

Payeezy Direct API Integration Guide

April 2015

Contents

Quickly integrate your applications with Payeezy Direct API	3
Payeezy Direct API initialization	3
Steps to integrate code with Payeezy Direct API	3
Getting started.....	3
Payeezy Integration for Java binding.....	4
Submitting/Generating transaction (method of payments) with example.....	7
Additional capabilities API (if applicable)	7
Security related (HMAC, Token generation) with example	8

Quickly integrate your applications with Payeezy Direct API

If you want to enable secure and convenient payments to your payment applications, this guide will get you up and running quickly. Payeezy handles all the heavy lifting of the complex tokenization and protects your customers' transactions. It is simple to create a developer test account and apply for a merchant account through our developer portal. Payeezy Direct API provides client libraries for Java, PHP, Python, ruby, nodeJS and curl programming languages.

PayeezySM, from First Data, is part of the **Small Business Solutions Suite**, which includes CloverTM, InsighticsSM, PerkaTM and TransArmor®. The **Payeezy eCommerce Solution** empowers SMBs to expand their horizons and easily grow their business online or via mobile by reaching new customers no matter where they are.

This document refers to the Direct API integration method included within the overall Payeezy eCommerce Solution. Henceforth, all references to Payeezy are in relation to Payeezy APIs.

Payeezy Direct API initialization

Please go through **prerequisites**, to get an overview of our developer portal. Link for prerequisites

https://github.com/payeezy/get_started_with_payeezy/blob/master/get_started_with_payeezy042015.pdf

Note: The merchant/third party is responsible for PCI compliance and may be required to provide First Data with additional documentation and/or their attestation of compliance

Steps to integrate code with Payeezy Direct API

1. Download the Payeezy Direct API Client library from GitHub
2. Define merchant Token, Apikey and ApiSecret (respective file)
3. Call transaction method and pass the required parameters

For test credit card/eCheck (telecheck) test data, refer

https://github.com/payeezy/testing_payeezy/blob/master/payeezy_testdata042015.pdf

Getting started

Using GitHub

Clone Payeezy SDK using with HTTPS or Subversion

Using clone command: `git clone https://github.com/payeezy/payeezy_direct_API.git`

Or simply download zip file: https://github.com/payeezy/payeezy_direct_API/archive/master.zip.

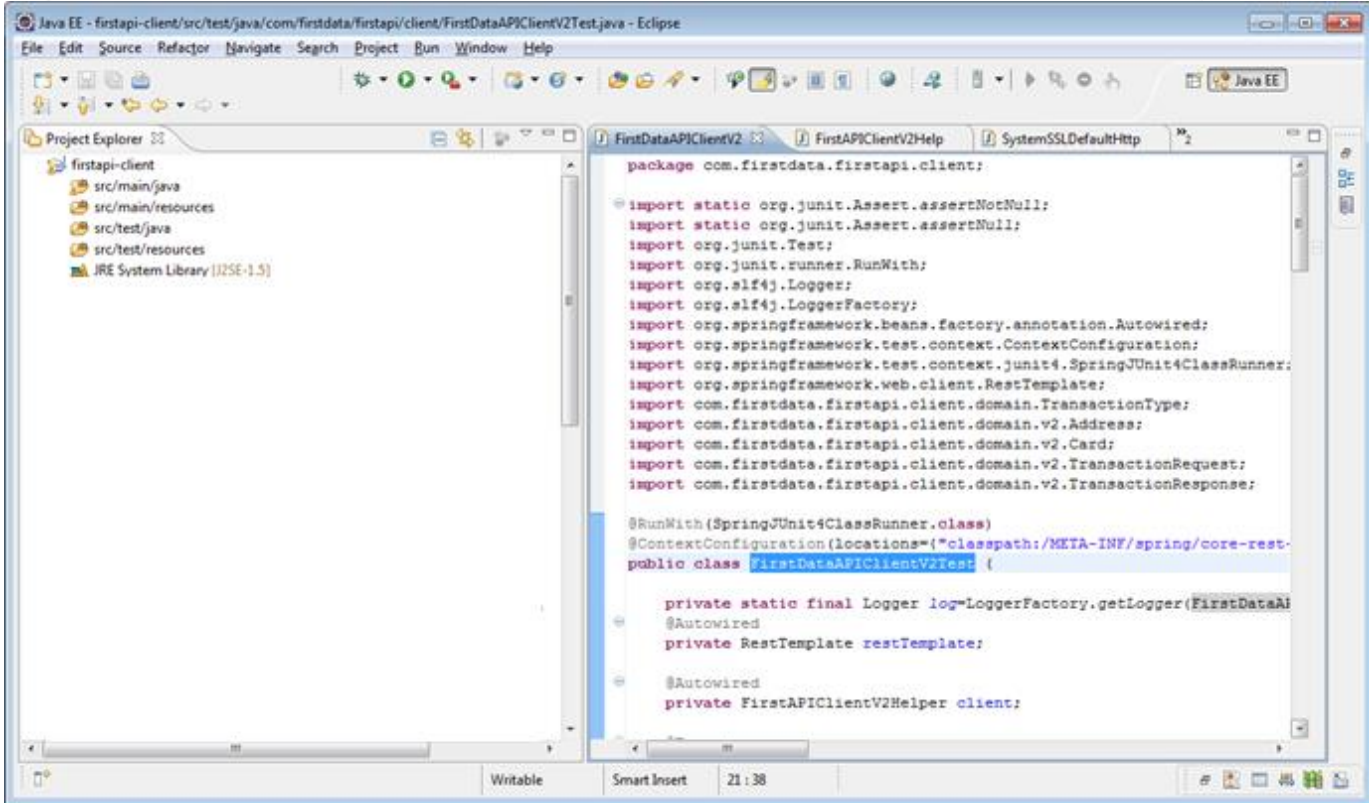
To know more about GitHub, click <http://github.com>

Payeezy Integration for Java binding

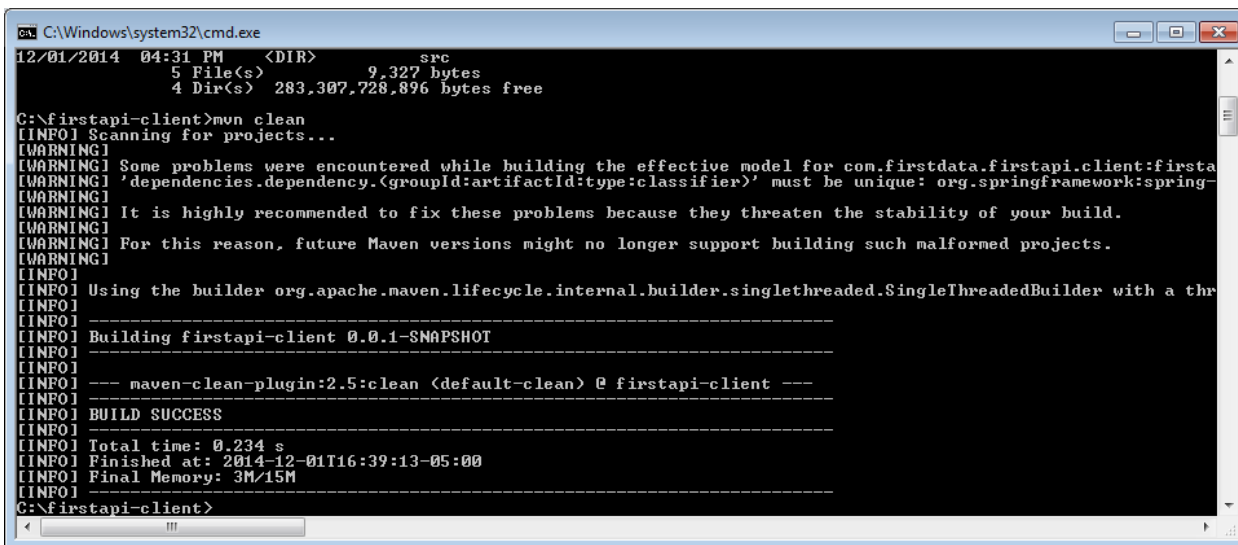
Build/install - Maven dependencies

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>${spring-version}</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>2.5</version>
</dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.6.4</version>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>1.6.4</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>3.2.8.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-mapper-asl</artifactId>
    <version>1.8.8</version>
</dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.3.2</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.10</version>
    <scope>test</scope>
  </dependency></dependencies>
```

You can also import code as Java project



Note that you need to run maven command in order to compile the code.



Customize

Update properties file.

Please go to \$PAYEEZY \src\main\resources\META-INF\spring\core-rest-client-v2-context.xml
And update with your appid, securedSecret and token

Where

For example: \$PAYEEZY = C:\firstapi-client

Note that

securedSecret = API Secret

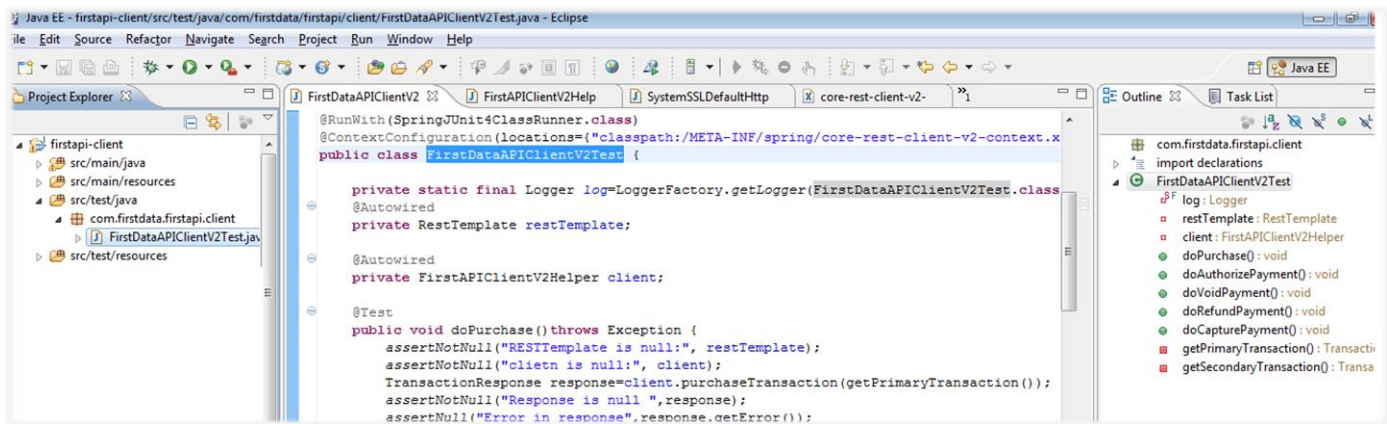
appid = API Key

token= Merchant token

```
</bean>
```

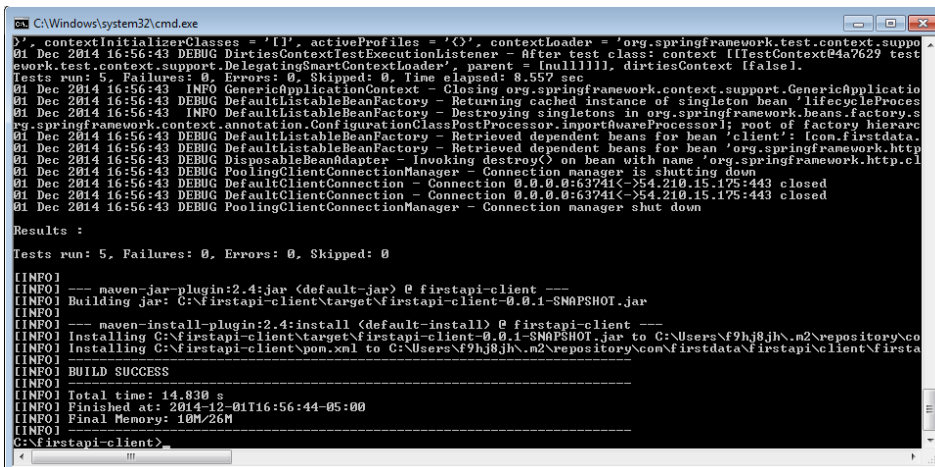
```
<bean id="client" class="com.firstdata.firstapi.client.FirstAPIClientV2Helper">
  <property name="url" value="https://api-cert.payeezy.com/v1" />
  <property name="appid" value="PBfqmeTMVs6yezIYz3TBmABBJvNhsW8t" />
  <property name="securedSecret" value="bf5e9fc499cb523db1a28606fd5bcfa3c6efa5f85ab627fa68e9a4059a05857b" />
  <property name="token" value="fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6" />
</bean>
```

Please go through \$PAYEEZY \src\test\java\com\firstdata\firstapi\client\FirstDataAPIClientV2Test.java



Use/execute

Please go to download directory and run \$ mvn clean install



For Payeezy ruby, PHP, Python, nodeJS bindings, refer to GitHub@ https://github.com/payeezy/payeezy_direct_API

Submitting/Generating transaction (method of payments) with example

Payeezy supports the following method of payments

- Credit Card Payments
- PayPal Transactions
- Gift Card (via ValueLink) Transactions
- eCheck (via TeleCheck) Transactions
- 3D Secure Transactions

For API processing details, click here [Method of Payments](#)

Or click on the link <https://developer.payeezy.com/payeezy-api-reference/apis>

Additional capabilities API (if applicable)

- Partner Reporting API - Use our powerful query engine to retrieve payment records. Supports complex filtering, sorting, pagination and more. This is exclusively for Third Party Partners and applicable for a **live** environment only.

For Reporting API processing details, click here <https://developer.payeezy.com/payeezy-api-reference/apis/get/transactions>

Security related (HMAC, Token generation) with example

Construct the data param by appending the parameters below in the same order as shown.

- a. apikey - API key of the developer.
- b. nonce - secure random number.
- c. timestamp – epoch timestamp in milliseconds.
- d. token - Merchant Token.
- e. payload - Actual body content passed as post request. Compute HMAC SHA256 hash on the above data param using the key below
- f. apiSecret - Consumer Secret token for the given api key Calculate the base64 of the hash which would be our required Authorization header value.

```
public String getMacValue(Map<String,String> data) throws Exception{
    Mac mac=Mac.getInstance("HmacSHA256");
    String apiSecret= data.get(APISECRET);
    log.debug("API_SECRET:{},apiSecret);
    SecretKeySpec secret_key = new SecretKeySpec(apiSecret.getBytes(), "HmacSHA256");
    mac.init(secret_key);
    StringBuilder buff=new StringBuilder();
    buff.append(data.get(APIKEY)).append(data.get(NONCE)).append(data.get(TIMESTAMP));
    if(data.get(TOKEN)!=null)
        buff.append(data.get(TOKEN));
    if(data.get(PAYLOAD)!=null)
        buff.append(data.get(PAYLOAD));
    log.info(buff.toString());
    byte[] macHash=mac.doFinal(buff.toString().getBytes("UTF-8"));
    log.info("MacHash:{},Arrays.toString( macHash));
    String authorizeString=Base64.encodeBase64String(toHex(macHash));
    log.info("Authorize: {}",authorizeString);
    return authorizeString;
}
```

For more details refer `com.firstdata.firstapi.client.FirstAPIClientV2Helper.java` class