.size());

List<String> firmSids = firmUserList.stream().map(UserAssociatedFirms::getFrmAccountId).collect(Collectors.toList());

String firmSidsStr = firmSids.toString().replace("[", "").replace("]", "");

// convert list to a string

attributeMap = **new** LinkedMultiValueMap<>();

**try** {

attributeMap.add("firmSids", URLDecoder.decode(firmSidsStr, "UTF-8"));

} **catch** (UnsupportedEncodingException e) {

LOGGER.error("UnsupportedEncodingException in currentUserAuthAndFirmAssociations api");

}

StringBuffer fmurl = **new** StringBuffer();

fmurl.append(env.getProperty(protocol)).append(env.getProperty(DCP\_SCS\_FIRM\_MGMT\_ENDPOINT)).append(env.getProperty(suffix));

builder = UriComponentsBuilder

.fromUriString(fmurl.toString())

.path(fdcScsFirmManagementPathEnableFirms)

.queryParams(attributeMap);

LOGGER.info("Firm Mgmt URL : {}", builder.toUriString());

ResponseEntity<?> enabledFirmsE = restTemplate.exchange(builder.toUriString(), HttpMethod.GET, entity, Object.**class**);

// same as above

List<Firm> enabledFirmsList = (List<Firm>) enabledFirmsE.getBody();

Long end = System.currentTimeMillis();

LOGGER.info("getCurrentUserAuthAndFirmAssociations Time End {}", LocalTime.now());

LOGGER.info("getCurrentUserAuthAndFirmAssociations Time consumed {}{}", TimeUnit.MILLISECONDS.toSeconds(end - start), "s");

**return** **new** ResponseEntity<>("", HttpStatus.OK);

}