

Serkan Berk Bilgiç – 71571 - sbilgic19@ku.edu.tr

- There are 2 types of request is allowed in my protocol. First one, in below fetches all the NFTs and lists them.

sbbp -m/GET -r/NFTServer -v/1.0

- The other one is to fetch information for the specific NFTs by ids. Here I designed in such a way that single or multiple ids can be given simultaneously and the result returned from the NFTMultithreadedServer will show the information such as **name**, **asset_platform_id**, and **price** for all the given ids.

sbbp -m/GET -r/NFTServer -v/1.0 -i/{nft_id1, nft_id2, nft_id3, ...}

As an example for the second case I can give the following request and you can see the result returned.

sbbp -m/GET -r/NFTServer -v/1.0 -i/{ethlings, squiggly}

Note: -m represents method type, -r represents resource type, -v represents version and I did not extend but for generality I wanted to design this way and if you do not provide any of this 3 with the exact -m|-r|-v/ notation or say you request -v/1.1 or -r/CoinSomething it will give an error. In the below you can find examples for each of them.

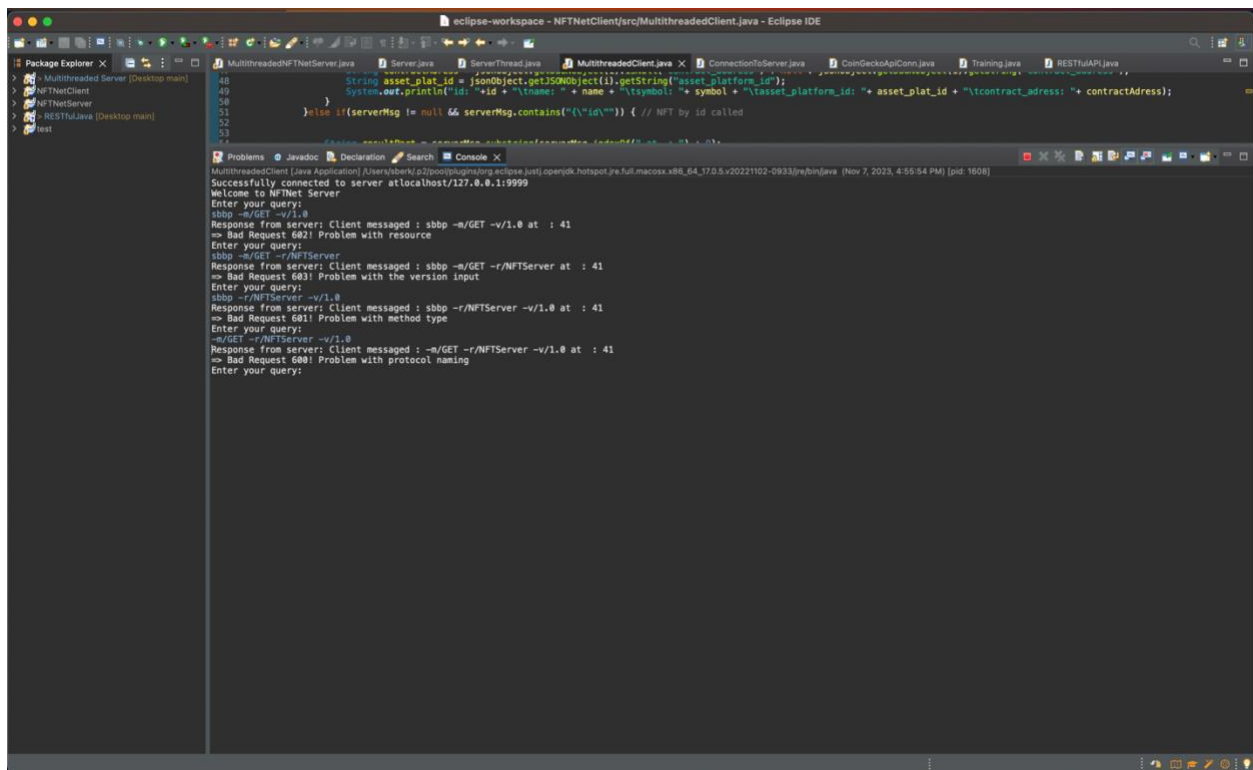
Client Side Cases:

Some Problematic Cases : For now only GET method exist and the only resource is NFTServer and only version is 1.0 and -m|-r|-v/ notation must be used. In the below you can see examples of different wrong request such as not providing resource

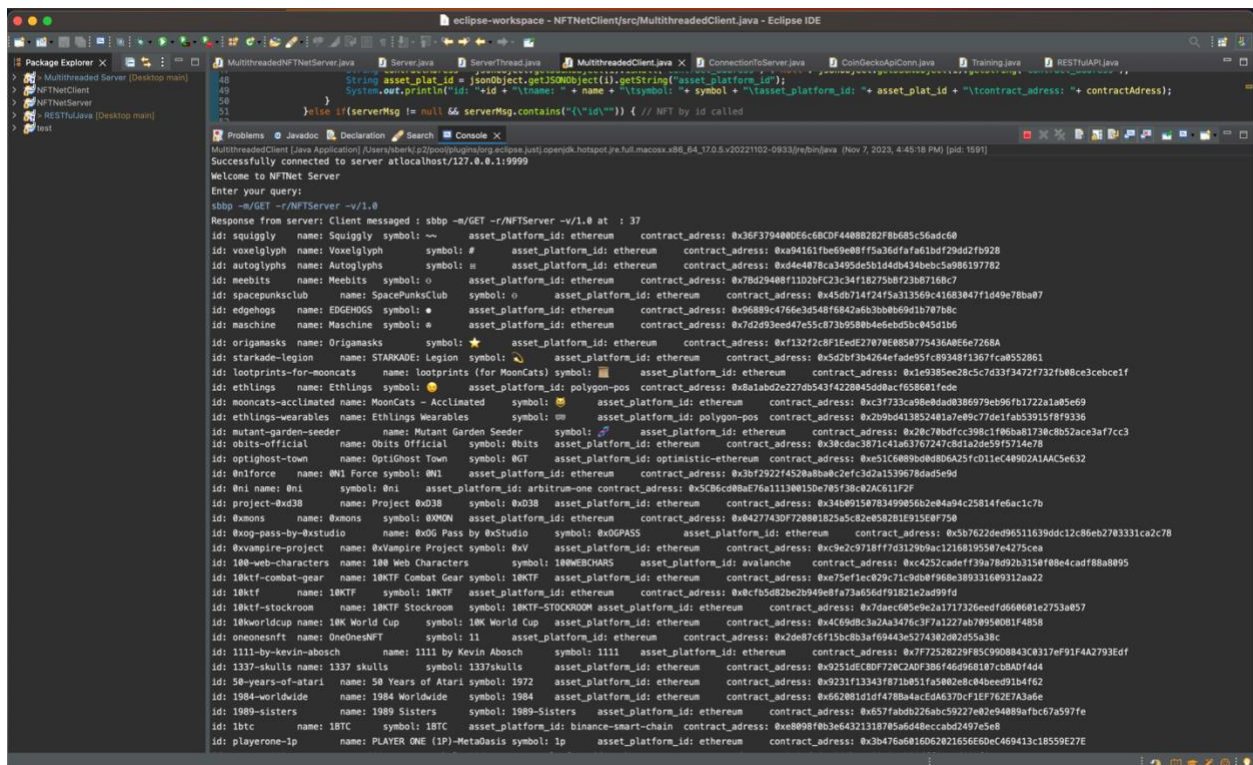
1: Adding noise to the any part of the request for example sbbps or -v123 or adding .. at the end all will result in Unexpected Bad Request with code 605. I made up codes this is the most general error code.

```
Enter your query:
sbbp -m/GET -r/NFTServer -v123/1.0
Response from server: Client messaged : sbbp -m/GET -r/NFTServer -v123/1.0 at : 37
=> Unexpected Bad Request 605!
Enter your query:
```

2: Not providing protocol_name (sbbp), resource, method or version or providing unavailable for any of them will give different error codes. For protocol name which is exactly sbbp it is 600, for method type(only GET method is available) 601, for resource(only NFTServer available) 602, for version(only 1.0 is available) 603. You can see examples for each of them in the below screenshot.



3: If correct request is made and -i flag is not provided it will list all the results.



4: If multiple ids provided the result will be returned for each

```

eclipse-workspace - NFTNetClient/src/MultithreadedClient.java - Eclipse IDE

MultithreadedClient.java
48 String asset_plat_id = jsonObject.getString("asset_platform_id");
49 System.out.println("id: " + id + "name: " + name + "symbol: " + symbol + "asset_platform_id: " + asset_plat_id + "contract_address: " + contractAddress);
50 }
51 } else if (serverMsg != null && serverMsg.contains("\\\\id\\\\")) { // NFT by id called
52
53
54 String resultPart = serverMsg.substring(serverMsg.indexOf("at: ") + 9);
55 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
56 System.out.println("Response from server: " + initialHandShaking);
57
58 String[] strArr = resultPart.split("\\\\");
59 for (int k=0; k<strArr.length; k++) {
60     JSONObject jsonObject = new JSONObject(strArr[k]);
61
62     String id = jsonObject.getString("id");
63     String asset_platform_id = jsonObject.getString("asset_platform_id");
64     String name = jsonObject.getString("name");
65
66     JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
67     Double floor_price_usd = floor_price.getDouble("usd");
68     Double floor_price_native_currency = floor_price.getDouble("native_currency");
69 }
70 }

Problems | JavaDoc | Declaration | Search | Console
MultithreadedClient [Java Application] /Users/ethan/p2/p2nftclient/src/MultithreadedClient.java openjdk.hotspot.jre.full/macosx-a64-17.0.5-v20221102-0933prebin (Nov 7, 2023, 4:33:00 PM) [pid: 1523]
Successfully connected to server at local host/127.0.0.1:9999
Welcome to NFTNet Server
Enter your query:
sbpp -M/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly}
Response from server: Client messaged: sbpp -M/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly} at : 22
id: ethlings asset_platform_id: polygon-pos name: Ethlings floor_price_usd: 93.89 floor_price_native_currency: 0.85 market_cap_usd: 727786.0 market_cap_native_currency: 387.55
id: squiggly asset_platform_id: ethereum name: Squiggly floor_price_usd: 28145.0 floor_price_native_currency: 15.0 market_cap_usd: 2814450.0 market_cap_native_currency: 15.0
Enter your query:

```

5- If single id provided the result will be returned for each

```

eclipse-workspace - NFTNetClient/src/MultithreadedClient.java - Eclipse IDE

MultithreadedClient.java
48 String asset_plat_id = jsonObject.getString("asset_platform_id");
49 System.out.println("id: " + id + "name: " + name + "symbol: " + symbol + "asset_platform_id: " + asset_plat_id + "contract_address: " + contractAddress);
50 }
51 } else if (serverMsg != null && serverMsg.contains("\\\\id\\\\")) { // NFT by id called
52
53
54 String resultPart = serverMsg.substring(serverMsg.indexOf("at: ") + 9);
55 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
56 System.out.println("Response from server: " + initialHandShaking);
57
58 String[] strArr = resultPart.split("\\\\");
59 for (int k=0; k<strArr.length; k++) {
60     JSONObject jsonObject = new JSONObject(strArr[k]);
61
62     String id = jsonObject.getString("id");
63     String asset_platform_id = jsonObject.getString("asset_platform_id");
64     String name = jsonObject.getString("name");
65
66     JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
67     Double floor_price_usd = floor_price.getDouble("usd");
68     Double floor_price_native_currency = floor_price.getDouble("native_currency");
69 }
70 }

Problems | JavaDoc | Declaration | Search | Console
MultithreadedClient [Java Application] /Users/ethan/p2/p2nftclient/src/MultithreadedClient.java openjdk.hotspot.jre.full/macosx-a64-17.0.5-v20221102-0933prebin (Nov 7, 2023, 4:33:00 PM) [pid: 1523]
Successfully connected to server at local host/127.0.0.1:9999
Welcome to NFTNet Server
Enter your query:
sbpp -M/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly}
Response from server: Client messaged: sbpp -M/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly} at : 22
id: ethlings asset_platform_id: polygon-pos name: Ethlings floor_price_usd: 93.89 floor_price_native_currency: 0.85 market_cap_usd: 727786.0 market_cap_native_currency: 387.55
id: squiggly asset_platform_id: ethereum name: Squiggly floor_price_usd: 28145.0 floor_price_native_currency: 15.0 market_cap_usd: 2814450.0 market_cap_native_currency: 15.0
Enter your query:
sbpp -M/GET -r/NFTServer -v/1.0 -l/{voxelglyph}
Response from server: Client messaged: sbpp -M/GET -r/NFTServer -v/1.0 -l/{voxelglyph} at : 22
id: voxelglyph asset_platform_id: ethereum name: Voxelglyph floor_price_usd: 9169.32 floor_price_native_currency: 4.89 market_cap_usd: 1684632.0 market_cap_native_currency: 4.89
Enter your query:

```

6- If any of the id is incorrect it will return an error which says “Something unexpected happened. Please check your NFT ids”.

```
Package Explorer X  MultithreadedNFTServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java
> Multithreaded Server [Desktop main]
> NFTNetClient
> NFTNetServer
> RESTfulJava [Desktop main]
> test

48 String asset_plat_id = jsonObject.getString("asset_platform_id");
49 System.out.println("id: " + id + " \tname: " + name + " \tsymbol: " + symbol + " \tasset_platform_id: " + asset_plat_id + " \tcontract_adress: " + contractAddress);
50 }
51 }else if(serverMsg != null && serverMsg.contains("\\id\\")) { // NFT by id called
52
53
54 String resultPart = serverMsg.substring(serverMsg.indexOf(" at : ") + 9);
55 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
56 System.out.println("Response from server: " + initialHandShaking);
57
58 String[] strArr = resultPart.split("\\-");
59 for(int k=0; k<strArr.length; k++) {
60     JSONObject jsonObject = new JSONObject(strArr[k]);
61
62     String id = jsonObject.getString("id");
63     String asset_platform_id = jsonObject.getString("asset_platform_id");
64     String name = jsonObject.getString("name");
65
66     JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
67     Double floor_price_usd = floor_price.getDouble("usd");
68     Double floor_price_native_currency = floor_price.getDouble("native_currency");
69 }

Problems  Javadoc  Declaration  Search  Console X
MultithreadedClient [Java Application] [Javerts@berk:/p2pool/p2poolorg eclipse just; openjdk.hotspot.jre.full.macosx.x86_64_17.0.5.v20221102-0933/re/bin/java (Nov 7, 2023, 4:42:34 PM) [pid: 1576]]
Successfully connected to server atlocalhost/127.0.0.1:9999
Welcome to NFTNet Server
Enter your query:
sbpp -m/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly}
Response from server: Client messaged : sbpp -m/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly} at : 32
Something unexpected happened. Please check your NFT ids
Enter your query:
```

7- If no id is provided but -i flag is given it returns an error which says “Something unexpected happened. Please check your NFT ids”.

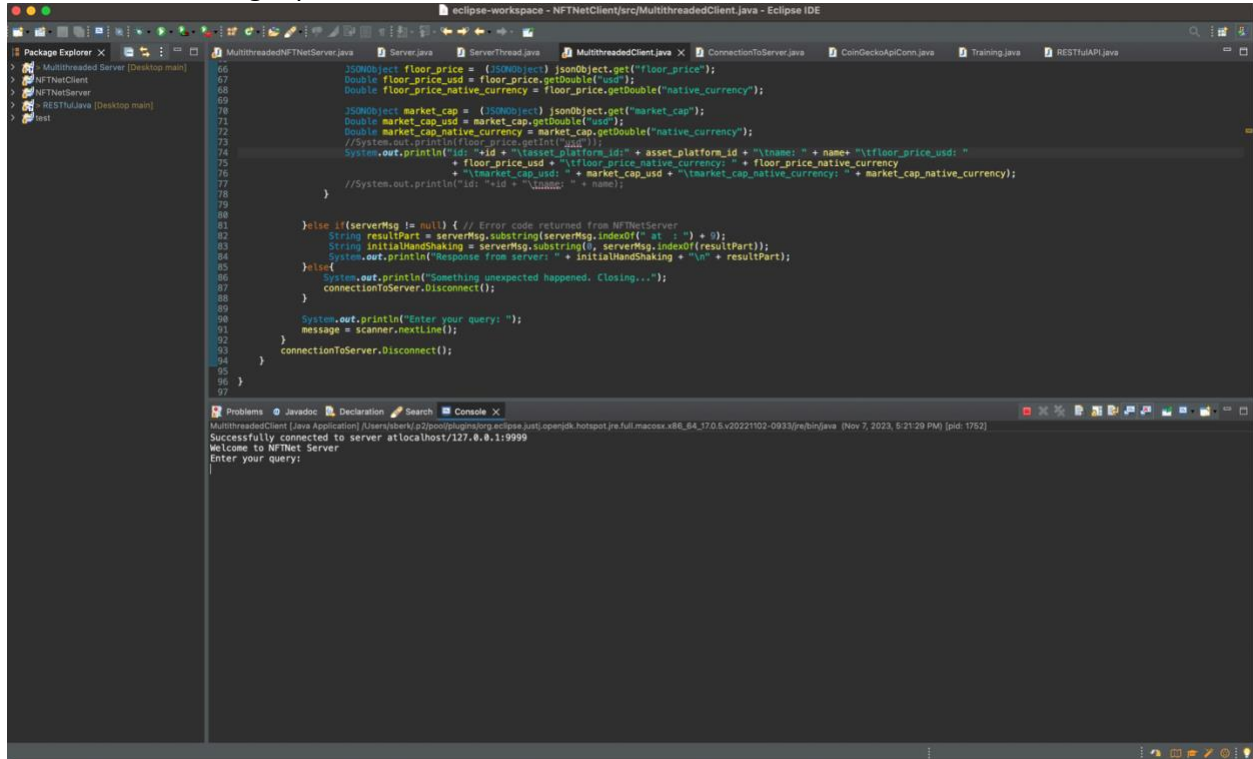
```
Package Explorer X  MultithreadedNFTServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java
> Multithreaded Server [Desktop main]
> NFTNetClient
> NFTNetServer
> RESTfulJava [Desktop main]
> test

48 String asset_plat_id = jsonObject.getString("asset_platform_id");
49 System.out.println("id: " + id + " \tname: " + name + " \tsymbol: " + symbol + " \tasset_platform_id: " + asset_plat_id + " \tcontract_adress: " + contractAddress);
50 }
51 }else if(serverMsg != null && serverMsg.contains("\\id\\")) { // NFT by id called
52
53
54 String resultPart = serverMsg.substring(serverMsg.indexOf(" at : ") + 9);
55 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
56 System.out.println("Response from server: " + initialHandShaking);
57
58 String[] strArr = resultPart.split("\\-");
59 for(int k=0; k<strArr.length; k++) {
60     JSONObject jsonObject = new JSONObject(strArr[k]);
61
62     String id = jsonObject.getString("id");
63     String asset_platform_id = jsonObject.getString("asset_platform_id");
64     String name = jsonObject.getString("name");
65
66     JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
67     Double floor_price_usd = floor_price.getDouble("usd");
68     Double floor_price_native_currency = floor_price.getDouble("native_currency");
69 }

Problems  Javadoc  Declaration  Search  Console X
MultithreadedClient [Java Application] [Javerts@berk:/p2pool/p2poolorg eclipse just; openjdk.hotspot.jre.full.macosx.x86_64_17.0.5.v20221102-0933/re/bin/java (Nov 7, 2023, 4:33:00 PM) [pid: 1523]]
Successfully connected to server atlocalhost/127.0.0.1:9999
Welcome to NFTNet Server
Enter your query:
sbpp -m/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly}
Response from server: Client messaged : sbpp -m/GET -r/NFTServer -v/1.0 -l/{ethlings, squiggly} at : 22
id: ethlings asset_platform_id:polygon-pos name: ethlings floor_price_usd: 83.89 floor_price_native_currency: 0.85 market_cap_usd: 727786.0 market_cap_native_currency: 387.55
id: squiggly asset_platform_id:ethereum name: Squiggly floor_price_usd: 28145.0 floor_price_native_currency: 15.0 market_cap_usd: 2814450.0 market_cap_native_currency: 15.0
Enter your query:
sbpp -m/GET -r/NFTServer -v/1.0 -l/{voxlgllyph}
Response from server: Client messaged : sbpp -m/GET -r/NFTServer -v/1.0 -l/{voxlgllyph} at : 22
id: voxlgllyph asset_platform_id:ethereum name: Voxlgllyph floor_price_usd: 9169.32 floor_price_native_currency: 4.89 market_cap_usd: 1684632.0 market_cap_native_
Enter your query:
sbpp -m/GET -r/NFTServer -v/1.0 -l/{
Response from server: Client messaged : sbpp -m/GET -r/NFTServer -v/1.0 -l/{
Something unexpected happened. Please check your NFT ids
Enter your query:
```

8: If for 30 seconds no query is provided by client the server will close the Socket and prints “Read Timeout”. After this point even if in the client terminal there is no error, when client tries to give an input it gives an error.

Client Before Giving Input



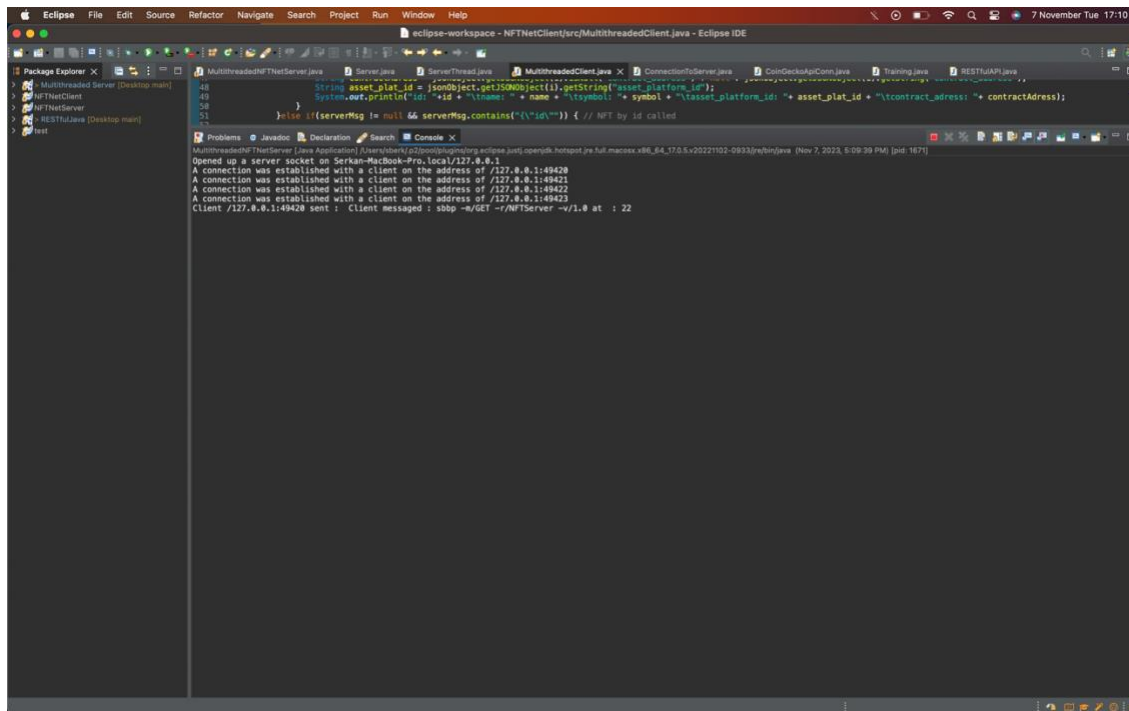
The screenshot shows the Eclipse IDE with the `MultithreadedClient.java` file open. The code is a Java application that connects to an NFTNetServer. It includes a `main` method that creates a `Scanner` to read user input and a `connectionToServer` object. The code also includes a `try` block that handles the connection and a `catch` block that handles exceptions. The console output shows the following messages:

```
Successfully connected to server at localhost/127.0.0.1:9999
Welcome to NFTNet Server
Enter your query:
```

9: If client waits more than 10 seconds without providing any requests Socket will be closed for that specific client.

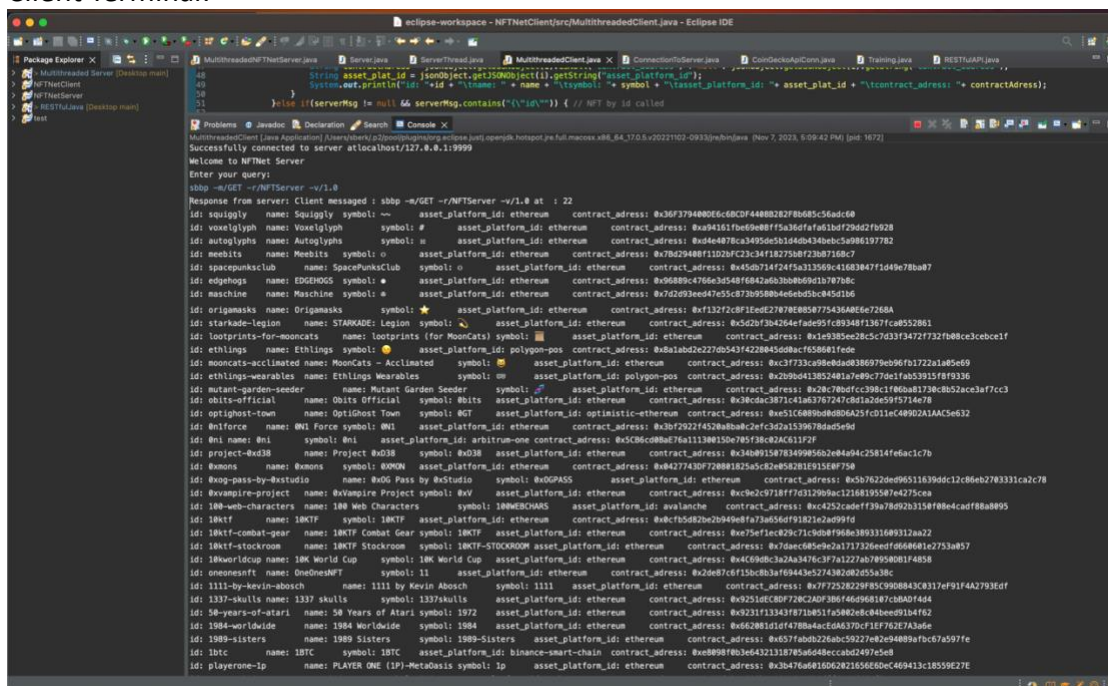
Server:

Server Terminal:



```
Opened up a server socket on Serkan-MacBook-Pro.local/127.0.0.1:40420
A connection was established with a client on the address of /127.0.0.1:40420
A connection was established with a client on the address of /127.0.0.1:40421
A connection was established with a client on the address of /127.0.0.1:40422
A connection was established with a client on the address of /127.0.0.1:40423
Client /127.0.0.1:40420 sent : Client message : sbob -v/GET -r/NFTServer -v/1.0 at : 22
```

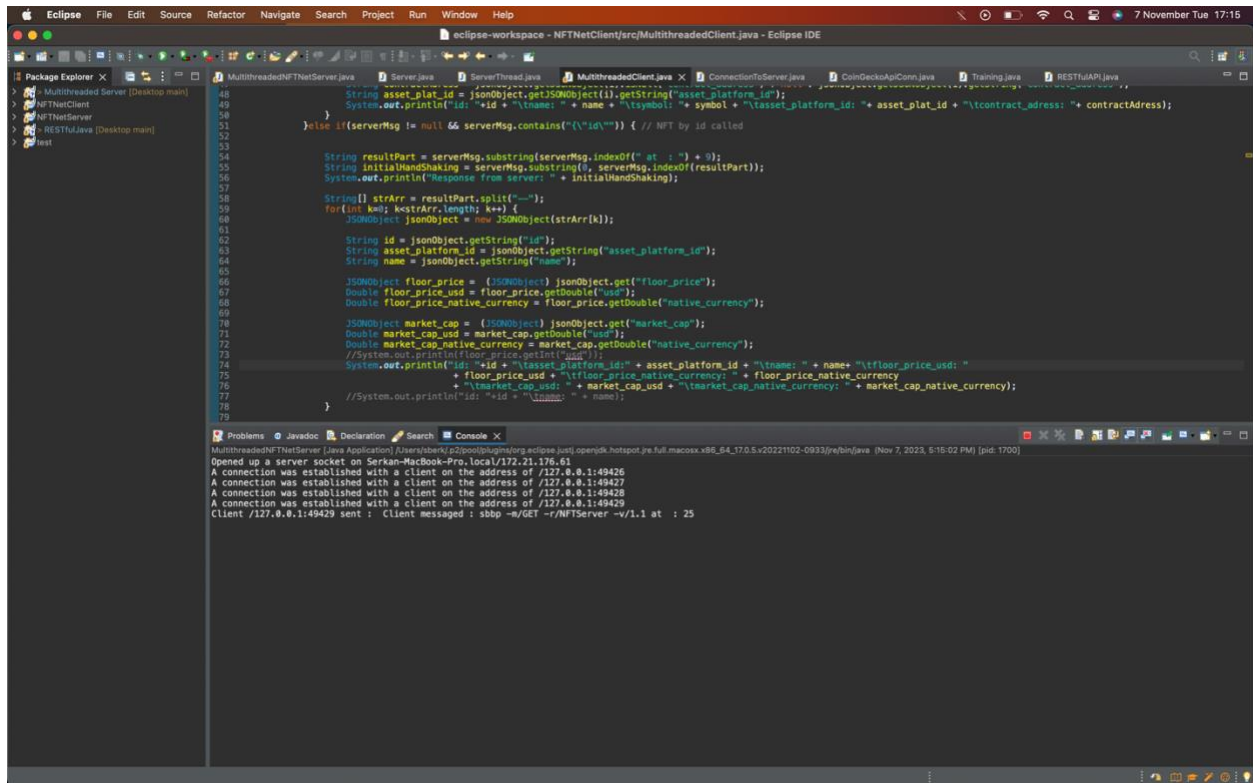
Client Terminal:



```
Successfully connected to server at localhost/127.0.0.1:19999
Welcome to NFTNet Server
Enter your query:
sbob -v/GET -r/NFTServer -v/1.0
Response from server: Client message : sbob -v/GET -r/NFTServer -v/1.0 at : 22
id: squiggly name: Squiggly symbol: ~ asset_platform_id: ethereum contract_address: 0x36f3794800e6c60c2f4880282f8b05c56adc68
id: voxelglyph name: Voxelglyph symbol: # asset_platform_id: ethereum contract_address: 0xa94361f9e6e08f5a36dfafab3bf79d6d7b928
id: autoglyphs name: Autoglyphs symbol: u asset_platform_id: ethereum contract_address: 0x4e4878ca3495de5b1d4db434bebc5a986197782
id: meebits name: Meebits symbol: o asset_platform_id: ethereum contract_address: 0x78629488f11028f2c34f182758b723b07168c7
id: spacepunksclub name: SpacePunksClub symbol: o asset_platform_id: ethereum contract_address: 0x45db7147245a3135569c416b384771049e78ba87
id: edgelyghs name: EDGELYGHs symbol: # asset_platform_id: polygon contract_address: 0x6d89c4766c3e348f68a0230a086b1b17070a
id: machine name: Machine symbol: # asset_platform_id: ethereum contract_address: 0x7d2093e0d4765c8730958084e0eb5b845136
id: origamasks name: Origamasks symbol: ★ asset_platform_id: ethereum contract_address: 0xf1372fd8f1edc27878e858775436a8e672684
id: stakade-legion name: STAKADE: Legion symbol: 🦋 asset_platform_id: ethereum contract_address: 0x5d2bf3b4264efade95fcb9340f1367fca8552861
id: lootprints-for-mooncats name: Lootprints (for MoonCats) symbol: 🐾 asset_platform_id: ethereum contract_address: 0x1e9385e028c3c763f374727327b08ce3cebc1f
id: ethlings name: Ethlings symbol: 🍌 asset_platform_id: polygon-pos contract_address: 0x8abab2c227d0543f4228045d08acf658681fede
id: mooncats-acclimated name: MoonCats - Acclimated symbol: 🌕 asset_platform_id: ethereum contract_address: 0xc37733c09e0da0838979c0b6f917221a1f0cd9
id: ethlings-wearables name: Ethlings Wearables symbol: 🧢 asset_platform_id: polygon-pos contract_address: 0x2db0c4c3834037e08c776c1fab53912f8f3336
id: mutant-garden-seeder name: Mutant Garden Seeder symbol: 🌱 asset_platform_id: ethereum contract_address: 0x28c7b0dccc308c1f60ba81736c8b52ace3a7cc3
id: obits-official name: Obits Official symbol: 📄 asset_platform_id: ethereum contract_address: 0x38dca3871c41a63767247d81a2de595714e78
id: optighost-town name: Optighost Town symbol: 🏠 asset_platform_id: optimistic-ethereum contract_address: 0xe51c6889bd80806a25fc011c48902a1AAC5e632
id: 0x1force name: 0x1 Force symbol: 0x1 asset_platform_id: ethereum contract_address: 0x30f2927f4520ab0ab2efc3d2a1539678d0e9d9
id: 0x1 name: 0x1 symbol: 0x1 asset_platform_id: arbitrum-one contract_address: 0x5c06c08b70a1139d150b781f38c03ca1597f
id: project-0x038 name: Project 0x038 symbol: 0x038 asset_platform_id: ethereum contract_address: 0x34089158781499856248494c258147efdc1c7b
id: 0xmons name: 0xmons symbol: 0xMON asset_platform_id: ethereum contract_address: 0x04277430f720881825a5c82e858281e915f0f758
id: 0xog-pass-by-0xstudio name: 0xOG Pass by 0xStudio symbol: 0xOGPASS asset_platform_id: ethereum contract_address: 0x5b7622ed965116390dc12c8eb278331ca2c78
id: 0xvampire-project name: 0xVampire Project symbol: 0xV asset_platform_id: ethereum contract_address: 0xc9e2c9718ff7f312909ac12168195587e4275cea
id: 180-web-characters name: 180 Web Characters symbol: 180WCHARMS asset_platform_id: avalanche contract_address: 0xc325ce0eff39a789263158f98e4cadf8ba8895
id: 18k1f name: 18K1F symbol: 18K1F asset_platform_id: ethereum contract_address: 0x4cf5d820b2b04ebf7a3d5d6f9121ca0a99f6
id: 18k1f-combat-gear name: 18K1F Combat Gear symbol: 18K1F asset_platform_id: ethereum contract_address: 0x75ef1ac29c71c9dbf986e389331689312aa22
id: 18k1f-stockroom name: 18K1F Stockroom symbol: 18K1F-STOCKROOM asset_platform_id: ethereum contract_address: 0x7daec608e26a171726e0df66681e2753a857
id: 18kworldcup name: 18K World Cup symbol: 18K World Cup asset_platform_id: ethereum contract_address: 0x4c9db8c3a2a3476c3f7a12726e0df66681e2753a857
id: 0xoneofnft name: 0xOneofNFT symbol: 11 asset_platform_id: ethereum contract_address: 0x20e87c6f150b3a769443e57438208253a3bc
id: 1111-by-kevin-abosh name: 1111 by Kevin Abosh symbol: 1111 asset_platform_id: ethereum contract_address: 0x7752322f9f3c090843c017ef91f442793e0f
id: 1337-shells name: 1337 shells symbol: 1337shells asset_platform_id: ethereum contract_address: 0x5251deCB0728C2A030f464968187c080f444
id: 58-years-of-otari name: 58 Years of Atari symbol: 1972 asset_platform_id: ethereum contract_address: 0x3231f1343f871b451f30802e8c840eb91b4762
id: 1984-worldwide name: 1984 Worldwide symbol: 1984 asset_platform_id: ethereum contract_address: 0x6d2081d1d478ba4cd6a637dc1f762e73ade
id: 1989-sisters name: 1989 Sisters symbol: 1989-Sisters asset_platform_id: ethereum contract_address: 0x657fadb226ac59227e2e94889a3c67a597fe
id: 18tc name: 18TC symbol: 18TC asset_platform_id: binance-smart-chain contract_address: 0xe809f703e6432131875a6d48ccab2497e5e8
id: playerone-1p name: PLAYER ONE (1P)-MetaOasis symbol: 1p asset_platform_id: ethereum contract_address: 0x30476a61d062821856E0dc40941318559E27E
```

2: For inappropriate request Server sends a message to the Client. In the below example you can see that client sends request with -v/1.1 but only available version is 1.0. So as can be seen from the screenshots the client terminal prints an error.

Server Terminal:



The screenshot shows the Eclipse IDE with the file `multithreadedClient.java` open. The code is a Java class that implements a RESTful API for an NFT server. It includes methods for `getAssetPlatformById`, `getAssetPlatformBySymbol`, `getAssetPlatformByContractAddress`, `getAssetPlatformByFloorPrice`, `getAssetPlatformByMarketCap`, and `getAssetPlatformByNativeCurrency`. The code uses `JSONObjects` to parse and return data. The console output shows the following messages:

```
Opened up a server socket on Serkan-MacBook-Pro, local/172.21.176.61
A connection was established with a client on the address of /127.0.0.1:49426
A connection was established with a client on the address of /127.0.0.1:49427
A connection was established with a client on the address of /127.0.0.1:49428
A connection was established with a client on the address of /127.0.0.1:49429
Client /127.0.0.1:49429 sent : Client messaged : sbop -r/GET -r/NFTServer -v/1.1 at : 25
```

Client Terminal:


```

eclipse-workspace - NFTNetClient/src/MultithreadedClient.java - Eclipse IDE

Package Explorer X
> Multithreaded Server [Desktop main]
> NFTNetClient
> NFTNetServer
> RESTfulJava [Desktop main]
> test

MultithreadedNFTServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java

48 String asset_plat_id = jsonObject.getString("asset_platform_id");
49 System.out.println("id: " + id + "\tname: " + name + "\tsymbol: " + symbol + "\tasset_platform_id: " + asset_plat_id + "\tcontract_adress: " + contractAddress);
50 }
51 }else if(serverMsg != null && serverMsg.contains("\tid=")) { // NFT by id called
52
53
54 String resultPart = serverMsg.substring(serverMsg.indexOf(" at : ") + 9);
55 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
56 System.out.println("Response from server: " + initialHandShaking);
57
58 String[] strArr = resultPart.split(" ");
59 for(int k=0; k<strArr.length; k++) {
60     JSONObject jsonObject = new JSONObject(strArr[k]);
61
62     String id = jsonObject.getString("id");
63     String asset_platform_id = jsonObject.getString("asset_platform_id");
64     String name = jsonObject.getString("name");
65
66     JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
67     Double floor_price_usd = floor_price.getDouble("usd");
68     Double floor_price_native_currency = floor_price.getDouble("native_currency");
69
70     JSONObject market_cap = (JSONObject) jsonObject.get("market_cap");
71     Double market_cap_usd = market_cap.getDouble("usd");
72     Double market_cap_native_currency = market_cap.getDouble("native_currency");
73     //System.out.println(floor_price.get("usd"));
74     System.out.println("id: " + id + "\tasset_platform_id: " + asset_platform_id + "\tname: " + name + "\tfloor_price_usd: "
75         + floor_price_usd + "\tfloor_price_native_currency: " + floor_price_native_currency
76         + "\tmarket_cap_usd: " + market_cap_usd + "\tmarket_cap_native_currency: " + market_cap_native_currency);
77     //System.out.println("id: " + id + "\tname: " + name);
78 }
79 }

Problems Javadoc Declaration Search Console X
MultithreadedClient (Java Application) J:\Users\shern\p2\workspace\org.eclipse.justi.openjdk.hotspot.jre.full\macosx.x86_64_17.0.5.v20221102-0933\rebin\java (Nov 7, 2023, 5:16:11 PM) [pid: 1704]
Successfully connected to server at local host/127.0.0.1:9999
Welcome to NFTNet Server
Enter your query:
abbb ->/GET -r/NFTServer -v/1.1
Response from server: Client messaged: abbb ->/GET -r/NFTServer -v/1.1 at : 25
=> bad Request 683! Problem with the version input
Enter your query:

```

3: If client waits more than 10 seconds without providing any requests Socket will be closed for that specific client.

Server:

```

Package Explorer X
> Multithreaded Server [Desktop main]
> NFTNetClient
> NFTNetServer
> RESTfulJava [Desktop main]
> test

MultithreadedNFTServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java

68 Double floor_price_usd = floor_price.getDouble("usd");
69 Double floor_price_native_currency = floor_price.getDouble("native_currency");
70
71 JSONObject market_cap = (JSONObject) jsonObject.get("market_cap");
72 Double market_cap_usd = market_cap.getDouble("usd");
73 Double market_cap_native_currency = market_cap.getDouble("native_currency");
74 //System.out.println(floor_price.get("usd"));
75 System.out.println("id: " + id + "\tasset_platform_id: " + asset_platform_id + "\tname: " + name + "\tfloor_price_usd: "
76     + floor_price_usd + "\tfloor_price_native_currency: " + floor_price_native_currency
77     + "\tmarket_cap_usd: " + market_cap_usd + "\tmarket_cap_native_currency: " + market_cap_native_currency);
78 //System.out.println("id: " + id + "\tname: " + name);
79 }
80
81 }else if(serverMsg != null && serverMsg.contains("68")) { // Error code returned from NFTNetServer
82
83 String resultPart = serverMsg.substring(serverMsg.indexOf(" at : ") + 9);
84 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
85 System.out.println("Response from server: " + initialHandShaking + "\n" + resultPart);
86
87 }else{
88     System.err.println("Something unexpected happened. Server might closed Socket because of timeout. Closing the Connection...");
89     //connectionToServer.Disconnect();
90     scanner.close();
91     break;
92 }
93
94 System.out.println("Enter your query: ");
95 message = scanner.nextLine();
96 }
97 connectionToServer.Disconnect();
98 }
99 }
100 }

Problems Javadoc Declaration Search Console X
MultithreadedNFTServer (Java Application) J:\Users\shern\p2\workspace\org.eclipse.justi.openjdk.hotspot.jre.full\macosx.x86_64_17.0.5.v20221102-0933\rebin\java (Nov 7, 2023, 11:30:29 PM) [pid: 2442]
Opened up a server socket on Serkan-MacBook-Pro, local/127.0.0.1
A connection was established with a client on the address of /127.0.0.1:49585
Server has been Run. Closed! (Exception: Thread-0 terminated due to timeout)
Read timed out
Closing the connection
Socket Input Stream Closed
Socket Out Closed
Socket Closed

```

Client:

The image shows a screenshot of an IDE, likely IntelliJ IDEA, with a project named "NFTNetServer". The main editor window displays the source code for "MultithreadedNFTNetServer.java". The code is a Java class that implements a multi-threaded server. It has a main method that starts a Scanner to listen for client connections. When a connection is established, it creates a new thread to handle the client's request. The request is a JSON object containing market cap, floor price, and native currency. The server responds with a JSON object containing the same information. The code is written in Java 8, using lambda expressions and the Stream API. The console window at the bottom shows the output of the program, indicating that the server is running and has successfully connected to a client at localhost/127.0.0.1:9999. The console also shows the client's query and the server's response, as well as the connection being closed.

Screenshots from Codes:

The screenshot shows an IDE with the following components:

- Package Explorer (Left):** Shows a project named 'Multithreaded Server (Desktop main)' with files 'NetClient', 'NetServer', 'RESTfulJava (Desktop main)', and 'test'.
- Main Editor:** Displays the file 'MultithreadedNetServer.java' with the following code:


```

1 package sblgic19_hw2_server;
2
3 public class MultithreadedNetServer {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         Server server = new Server(Server.DEFAULT_SERVER_PORT);
9
10    }
11
12 }
```
- Console (Bottom):** Shows the output of the 'MultithreadedClient' application, including the URL 'http://localhost:8080', the IP address '192.168.1.100', and the port '8080'. It also shows the output of the 'Agora Early Users' application, displaying the 'asset_platform_id' and 'contract_address' for the 'Agora Early Users' and 'Aerial Explorers Launcher'.
- Status Bar (Bottom):** Shows the file is 'Writable', the cursor is at 'Smart Insert', and the line number is '10:1: 216'.

```
Package Explorer X  MultithreadedServer (Desktop main)  MultithreadedServer.java  Server.java X  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java  test

package sbilic19_hw2_server;
import java.io.IOException;

public class Server {

    private ServerSocket serverSocket;

    public static final int DEFAULT_SERVER_PORT = 9999;

    /**
     * Initializes a server socket on the input port, listens to the line, on receiving an incoming
     * connection creates and starts a ServerThread on the client
     * @param port
     */
    public Server(int port)
    {
        try
        {
            serverSocket = new ServerSocket(port);
            System.out.println("Opened up a server socket on " + InetAddress.getLocalHost());
        }
        catch (IOException e)
        {
            e.printStackTrace();
            System.err.println("Server class. Constructor exception on opening a server socket");
        }
        while (true)
        {
            listenAndAccept();
        }
    }

    /**
     * Listens to the line and starts a connection on receiving a request from the client
     * The connection is started and initiated as a ServerThread object
     */
    private void listenAndAccept()
    {
        Socket s;
        try
        {
            s = serverSocket.accept();
            System.out.println("A connection was established with a client on the address of " + s.getRemoteSocketAddress());
            ServerThread st = new ServerThread(s);
            st.start();
        }
        catch (Exception e)
        {
            e.printStackTrace();
            System.err.println("Server Class. Connection establishment error inside listen and accept function");
        }
    }
}
```

```
1 package sblogic19_hw2_server;
2
3 import java.io.BufferedReader;
4
5 public class ServerThread extends Thread {
6
7     protected BufferedReader is;
8     protected PrintWriter os;
9     protected Socket s;
10     private String line = new String();
11     private String lines = new String();
12
13     public ServerThread(Socket s)
14     {
15         this.s = s;
16     }
17
18     //s
19     // The server thread, echos the client until it receives the QUIT string from the client
20     //s
21     public void run()
22     {
23         try
24         {
25             is = new BufferedReader(new InputStreamReader(s.getInputStream()));
26             os = new PrintWriter(s.getOutputStream());
27
28             s.setSoTimeout(10000);
29             line = is.readLine();
30             s.setSoTimeout(10000);
31             while (line.compareTo("QUIT") != 0)
32             {
33                 lines = "Client messaged : " + line + " at : " + Thread.currentThread().getId();
34
35                 //os.println(lines);
36                 //os.flush();
37                 System.out.println("Client " + s.getRemoteSocketAddress() + " sent : " + lines);
38                 //lines += "\n";
39
40                 CoinGeckoApiConn apiConn = new CoinGeckoApiConn();
41
42                 String lowercase_line = line.toLowerCase();
43                 String[] arr = lowercase_line.split("-");
44                 String idOfMfts = "";
45                 boolean[] violationFlags = {true, true, true, true, true, false}; // (protocol_name, method_type, resource, version, id, general)
46                 boolean idOfmg = false;
47                 if(arr[0].trim().equals("sbbp")) {
48                     violationFlags[0] = false;
49                 }
50
51                 //s
52                 for(int i=1; i<arr.length; i++) {
53                     if(arr[i].substring(2).trim().equals("afternoon")) {
54                         violationFlags[2] = false;
55                     }
56                     else if(arr[i].substring(2).trim().equals("get")) {
57                         violationFlags[1] = false;
58                     }
59                     else if(arr[i].substring(2).trim().equals("1.0")) {
60                         violationFlags[3] = false;
61                     }
62                     else {
63                         //System.out.println(arr[i].substring(2).trim().equals("get"));
64                         violationFlags[5] = true;
65                     }
66                 }
67                 //System.out.println("Violation");
68             }
69         }
70     }
71 }
72
73 //s
74
75 //s
76
```



```
Package Explorer X
> Multithreaded Server (Desktop main)
> NFTNetClient
> NFTNetServer
> RESTfulJava (Desktop main)
> test

MultithreadedNFTServer.java
1 import java.util.Scanner;
2 import java.util.concurrent.Executors;
3 import java.util.concurrent.ScheduledExecutorService;
4 import java.util.concurrent.TimeUnit;
5 import org.json.JSONArray;
6 import org.json.JSONObject;
7
8 public class MultithreadedClient {
9
10     public static void main(String[] args) {
11         // connectionToServer = new ConnectionToServer(ConnectionToServer.DEFAULT_SERVER_ADDRESS, ConnectionToServer.DEFAULT_SERVER_PORT);
12         connectionToServer.connect();
13         ScheduledExecutorService scheduler = Executors.newScheduledThreadPool(1);
14         Runnable task = () -> {
15             if(connectionToServer.checkConnection() == false) {
16                 System.out.println("Closing the connection");
17                 connectionToServer.disconnect();
18             }
19         };
20         // Schedule the task to run every 1 second
21         scheduler.scheduleAtFixedRate(task, 0, 1, TimeUnit.SECONDS);
22         System.out.println("Welcome to NFTNet Server");
23         Scanner scanner = new Scanner(System.in);
24         System.out.println("Enter your query: ");
25         String message = scanner.nextLine();
26         while (!message.equals("QUIT")) {
27
28             String serverMsg = connectionToServer.sendForAnswer(message);
29             if(serverMsg != null && serverMsg.contains("{\"id\":\"\"}")) { // all of the NFTs are called
30                 String resultPart = serverMsg.substring(serverMsg.indexOf("at : ") + 9);
31                 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
32                 int subStringIndex = resultPart.indexOf("[");
33                 System.out.println("Response from server: " + initialHandShaking);
34                 //System.out.println("=> Response: ");
35                 String resultJson = resultPart.substring(subStringIndex);
36                 JSONArray jsonArray = new JSONArray(resultJson);
37                 for (int i = 0; i < jsonArray.length(); i++) {
38                     String id = jsonArray.getJSONObject(i).getString("id");
39                     String name = jsonArray.getJSONObject(i).getString("name");
40                     String symbol = jsonArray.getJSONObject(i).getString("symbol");
41                     String contractAddress = jsonArray.getJSONObject(i).getString("contract_address") ? null : jsonArray.getJSONObject(i).getString("contract_address");
42                     String asset_plat_id = jsonArray.getJSONObject(i).getString("asset_platform_id");
43                     System.out.println("id: " + id + " \t name: " + name + " \t symbol: " + symbol + " \t asset_platform_id: " + asset_plat_id + " \t contract_address: " + contractAddress);
44                 }
45             } else if(serverMsg != null && serverMsg.contains("{\"id\":\"\"}")) { // NFT by id called
46                 String resultPart = serverMsg.substring(serverMsg.indexOf("at : ") + 9);
47                 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
48                 System.out.println("Response from server: " + initialHandShaking);
49                 String[] strArr = resultPart.split("--");
50                 for(int k=0; k<strArr.length; k++) {
51                     JSONObject jsonObject = new JSONObject(strArr[k]);
52
53                     String id = jsonObject.getString("id");
54                     String asset_platform_id = jsonObject.getString("asset_platform_id");
55                     String name = jsonObject.getString("name");
56                     JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
57                     Double floor_price_usd = floor_price.getDouble("usd");
58                     Double floor_price_native_currency = floor_price.getDouble("native_currency");
59                     JSONObject market_cap = (JSONObject) jsonObject.get("market_cap");
60                     Double market_cap_usd = market_cap.getDouble("usd");
61                     Double market_cap_native_currency = market_cap.getDouble("native_currency");
62                     //System.out.println(floor_price.toString());
63                     System.out.println("id: " + id + " \t asset_platform_id: " + asset_platform_id + " \t name: " + name + " \t floor_price_usd: " + floor_price_usd + " \t floor_price_native_currency: " + floor_price_native_currency + " \t market_cap_usd: " + market_cap_usd + " \t market_cap_native_currency: " + market_cap_native_currency);
64                     //System.out.println("id: " + id + " \t name: " + name);
65                 }
66             }
67             else if(serverMsg != null && serverMsg.contains("Error")) { // Error code returned from NFTNetServer
68                 String resultPart = serverMsg.substring(serverMsg.indexOf("at : ") + 9);
69                 String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
70                 System.out.println("Response from server: " + initialHandShaking + "\n" + resultPart);
71             } else {
72                 System.err.println("Something unexpected happened. Server might closed Socket because of timeout. Closing the Connection...");
73                 connectionToServer.disconnect();
74                 scanner.close();
75                 break;
76             }
77             System.out.println("Enter your query: ");
78             message = scanner.nextLine();
79         }
80         connectionToServer.disconnect();
81     }
82 }
```

Problems Javadoc Declaration Search Console X

terminated: MultithreadedClient (Java Application) J:\devs\berk\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_17.0.5.v20221102-0933\jre\bin\java (Nov 7, 2023, 11:30:32 PM - 11:31:02 PM) [pid: 2443]

Successfully connected to server at localhost/127.0.0.1:9999

Welcome to NFTNet Server

Enter your query:

sbop -u/GET -r/NFTServer -v/1.0

Writable Smart Insert 96 : 41 : 5374

```
Package Explorer X
> Multithreaded Server (Desktop main)
> NFTNetClient
> NFTNetServer
> RESTfulJava (Desktop main)
> test

MultithreadedNFTServer.java
40 //System.out.println("=> Response: ");
41 String resultPart = serverMsg.substring(subStringIndex);
42 JSONArray jsonArray = new JSONArray(resultJson);
43 for (int i = 0; i < jsonArray.length(); i++) {
44     String id = jsonArray.getJSONObject(i).getString("id");
45     String name = jsonArray.getJSONObject(i).getString("name");
46     String symbol = jsonArray.getJSONObject(i).getString("symbol");
47     String contractAddress = jsonArray.getJSONObject(i).getString("contract_address") ? null : jsonArray.getJSONObject(i).getString("contract_address");
48     String asset_plat_id = jsonArray.getJSONObject(i).getString("asset_platform_id");
49     System.out.println("id: " + id + " \t name: " + name + " \t symbol: " + symbol + " \t asset_platform_id: " + asset_plat_id + " \t contract_address: " + contractAddress);
50 }
51 } else if(serverMsg != null && serverMsg.contains("{\"id\":\"\"}")) { // NFT by id called
52
53     String resultPart = serverMsg.substring(serverMsg.indexOf("at : ") + 9);
54     String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
55     System.out.println("Response from server: " + initialHandShaking);
56     String[] strArr = resultPart.split("--");
57     for(int k=0; k<strArr.length; k++) {
58         JSONObject jsonObject = new JSONObject(strArr[k]);
59
60         String id = jsonObject.getString("id");
61         String asset_platform_id = jsonObject.getString("asset_platform_id");
62         String name = jsonObject.getString("name");
63         JSONObject floor_price = (JSONObject) jsonObject.get("floor_price");
64         Double floor_price_usd = floor_price.getDouble("usd");
65         Double floor_price_native_currency = floor_price.getDouble("native_currency");
66         JSONObject market_cap = (JSONObject) jsonObject.get("market_cap");
67         Double market_cap_usd = market_cap.getDouble("usd");
68         Double market_cap_native_currency = market_cap.getDouble("native_currency");
69         //System.out.println(floor_price.toString());
70         System.out.println("id: " + id + " \t asset_platform_id: " + asset_platform_id + " \t name: " + name + " \t floor_price_usd: " + floor_price_usd + " \t floor_price_native_currency: " + floor_price_native_currency + " \t market_cap_usd: " + market_cap_usd + " \t market_cap_native_currency: " + market_cap_native_currency);
71         //System.out.println("id: " + id + " \t name: " + name);
72     }
73 }
74 } else if(serverMsg != null && serverMsg.contains("Error")) { // Error code returned from NFTNetServer
75     String resultPart = serverMsg.substring(serverMsg.indexOf("at : ") + 9);
76     String initialHandShaking = serverMsg.substring(0, serverMsg.indexOf(resultPart));
77     System.out.println("Response from server: " + initialHandShaking + "\n" + resultPart);
78 } else {
79     System.err.println("Something unexpected happened. Server might closed Socket because of timeout. Closing the Connection...");
80     connectionToServer.disconnect();
81     scanner.close();
82     break;
83 }
84 System.out.println("Enter your query: ");
85 message = scanner.nextLine();
86 }
87 connectionToServer.disconnect();
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Problems Javadoc Declaration Search Console X

terminated: MultithreadedClient (Java Application) J:\devs\berk\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_17.0.5.v20221102-0933\jre\bin\java (Nov 7, 2023, 11:30:32 PM - 11:31:02 PM) [pid: 2443]

Successfully connected to server at localhost/127.0.0.1:9999

Welcome to NFTNet Server

Enter your query:

sbop -u/GET -r/NFTServer -v/1.0

Writable Smart Insert 96 : 41 : 5374


```
Package Explorer X
> Multithreaded Server (Desktop main)
> NFTNetClient
> NFTNetServer
> RESTfulJava (Desktop main)
> test

MultithreadedNFTNetServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java X  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java

10 import java.io.BufferedReader;
11
12 public class ConnectionToServer {
13
14     public static final String DEFAULT_SERVER_ADDRESS = "localhost";
15     public static final int DEFAULT_SERVER_PORT = 9999;
16     private Socket s;
17     private BufferedReader is;
18     protected BufferedWriter os;
19     protected PrintWriter pw;
20
21     protected String serverAddress;
22     protected int serverPort;
23
24     /**
25      * @param address IP address of the server, if you are running the server on the same computer as client, put the address as "localhost"
26      * @param port port number of the server
27      */
28     public ConnectionToServer(String address, int port) {
29         serverAddress = address;
30         serverPort = port;
31     }
32
33     /**
34      * Establishes a socket connection to the server that is identified by the serverAddress and the serverPort
35      */
36     public void Connect() {
37         try {
38             s = new Socket(serverAddress, serverPort);
39             //is = new BufferedReader(new InputStreamReader(System.in));
40             //os = new BufferedWriter(new OutputStreamWriter(System.out));
41             Read and write buffers on the socket
42             is = new BufferedReader(new InputStreamReader(s.getInputStream()));
43             os = new PrintWriter(s.getOutputStream());
44             System.out.println("Successfully connected to server at " + s.getRemoteSocketAddress());
45         } catch (IOException e) {
46             //e.printStackTrace();
47             System.err.println("Error: no server has been found on " + serverAddress + ":" + serverPort);
48         }
49     }
50
51     /**
52      * sends the message String to the server and returns the answer
53      * @param message input message string to the server
54      * @return the received server answer
55      */
56     public String SendForAnswer(String message) {
57         String response = new String();
58         try {
59             //sends the message to the server via PrintWriter
60             os.println(message);
61             os.flush();
62             //Reads a line from the server via Buffer Reader
63             response = is.readLine();
64         } catch (IOException e) {
65             //e.printStackTrace();
66             System.err.println("Error: no server has been found on " + serverAddress + ":" + serverPort);
67         }
68         return response;
69     }
70
71     /**
72      * Disconnects the socket and closes the buffers
73      */
74     public void Disconnect() {
75         try {
76             is.close();
77             os.close();
78             s.close();
79             System.out.println("ConnectionToServer. SendForAnswer. Connection Closed");
80         } catch (IOException e) {
81             //e.printStackTrace();
82         }
83     }
84 }
```

```
Package Explorer X
> Multithreaded Server (Desktop main)
> NFTNetClient
> NFTNetServer
> RESTfulJava (Desktop main)
> test

MultithreadedNFTNetServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java X  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java

41 Read and write buffers on the socket
42 is = new BufferedReader(new InputStreamReader(s.getInputStream()));
43 os = new PrintWriter(s.getOutputStream());
44 System.out.println("Successfully connected to server at " + s.getRemoteSocketAddress());
45 } catch (IOException e) {
46     //e.printStackTrace();
47     System.err.println("Error: no server has been found on " + serverAddress + ":" + serverPort);
48 }
49
50 /**
51  * sends the message String to the server and returns the answer
52  * @param message input message string to the server
53  * @return the received server answer
54  */
55 public String SendForAnswer(String message) {
56     String response = new String();
57     try {
58         //sends the message to the server via PrintWriter
59         os.println(message);
60         os.flush();
61         //Reads a line from the server via Buffer Reader
62         response = is.readLine();
63     } catch (IOException e) {
64         //e.printStackTrace();
65         System.err.println("Error: no server has been found on " + serverAddress + ":" + serverPort);
66     }
67     return response;
68 }
69
70 /**
71  * Disconnects the socket and closes the buffers
72  */
73 public void Disconnect() {
74     try {
75         is.close();
76         os.close();
77         s.close();
78         System.out.println("ConnectionToServer. SendForAnswer. Connection Closed");
79     } catch (IOException e) {
80         //e.printStackTrace();
81     }
82 }
83 }
```

```
Package Explorer X  MultithreadedNFTServer.java  Server.java  ServerThread.java  MultithreadedClient.java  ConnectionToServer.java  CoinGeckoApiConn.java  Training.java  RESTfulAPI.java
> Multithreaded Server [Desktop main]
> NFTNetClient
> NFTNetServer
> RESTfulJava [Desktop main]
> test

1 package sbilgic19_m2_server;
2
3 import java.io.IOException;
4
5 public class CoinGeckoApiConn {
6
7     public CoinGeckoApiConn() {
8     }
9
10    public String getAllNFTs() throws MalformedURLException, IOException {
11        URL url = new URL("https://api.coingecko.com/api/v3/nfts/list");
12
13        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
14        conn.setRequestMethod("GET");
15        conn.connect();
16        String inline = "";
17        Scanner scanner = new Scanner(url.openStream());
18        while (scanner.hasNext()) {
19            inline += scanner.nextLine();
20        }
21        //Close the scanner
22        scanner.close();
23        return inline;
24    }
25
26    public String getNFTsById(String ids) throws IOException {
27        ids = ids.substring(1, ids.length()-1); // getting rid of () parenthesis
28        String[] idArr = ids.split(",");
29        String inline = "";
30        try {
31            for(int i=0; i<idArr.length; i++) {
32                URL url = new URL("https://api.coingecko.com/api/v3/nfts/"+idArr[i].trim());
33
34                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
35                conn.setRequestMethod("GET");
36                conn.connect();
37                Scanner scanner = new Scanner(url.openStream());
38                while (scanner.hasNext()) {
39                    inline += scanner.nextLine();
40                }
41                //Close the scanner
42                scanner.close();
43            }
44            return inline;
45        } catch (Exception ex) {
46            return "Something unexpected happened. Please check your NFT ids";
47        }
48    }
49
50 }
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
```