Payeezy

Payeezy Integration

Payeezy support Java, php, python, ruby , nodeJS and curl

Tools & RESOURCES

Download and build

- **Github** – For Payeezy source code downloads – visit https://github.com/
- **Eclipse** – For client library integration and development – for more information visit https://eclipse.org/
- **IntelliJ** – For client library integration and development – for more information please visit https://www.jetbrains.com/idea/
- **JDK** 1.6 and above - For compiling java code – for more information please visit http://www.oracle.com/technetwork/java/javase/downloads/index.html
- **Text editor** - if you don't have one already, you may want one for editing properties file. A few options: Atom, Sublime Text, Textmate, Brackets.

Package Managers

- **Maven** – For build and test – for more information please visit http://maven.apache.org/
- **Pip** - pip is a package management system used to install and manage software packages written in Python. If your system doesn't have pip, you can also use **easy_install**
- **Gem** - RubyGems is a package management framework for Ruby. For more information Pl visit http://guides.rubygems.org/rubygems-basics/
- **npm** - is the package manager for javascript, nodeJS etc. For more information pl visit https://www.npmjs.com/

Download

Download sample code from GitHub https://github.com/payeezy/payeezy/tree/master/java

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>
```
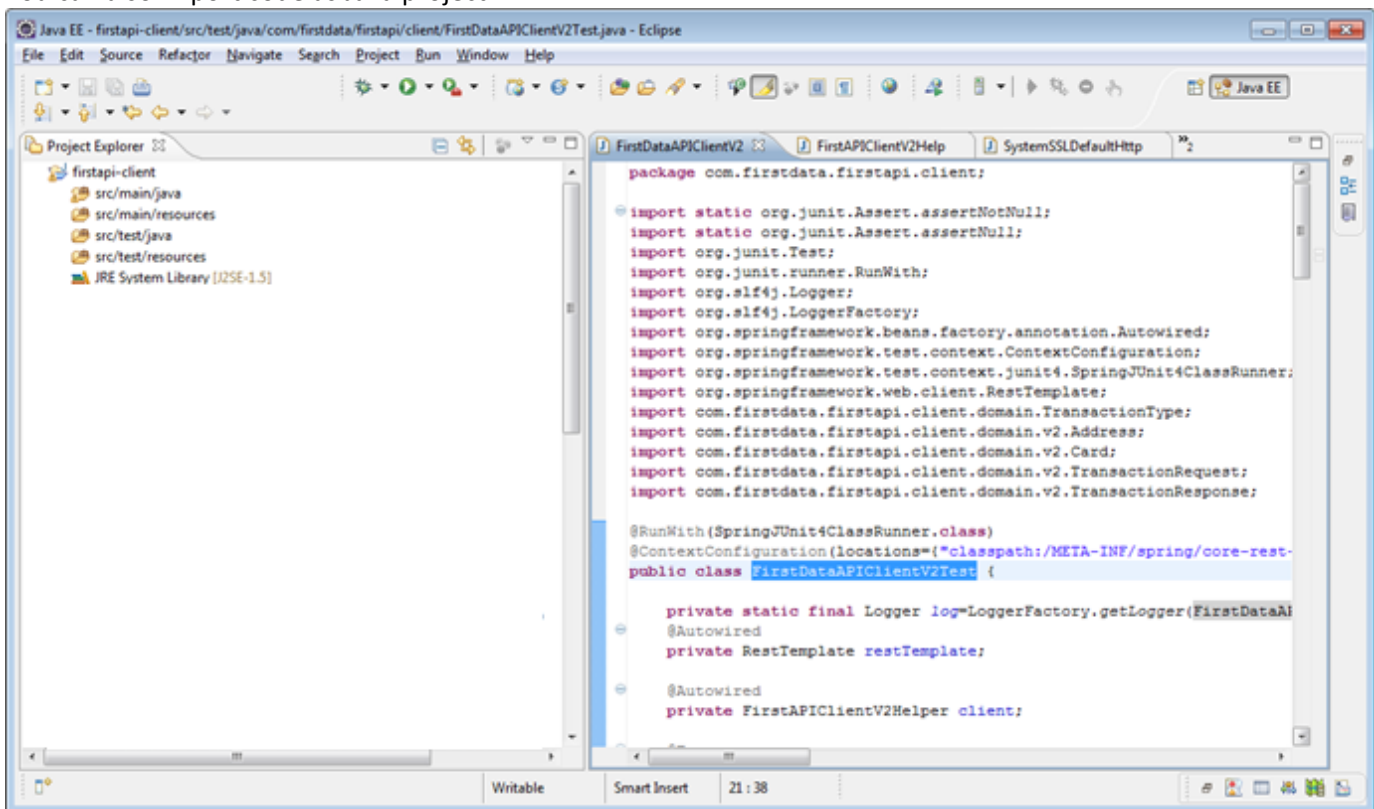
You can also import code as Java project

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

```
C:\Windows\system32\cmd.exe
12/01/2014  04:31 PM    <DIR>          src
              5 File(s)          9,327 bytes
              4 Dir(s)   283,307,728,896 bytes free

C:\firstapi-client>mvn clean
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for com.firstdata.firstapi.client:firsta
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must be unique: org.springframework:spring-
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] Using the builder org.apache.maven.lifecycle.internal.builder.singlethreaded.SingleThreadedBuilder with a thr
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building firstapi-client 0.0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ firstapi-client ---
[INFO]
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.234 s
[INFO] Finished at: 2014-12-01T16:39:13-05:00
[INFO] Final Memory: 3M/15M
[INFO] ------------------------------------------------------------------------
C:\firstapi-client>
```

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

```xml
    </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>
```
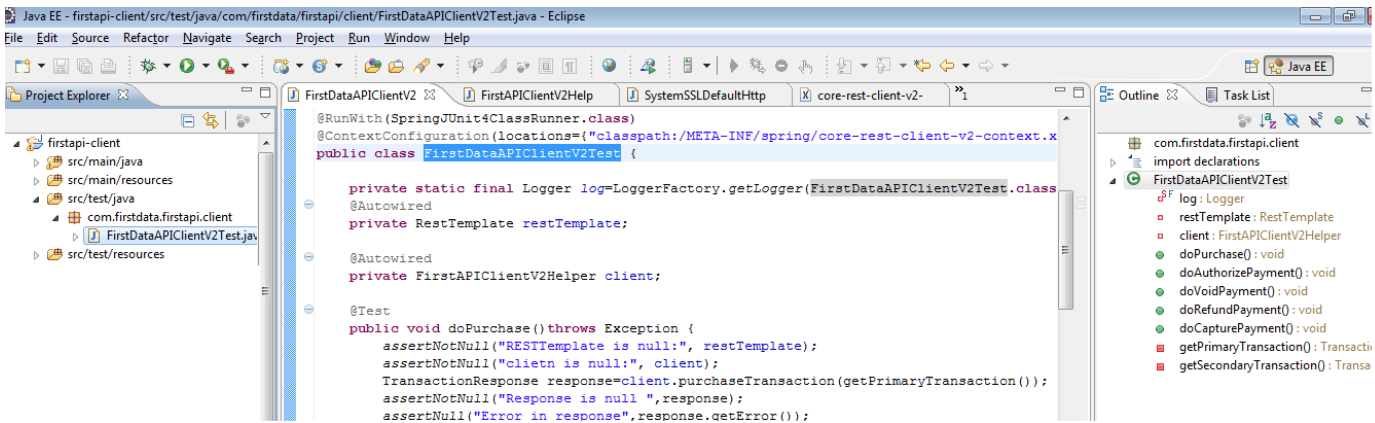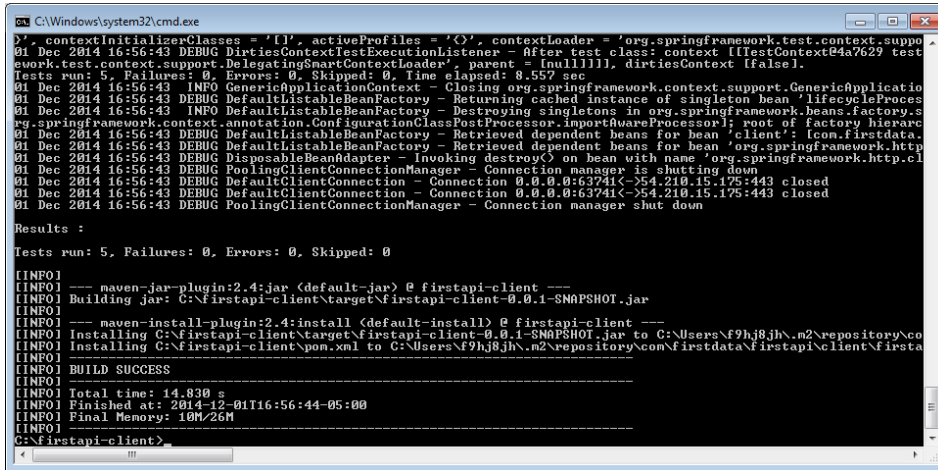
Please go through $PAYEEZY \src\test\java\com\firstdata\firstapi\client**FirstDataAPIClientV2Test.java**

Use/execute

Please go to download directory and run $ mvn clean install

Ruby Integration

Installation

Add this line to your application's Gemfile:

```
gem 'payeezy'
```

And then execute:

```
$ bundle
```

Or install it yourself as:

```
$ gem install payeezy
```

Usage

Run the specs using 'bundle exec rake spec'

Test cases can be run with 'bundle exec rake test'

Python

This is version payeezy 2.0 with which you could use transactions into your
project using following steps

Download and install Requests 2.3 from the link below which is a dependancy for
Payeezy.
https://github.com/kennethreitz/requests

1. unzip the folder and open cmd / terminal
2. use 'cd' command to payeezy folder

3. run 'python setup.py build'
4. run 'python setup.py install'
5. Now you should be able to run example.py using ' python example.py'

Using payeezy with Django or any other python script support platform

use statement 'import payeezy' to be able to call the methods in transactions#

Setting attributes:

```
payeezy.apikey= ""
payeezy.apisecret= ""
payeezy.token = ""
payeezy.url = ""
```