2018 年“花旗杯”金融创新应用大赛

API 调用报告

参赛题目：砺金--住房租赁资产证券化 REITs 平台

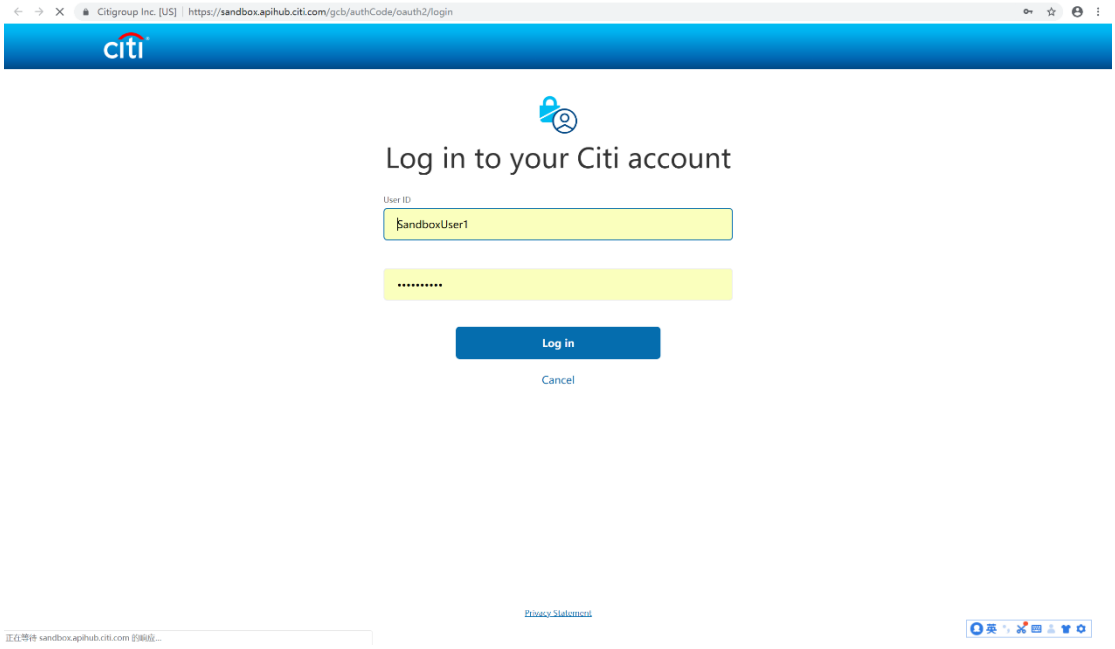
大赛队长：褚天硕

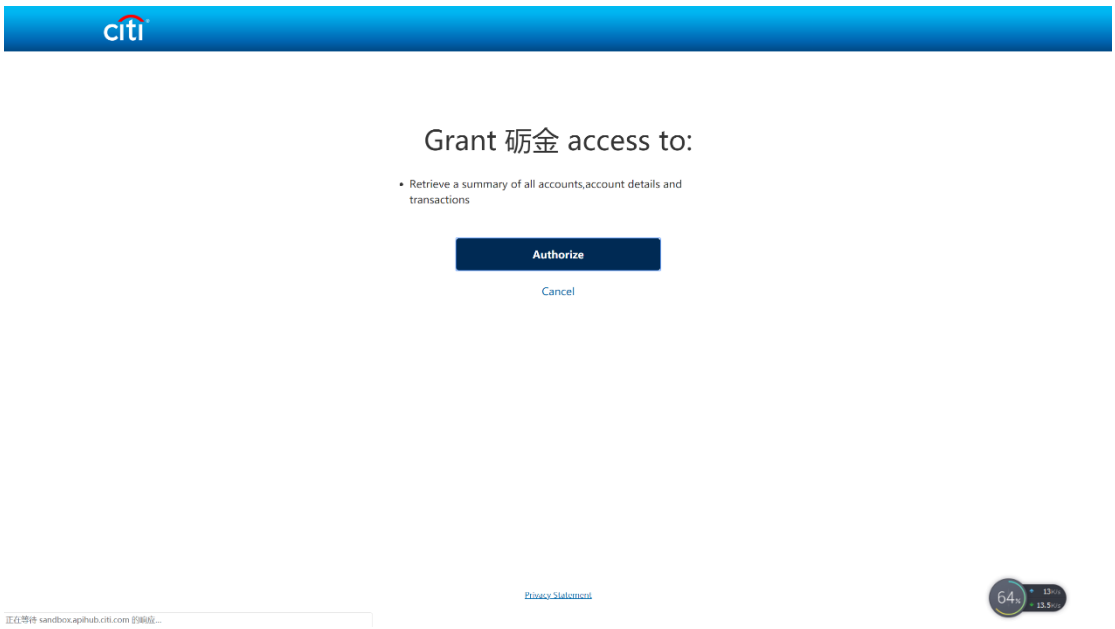
辅导老师：高 明、隋 聪

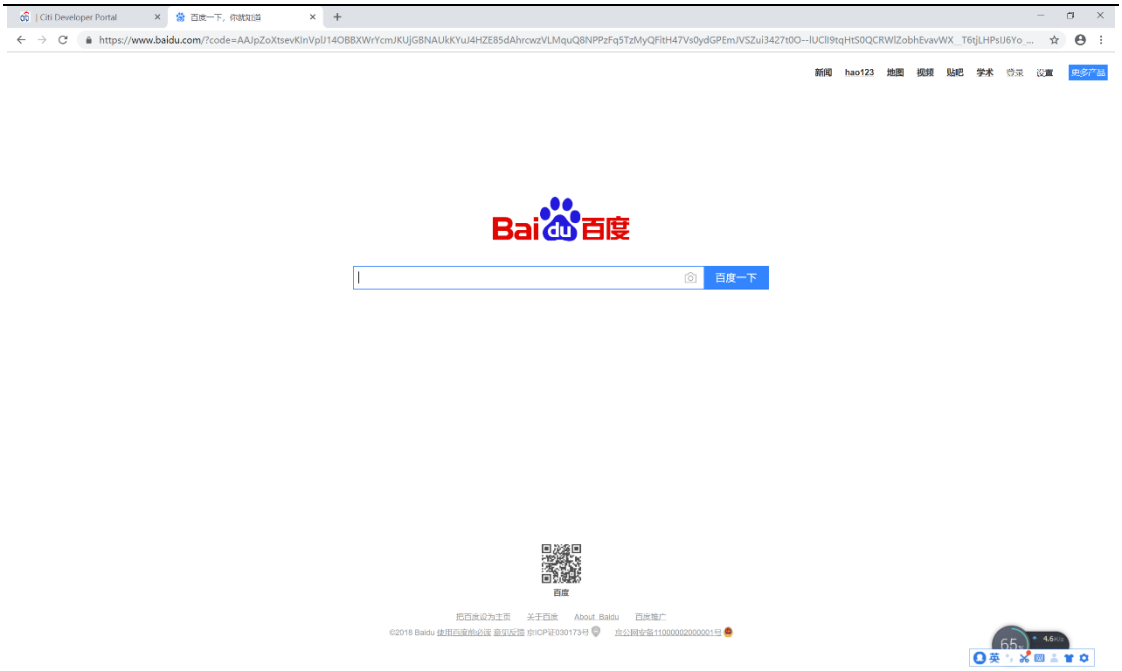
隶属学校：东北财经大学

目 录

1. 第一部分：项目展示

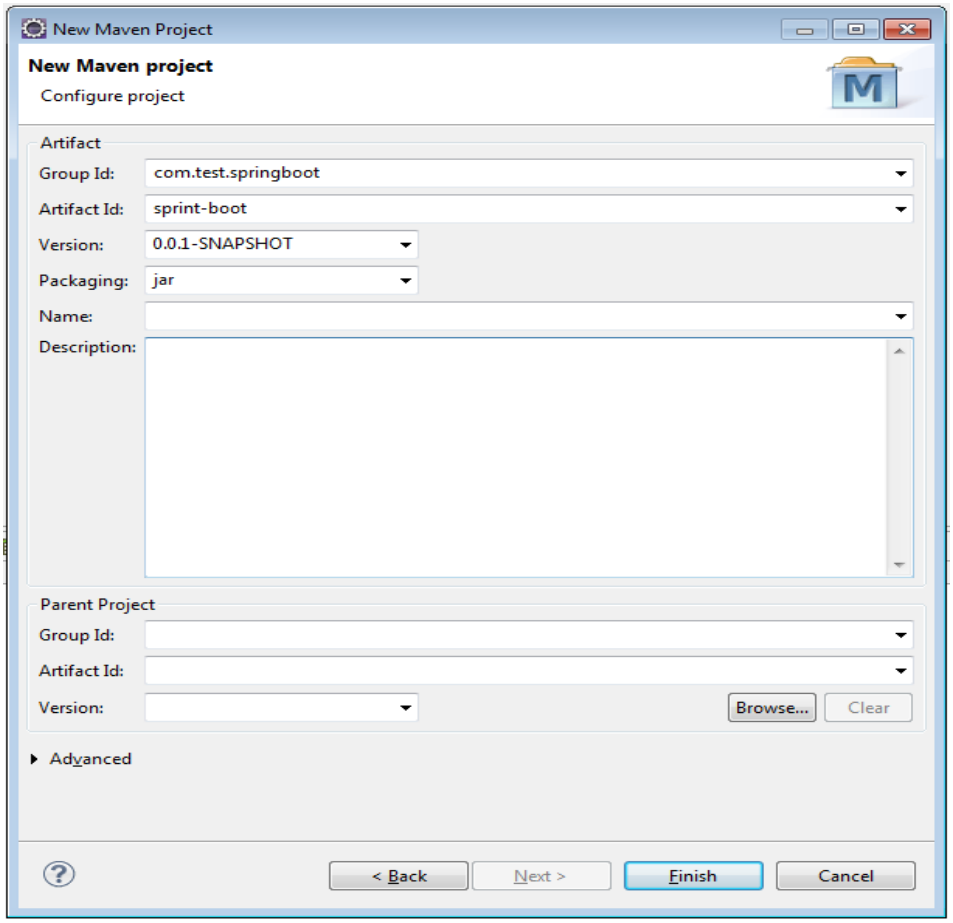

Citigroup Inc. [US] | https://sandbox.apihub.citi.com/gcb/authCode/oauth2/login
citi
Log in to your Citi account
User ID
sandboxUser1
Log in
Cancel
Privacy Statement


citi
Grant 砺金 access to:
Retrieve a summary of all accounts,account details and transactions
Authorize
Cancel
Privacy Statement



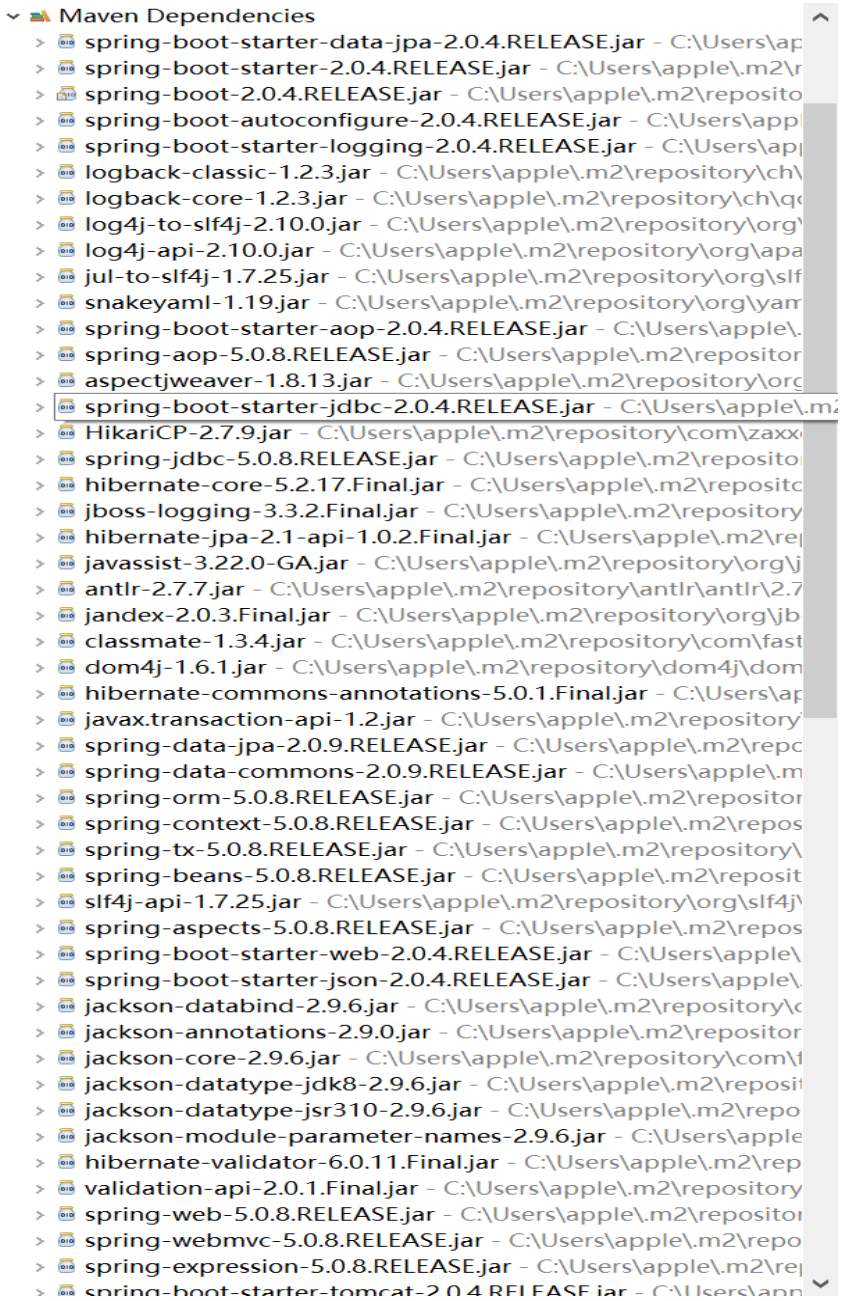
2. 第二部分：具体步骤

（1）创建 Maven Project。



（2）编辑 pom.xml 文件，加入需要的 jar 包。

```
<modelVersion>4.0.0</modelVersion>
<groupId>com.demo.developerapi</groupId>
<artifactId>springboot</artifactId>
<version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>springboot</name>
  <description>Demo project for Spring Boot</description>

  <parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>2.0.4.RELEASE</version>
      <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <properties>
      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
      <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
      <java.version>1.8</java.version>
  </properties>


<dependencies>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-tomcat</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.mybatis.spring.boot</groupId>
        <artifactId>mybatis-spring-boot-starter</artifactId>
        <version>1.3.1</version>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
```



（3）创建 Application.java，用于启动总程序。

```
package com.demo.developerapi;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

（4）创建 GetAccounts.java，写入要实现的 api 功能。

```java
package com.demo.developerapi;

import java.io.IOException;

public class GetAccounts {

    public Map<String, String> getBizToken(APIContext context) throws IOException{
        step1GetAccessToken(context);
        if(context.getAccessToken()==null){
            return null;
        }
        Map<String, String> map = step2GetBizToken(context);
        if(context.getEventId()==null){
            return null;
        }
        return map;
    }

    public String getAccounts(String username, String password, APIContext context) throws IOException {
        context.setUsername(username);
        context.setPassword(password);
        step3GetRealAccessToken(context);
        if(context.getRealAccessToken()==null){
            return null;
        }
        String accounts = step4GetAccounts(context);
        if(context.getAccounts()==null){
            return null;
        }
        return accounts;
    }

    public String getAccountDetail(String accountId, APIContext context) throws IOException {
        context.setAccountId(accountId);
        String accountDetail = step5GetAccountDetails(context);
        if(accountDetail == null) {
            return null;
        }
        return accountDetail;
    }

    public String getTransactions(String accountId, APIContext context) throws IOException {
        context.setAccountId(accountId);
        String transaction = step6GetTransaction(context);
        if(transaction == null) {
            return null;
        }
        return transaction;
    }

    public static String step1GetAccessToken(APIContext context) throws IOException {
        OkHttpClient client = new OkHttpClient();
        String client_id = APIConstant.CLIENT_ID;
        String client_scrent = APIConstant.CLIENT_SCRENT;
        String encode_key = client_id + ":" + client_scrent;
        String authorization = "Basic " + Base64.encodeBase64String(encode_key.getBytes());
        MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
        RequestBody body = RequestBody.create(mediaType, "grant_type=client_credentials&scope=/api");
```

（5）创建 SampleController.java，写入要调用的端口。

```java
package com.demo.developerapi;

import java.io.IOException;

@Controller
public class SampleController extends SpringBootServletInitializer {
    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(SampleController.class);
    }

    APIContext context = new APIContext();


    @RequestMapping("/")
    public String index(Model model) throws IOException {
        GetAccounts accs = new GetAccounts();
        Map<String, String> map = accs.getBizToken(context);
        String modulus = (String) map.get("modulus");
        String exponent = (String) map.get("exponent");
        String eventId = (String) map.get("eventId");
        model.addAttribute("modulus", modulus);
        model.addAttribute("exponent", exponent);
        model.addAttribute("eventId", eventId);
        if(modulus == null || exponent == null || eventId == null) {
            return "errorPage";
        }else {
            return "index";
        }
    }

    @RequestMapping("/login")
    public String login(HttpServletRequest request, Model model) throws IOException{
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        GetAccounts accs = new GetAccounts();
        String accounts = accs.getAccounts(username, password,context);
        if(accounts != null){
            model.addAttribute("accounts", accounts);
            return "accountSummary";
        }else{
            return "errorPage";
        }
    }


    @RequestMapping("/account")
    public String account(HttpServletRequest request, Model model) throws IOException{
        String accountId = request.getParameter("accountId");
        GetAccounts accs = new GetAccounts();
        String accountDetails = accs.getAccountDetail(accountId, context);
        String transactionDetails = accs.getTransactions(accountId, context);
        if(accountDetails != null) {
            model.addAttribute("accountDetails", accountDetails);
        }
        if(transactionDetails != null){
            model.addAttribute("transactionDetails", transactionDetails);
```

（6）创建 APIContext.java，定义内部类。

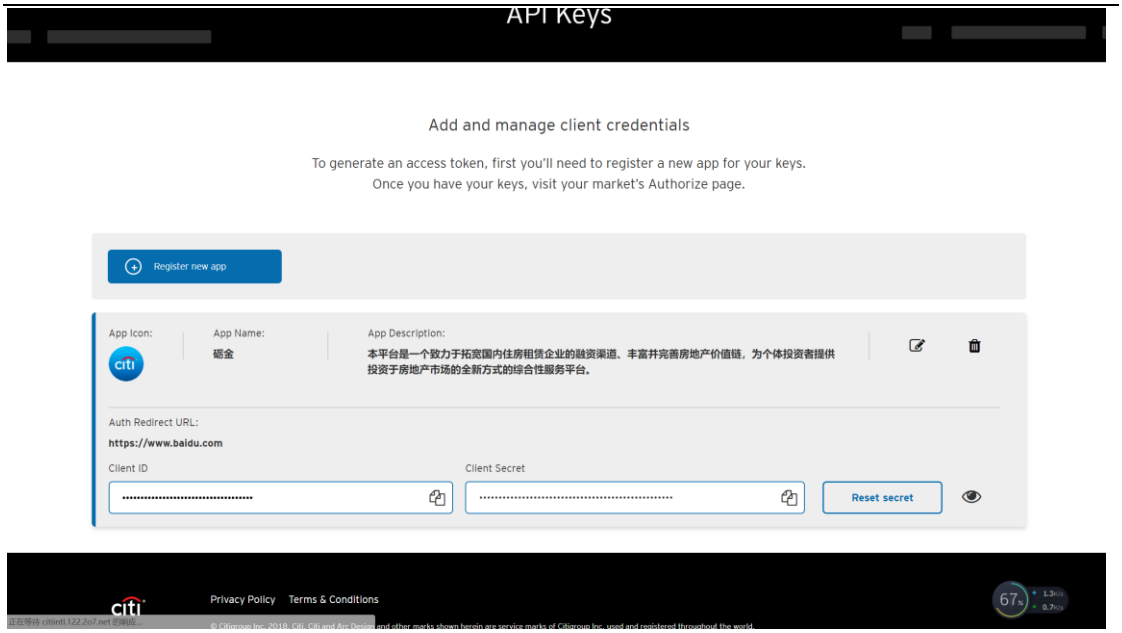
```java
package com.demo.developerapi.beans;

public class APIContext {
    private String accessToken;
    private String eventId;
    private String bizToken;
    private String realAccessToken;
    private String accounts;
    private String username;
    private String password;
    private String accountId;

    public String getAccessToken() {
        return accessToken;
    }

    public void setAccessToken(String accessToken) {
        this.accessToken = accessToken;
    }

    public String getEventId() {
        return eventId;
    }

    public void setEventId(String eventId) {
        this.eventId = eventId;
    }

    public String getBizToken() {
        return bizToken;
    }

    public void setBizToken(String bizToken) {
        this.bizToken = bizToken;
    }

    public String getRealAccessToken() {
        return realAccessToken;
    }

    public void setRealAccessToken(String realAccessToken) {
        this.realAccessToken = realAccessToken;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
```

（7）创建 APIConstant.java，放入客户机标识和客户机密匙。

```java
package com.demo.developerapi.beans;

public interface APIConstant {
    final String CLIENT_ID = "4796196b-b36a-4067-89f5-bfde2e19812f";
    final String CLIENT_SCRENT = "1H1gN6aO1fF0mM5jW0pU2kV0hS6hU5bM4cU4iO3hR8iW7wS4hG";
}
```

3. 第三部分：功能介绍

（1）获取 Client Id/Client Secret 并创建 app。



（2）获取客户端 Access Token（已实际运行功能）。

```
public static String step1GetAccessToken(APIContext context) throws IOException {
    OkHttpClient client = new OkHttpClient();
    String client_id = APIConstant.CLIENT_ID;
    String client_screnT = APIConstant.CLIENT_SCRENT;
    String encode_key = client_id + ":" + client_screnT;
    String authorization = "Basic " + Base64.encodeBase64String(encode_key.getBytes());
    MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
    RequestBody body = RequestBody.create(mediaType, "grant_type=client_credentials&scope=/api");
    Request request = new Request.Builder()
            .url("https://sandbox.apihub.citi.com/gcb/api/clientCredentials/oauth2/token/hk/gcb")
            .post(body)
            .addHeader("accept", "application/json")
            .addHeader("authorization", authorization)
            .addHeader("content-type", "application/x-www-form-urlencoded")
            .build();
    Response response = client.newCall(request).execute();
    JSONObject jsonObject = (JSONObject) JSONValue.parse(response.body().string());
    String accessToken = (String) jsonObject.get("access_token");
    context.setAccessToken(accessToken);
    System.out.println("step1 access_token:");
    System.out.println("\t" + accessToken);
    return accessToken;
}
```

返回结果：


```
{
  "token_type": "bearer",
  "access_token": "AAIkYTY5OTZkODMtNTEzYS00ZjE4LThjMGEtN2Q0MjllOGE4YmEy
    -zifgdzIFUO950GHK9mxXpO4Gy6XUgJKpF1RjQiQ1nnKHrqCn-B0llb
    -CX_buuGLOILa1sGDC12tRxsotkAd1kHWzD7H0g5UcaEhB1C_OtjU7LGUjAyI3_SlCeeqqdQm4pGnyuHsTchc1AI168m9LuY60J5_CS_x0C-AYuMSWs
    -_aMy3dml3qyxG12z_kcjWT0tJYh1LMtv4GeZ3C_GtJeZsv7eHGwvmwdG07Dq8-XU",
  "expires_in": 1800,
  "consented_on": 1531124357,
  "scope": "/api"
}
```


（3）获取 modulus/exponent/bizToken/eventId（未实际运行功能）。

```java
public static Map<String, String> step2GetBizToken(APIContext context) throws IOException {
    Map<String, String> map = new HashMap<String, String>();
    OkHttpClient client = new OkHttpClient();
    String client_id = APIConstant.CLIENT_ID;
    String accessToken = context.getAccessToken();
    String authorization = "Bearer " + accessToken;
    UUID uuid = UUID.randomUUID();
    Request request = new Request.Builder()
            .url("https://sandbox.apihub.citi.com/gcb/api/security/e2eKey")
            .get()
            .addHeader("authorization", authorization)
            .addHeader("client_id", client_id)
            .addHeader("uuid", uuid.toString())
            .addHeader("content-type", "application/json")
            .build();
    Response response = client.newCall(request).execute();
    JSONObject jsonObject = (JSONObject) JSONValue.parse(response.body().string());
    String modulus = null;
    String exponent = null;
    String bizToken = null;
    String eventId = null;
    if (jsonObject != null) {
        modulus = (String) jsonObject.get("modulus");
        exponent = (String) jsonObject.get("exponent");
        Headers headers = response.headers();
        bizToken = headers.get("bizToken");
        eventId = headers.get("eventId");
        map.put("modulus", modulus);
        map.put("exponent", exponent);
        map.put("bizToken", bizToken);
        map.put("eventId", eventId);
        context.setEventId(eventId);
        context.setBizToken(bizToken);
    }
    System.out.println("step2 map:");
    for (String s : map.keySet()) {
        System.out.println("\tkey:" + s + "\tvalues:" + map.get(s));
    }
    return map;
}
```

（4）获取登陆 Access Token（已实际运行功能）。

```java
public static String step3GetRealAccessToken(APIContext context) throws IOException{
    String client_id = APIConstant.CLIENT_ID;
    String client_scrent = APIConstant.CLIENT_SCRENT;
    String bizToken = context.getBizToken();
    System.err.println("bizToken: "+bizToken);
    String encode_key = client_id + ":" + client_scrent;
    String authorization = "Basic " + Base64.encodeBase64String(encode_key.getBytes());
    String username = context.getUsername();
    String password = context.getPassword();
    System.out.println(password);
    UUID uuid = UUID.randomUUID();
    OkHttpClient client = new OkHttpClient();
    MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
    RequestBody body = RequestBody.create(mediaType, "grant_type=password&scope=/api&username="+username+"&password="+password);
    Request request = new Request.Builder()
            .url("https://sandbox.apihub.citi.com/gcb/api/password/oauth2/token/hk/gcb")
            .post(body)
            .addHeader("authorization", authorization)
            .addHeader("bizToken", bizToken)
            .addHeader("uuid", uuid.toString())
            .addHeader("content-type", "application/x-www-form-urlencoded")
            .addHeader("accept", "application/json")
            .build();
    Response response = client.newCall(request).execute();
    JSONObject jsonObject = (JSONObject) JSONValue.parse(response.body().string());
    String realAccessToken = (String) jsonObject.get("access_token");
    context.setRealAccessToken(realAccessToken);
    System.out.println("step3 real_access_token:");
    System.out.println("\t" + realAccessToken);
    return realAccessToken;
}
```

返回结果：

```
{
  "token_type": "bearer",
  "access_token": "AAIkYTY5OTZkODMtNTEzYS00ZjE4LThjMGEtN2Q0MjllOGE4YmEyCfFYYVuHF55cui6nJU0XHwOfEgN7YKT2dQmi8GUEybZn7ppw
    B9v7TCgRGLNS3fQ3zmg9-smTPZ0jUe5BGMEGF06LFf0PMcJTPtcqMrD5WEQvb7bF_DkWbiKD8mthkiqvNrL9RGrTz2bsbOMgvnwybr8qPw_UVg6uhUb
    kImvT3S1_dvHaLn6REDQVLGBtj0-xpmCnFHpH4i2KkqqxwPcGRQ",
  "expires_in": 1800,
  "consented_on": 1531124360,
  "scope": "/api",
  "refresh_token": "AAI1Lq8is2srF29u6lTiyzi-o7JYZk2mm9BXwtPeu9oNLFBzv7rkn3JE0SeWsxof
    -P9yQ3Xll2znMNEMQGpIFLLxAhtxUZMfDZGVvyLH8cffBqz1YEBmdtCShsbu6cCItakx-hXTUfhEfmE
    -oD8JgTRgdBgArfFDuqfg4j25Rf3KjVbiqXskM6EzP8kPxmtlHVE15ZPr_buozuFTrjXKOYYTKwFpvjvKbYLXC85S-tNIeQ",
  "refresh_token_expires_in": 2592000
}
```


（5）获取账户信息（未实际运行功能）。

```
public static String step4GetAccounts(APIContext context) throws IOException{
    String client_id = APIConstant.CLIENT_ID;
    String authorization = "Bearer " + context.getRealAccessToken();
    UUID uuid = UUID.randomUUID();
    OkHttpClient client = new OkHttpClient();
    Request request = new Request.Builder()
            .url("https://sandbox.apihub.citi.com/gcb/api/v1/accounts")
            .get()
            .addHeader("authorization", authorization)
            .addHeader("uuid", uuid.toString())
            .addHeader("content-type", "application/json")
            .addHeader("accept", "application/json")
            .addHeader("client_id", client_id)
            .build();
    Response response = client.newCall(request).execute();
    String responseBodyString = response.body().string();
    context.setAccounts(responseBodyString);
    System.out.println("step4 accounts:");
    System.out.println("\t"+responseBodyString);
    return responseBodyString;
}
```

（6）获取账户的详细信息（未实际运行功能）。

```
public static String step5GetAccountDetails(APIContext context) throws IOException{
    String client_id = APIConstant.CLIENT_ID;
    String authorization = "Bearer " + context.getRealAccessToken();
    UUID uuid = UUID.randomUUID();
    String accountId = context.getAccountId();
    OkHttpClient client = new OkHttpClient();
    Request request = new Request.Builder()
            .url("https://sandbox.apihub.citi.com/gcb/api/v1/accounts/"+accountId)
            .get()
            .addHeader("authorization", authorization)
            .addHeader("uuid", uuid.toString())
            .addHeader("content-type", "application/json")
            .addHeader("accept", "application/json")
            .addHeader("client_id", client_id)
            .build();
    Response response = client.newCall(request).execute();
    String responseBodyString = response.body().string();
    context.setAccounts(responseBodyString);
    System.out.println("step5 account details:");
    System.out.println("\t"+responseBodyString);
    return responseBodyString;
}
```

（7）获取事务信息（未实际运行功能）。

```java
    public static String step6GetTransaction(APIContext context) throws IOException{
        String client_id = APIConstant.CLIENT_ID;
        String authorization = "Bearer " + context.getRealAccessToken();
        UUID uuid = UUID.randomUUID();
        String accountId = context.getAccountId();
        OkHttpClient client = new OkHttpClient();
        Request request = new Request.Builder()
                .url("https://sandbox.apihub.citi.com/gcb/api/v1/accounts/"+accountId+"/transactions")
                .get()
                .addHeader("authorization", authorization)
                .addHeader("uuid", uuid.toString())
                .addHeader("content-type", "application/json")
                .addHeader("accept", "application/json")
                .addHeader("client_id", client_id)
                .build();
        Response response = client.newCall(request).execute();
        String responseBodyString = response.body().string();
        context.setAccounts(responseBodyString);
        System.out.println("step6 transaction details:");
        System.out.println("\t"+responseBodyString);
        return responseBodyString;
    }
}
```

4. 第四部分：验证测试

砺金平台根据花旗开发者门户网站的功能介绍和代码示例，已经将大部分功能的代码编写完成，并如上第三部分所示，将所有功能封装成了类，只要在后台实现调用并可运行功能。由于平台业务与 API 的弱关联性及时间的紧迫性，本项目仅选取一个功能（实现对花旗用户的授权获取 access_token）进行验证测试，并成功返回结果。

测试代码如下：

```java
import java.io.IOException;

import org.apache.commons.codec.binary.Base64;

import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
import okhttp3.Headers;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class API {

    OkHttpClient client = new OkHttpClient();

    String run(String url) throws IOException {
    String client_id = "4796196b-b36a-4067-89f5-bfde2e19812f";
    String client_scrent = "1H1gN6aO1fF0mM5jWOpU2kV0hS6hU5bM4cU4iO3hR8iW7wS4hG";
    String encode_key = client_id + ":" + client_scrent;
    String authorization = "Basic " + Base64.encodeBase64String(encode_key.getBytes());

    MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
    RequestBody body = RequestBody.create(mediaType, "grant_type=client_credentials&scope=/api");
    Request request = new Request.Builder()
        .url("https://sandbox.apihub.citi.com/gcb/api/clientCredentials/oauth2/token/hk/gcb")
        .post(body)
        .addHeader("accept", "application/json")
        .addHeader("authorization", authorization)
        .addHeader("content-type", "application/x-www-form-urlencoded")
        .build();
    try (Response response = client.newCall(request).execute()){
     return response.body().string();
    }
}
    public static void main(String[] args) throws IOException {
        API api = new API();
        String response = api.run("https://sandbox.apihub.citi.com/gcb/api/clientCredentials/oauth2/token/hk/gcb");
        System.out.println(response);
    }
}
```

结果如下：

```java
import java.io.IOException;

import org.apache.commons.codec.binary.Base64;

import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
import okhttp3.Headers;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class API {

    OkHttpClient client = new OkHttpClient();

    String run(String url) throws IOException {
    String client_id = "4796196b-b36a-4067-89f5-bfde2e19812f";
    String client_scrent = "1H1gN6aOlfF0mM5jW0pU2kV0hS6hU5bM4cU4iO3hR8iW7wS4hG";
    String encode_key = client_id + ":" + client_scrent;
    String authorization = "Basic " + Base64.encodeBase64String(encode_key.getBytes());

    MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
    RequestBody body = RequestBody.create(mediaType, "grant_type=client_credentials&scope=/api");
```

```
<terminated> API [Java Application] D:\java\jdk\bin\javaw.exe (2018年9月16日 上午11:51:13)
{ "token_type":"bearer", "access_token":"AAIkNDc5NjE5NmItYjM2YS00MDY3LTg5ZjUtYmZkZTJlMTk4MTJmaa9AExtq-kxO3_PBNnpoaCsxhu4-vYVJHlYGc7L7oVotDQnzntiDg7ScbXVZJPZ
```

可以看到，系统已经成功返回 access_token；至此，砺金平台成功实现对花旗 API 的调用功能。