PDMG SSP

Programmatic Digital Gateway Supply-Side Platform

PDMG_SSP

Documentation

Version 1.1

08/21/2024

Table of Contents

1. Introduction

PDMG_SSP SUMMARY.

PDMG_SSP (Programmatic Digital Media Gateway – Supply Side Platform) is a comprehensive Supply Side Platform designed to manage and optimize ad inventory for digital publishers. It supports the OpenRTB (Real_Time Bidding) protocol, enabling publishers to connect with multiple demand channels (DSPs – Demand Side Platform), and maximize their ad revenue through programmatic advertising. The SSP handles bid requests, processes bids, selects the winning bid based on predetermined criteria, records the results for future daily and monthly reporting and optimization.

This documentation provides comprehensive technical documentation for the restify-based Supply-Side Platform project and provides guidelines and instructions for running and maintaining tests including unit tests, tests using postman file and high traffic testing using Apache Benche for the 'pdmg_ssp' project. This project uses Mocha as the test framework, Chai for assertions, and Sinon for mocking and stubbing.

Key Features

- **Auction Handling:** The cord functionality of the SSP is to manage real-time bidding (RTB) auction based on OpenRTB 2.6 protocol. The platform efficiently handles auction requests, processes bid responses, and determines the winning bid based on the highest or second highest bid price.
- **Private Marketplace (PMP):** The SSP supports Private Marketplace deals, enabling publishers to offer premium inventory to select buyers at negotiated rates. These deals ensure that the inventory is only accessible to specific DSPs, maintaining exclusivity and allowing for higher CPMs (Cost Per Mile).
- **Extensible API Endpoints**: The platform offers well-structured API endpoints for handling different aspects of the auction process, including managing publishers, devices, endpoints, deals, reporting and initiating auctions.
- **Daily and Monthly Reporting**: The SSP provides daily and monthly reporting to publishers on auction performance, including metrics such as total bids, total wins, win prices, and more along with csv file. This information is critical for publishers to optimize their inventory and maximize revenue.
- **Error Handling and Logging**: Integrated logging using `morgain` and `winston` ensures that all server activities are logged, and any errors encountered are properly handled and reported.
- **Scalable Architecture**: The SSP is designed with scalability in mind, allowing it to handle a large number of concurrent auction requests with minimal latency.

2. Structure

Project structure is as follows:

```
pdmg_ssp/
├── config/
├── controllers/
├── crons/
├── engine/
├── libs/
├── middlewares/
├── model/
├── openrtb/
├── reports/
├── services/
├── tempates/
├── test/
│   ├── apis/
│   │   └──deal.router.test.js
│   │   └──device.router.test.js
│   │   └──endpoint.router.test.js
│   │   └──publisher.router.test.js
│   ├── controllers/
│   ├── middlewares/
│   │   └──validate.middleware.test.js
│   ├── libs/
│   │   └──methods.test.js
│   ├── openrtb/
│   │   └──bid_request.test.js
│   ├── db.test.js
│   ├── server.test.js
├── validators/
├── index.js
└── server.js
```

3. Setup Environments

3.1 Install Node.js and npm:

Ensure Node.js and npm are installed on your system. You can download them from nodejs.org. Node version *16.15.1* required.

3.2 Move to the project directory

```
cd pdmg_ssp
```

3.3 Install Dependecies

```
npm install
```

3.4 Set up MySQL Database:

Ensure you have MySQL installed and running. Create a MySQL database and update the ‘config’ file with your database credentials.

3.4.1 Example MySQL Setup

```
sudo apt-get update
sudo apt-get install mysql-server
sudo mysql_secure_installation
mysql -u root -p
CREATE DATABASE pdmg_ssp;
```

4. Best Practices

4.1 Code Quality:

- **Modular Design**: Structure your code into modular components, each responsible for a specific functionality. This enhances code readability and maintainability.
- **Error Handling**: Implement consistent error handling across the application using middleware. Ensure that all errors are logged and that appropriate responses are sent to clients.
- **Documentation**: Write clear and concise comments in the code, and maintain up-to-date API documentation.

4.2 Performance Optimization:

- **Database Optimization**: Use indexing and query optimization techniques to enhance database performance. Avoid unnecessary joins and complex queries.
- **Caching**: Implement caching for frequently accessed data to reduce database load and improve response times.
- **Asynchronous Processing**: Use asynchronous programming to handle I/O-bound operations, such as database queries and API calls, to prevent blocking the event loop.

4.3 Security:

- **Input Validation**: Validate all incoming requests to prevent SQL injection, XSS, and other common attacks.
- **Authentication and Authorization**: Implement strong authentication and authorization mechanisms, especially for sensitive endpoints by using Admin access key and publisher security key.

4.4 Scalability:

- **Horizontal Scaling**: Design the application to scale horizontally by adding more instances to handle increased load.
- **Stateless Services**: Keep services stateless where possible, allowing for easier scaling and load balancing.
- **Microservices Architecture**: Consider breaking down the SSP into microservices for better scalability and maintainability.

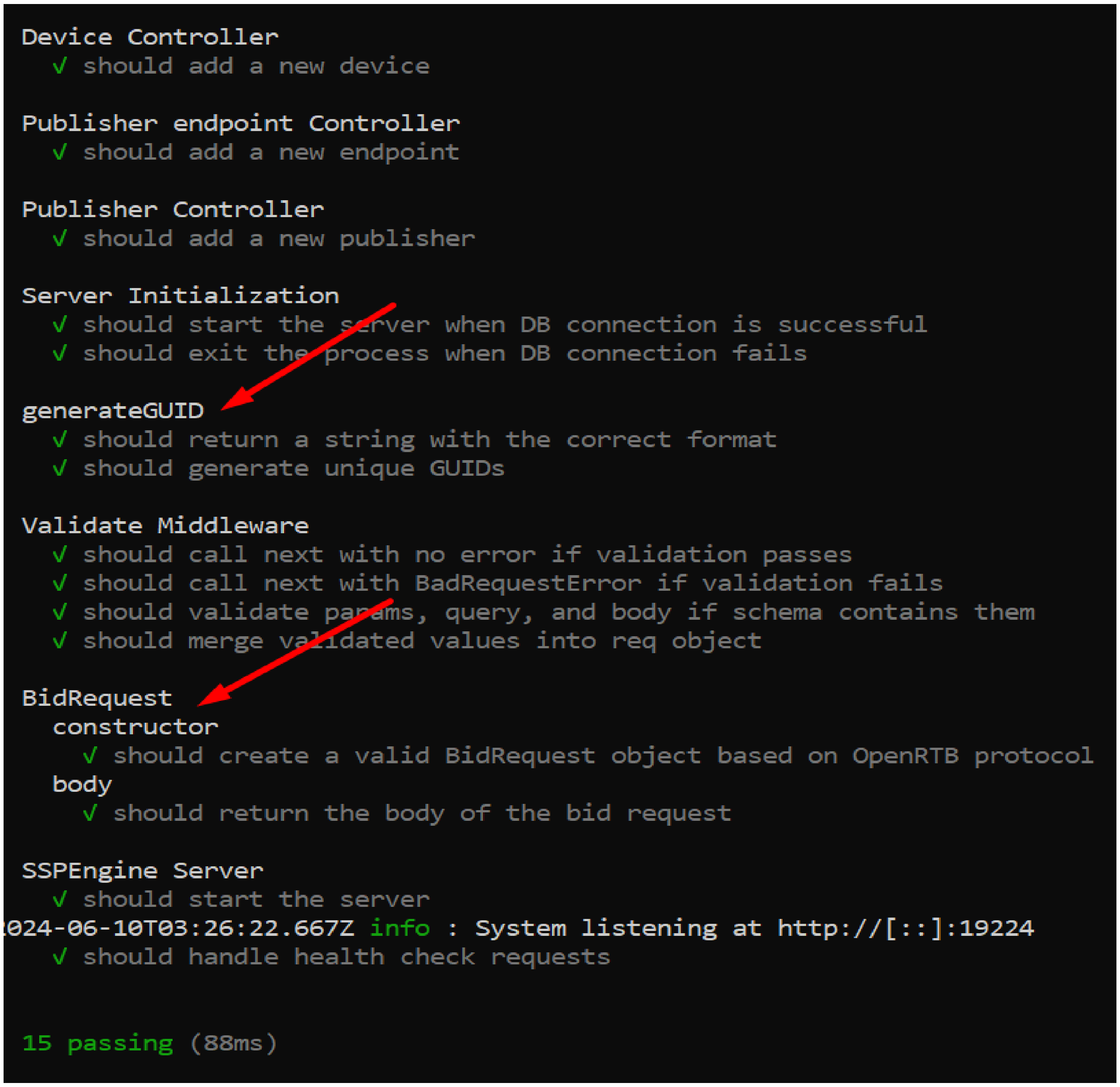
5. Running Tests

5.1 Run all tests

- **npm test**

This tests all test files in test folder.

- **Expected result**



```
Device Controller
  √ should add a new device

Publisher endpoint Controller
  √ should add a new endpoint

Publisher Controller
  √ should add a new publisher

Server Initialization
  √ should start the server when DB connection is successful
  √ should exit the process when DB connection fails

generateGUID
  √ should return a string with the correct format
  √ should generate unique GUIDs

Validate Middleware
  √ should call next with no error if validation passes
  √ should call next with BadRequestError if validation fails
  √ should validate params, query, and body if schema contains them
  √ should merge validated values into req object

BidRequest
  constructor
    √ should create a valid BidRequest object based on OpenRTB protocol
  body
    √ should return the body of the bid request

SSPEngine Server
  √ should start the server
024-06-10T03:26:22.667Z info : System listening at http://[::]:19224
  √ should handle health check requests


15 passing (88ms)
```

5.2 Use npx mocha

You can test just a file or a folder you want to test using `npx mocha` command.

5.2.1 Test database connection

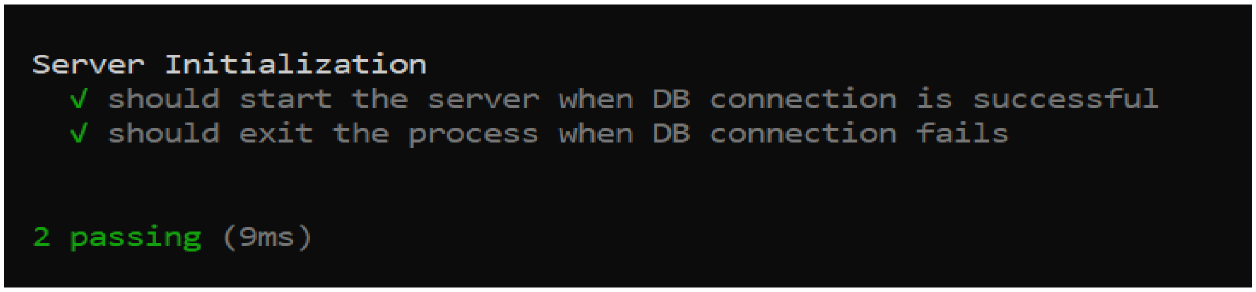
- **npx mocha test/db.test.js**

This tests whether your system is connected with MySQL database with your credentials or not.

db.test.js

```
 ... ... ...

  it('should start the server when DB connection is successful', (done) => {
    connectStub.callsFake((callback) => callback(null));

    DB_HANDLE.connect(err => {
      assert.strictEqual(err, null)
      done()
    })
  });

  it('should exit the process when DB connection fails', (done) => {
    const error = new Error('DB connection failed');
    connectStub.callsFake((callback) => callback(error));

    DB_HANDLE.connect(err => {
      assert.strictEqual(err, error)
      done()
    })
  });
});
```

- **Expected result**



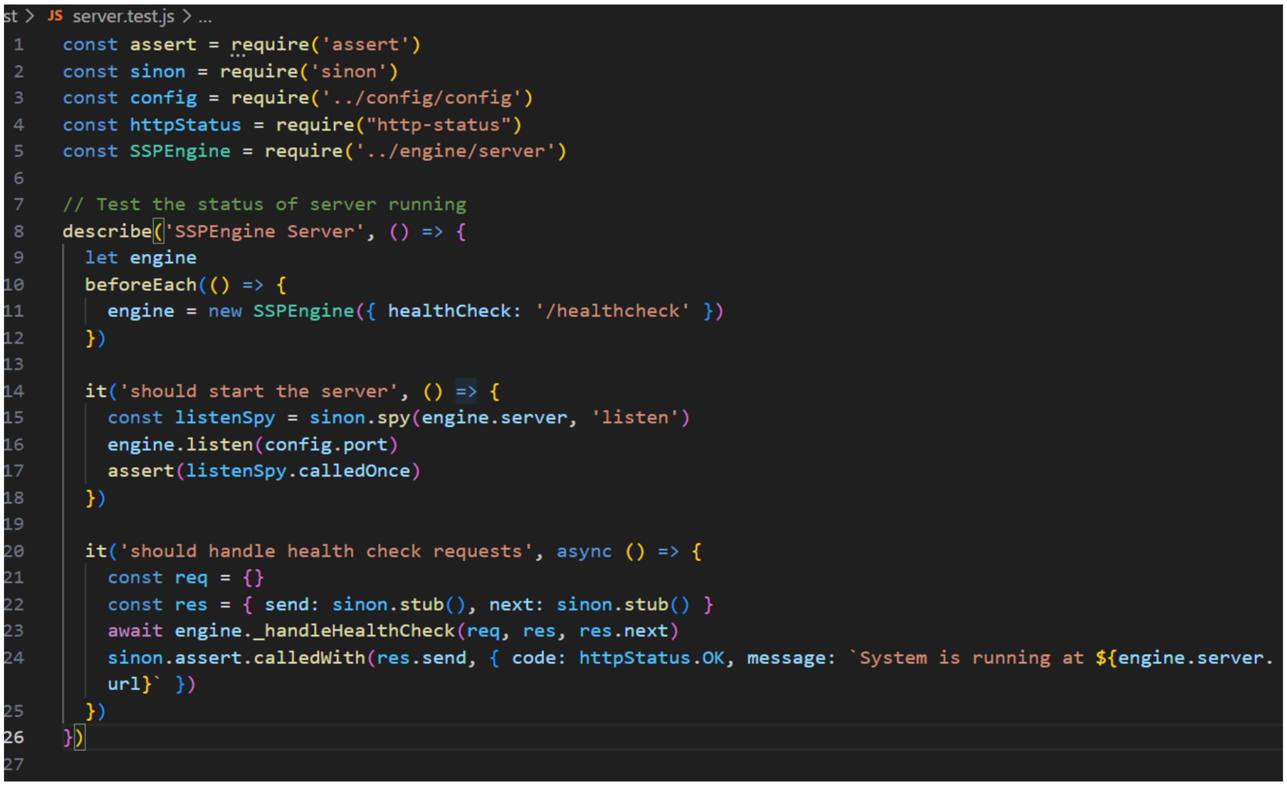
```
Server Initialization
  √ should start the server when DB connection is successful
  √ should exit the process when DB connection fails


2 passing (9ms)
```

5.2.2 Test server functionality

- **npx mocha test/server.test.js**

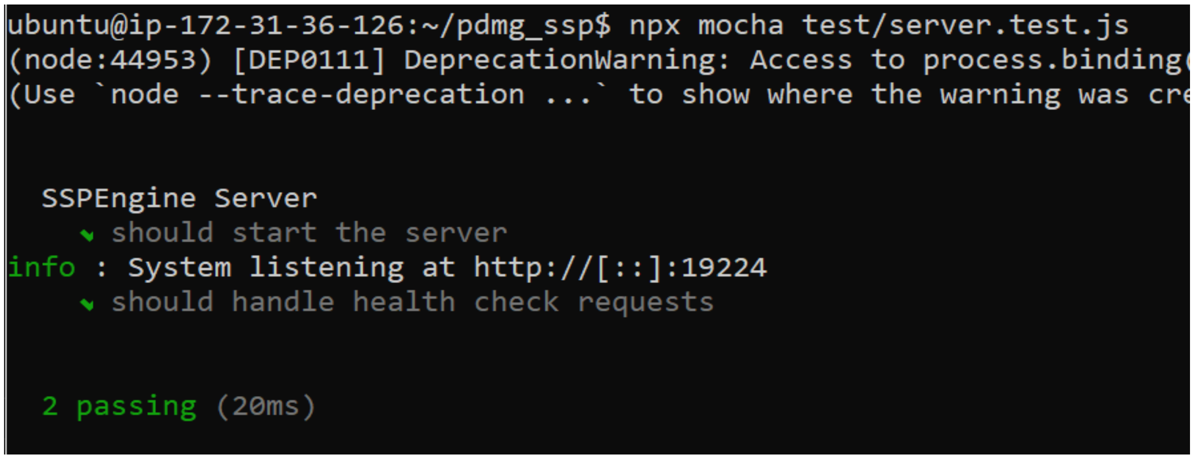
This tests whether your system is running correctly

server.test.js



```
st > JS server.test.js > ...
const assert = require('assert')
const sinon = require('sinon')
const config = require('../config/config')
const httpStatus = require("http-status")
const SSPEngine = require('../engine/server')

// Test the status of server running
describe('SSPEngine Server', () => {
  let engine
  beforeEach(() => {
    engine = new SSPEngine({ healthCheck: '/healthcheck' })
  })

  it('should start the server', () => {
    const listenSpy = sinon.spy(engine.server, 'listen')
    engine.listen(config.port)
    assert(listenSpy.calledOnce)
  })

  it('should handle health check requests', async () => {
    const req = {}
    const res = { send: sinon.stub(), next: sinon.stub() }
    await engine._handleHealthCheck(req, res, res.next)
    sinon.assert.calledWith(res.send, { code: httpStatus.OK, message: `System is running at ${engine.server.
    url}` })
  })
})
```

- **Expected result**



```
ubuntu@ip-172-31-36-126:~/pdmg_ssp$ npx mocha test/server.test.js
(node:44953) [DEP0111] DeprecationWarning: Access to process.binding(
(Use `node --trace-deprecation ...` to show where the warning was cre


  SSPEngine Server
    ✓ should start the server
info : System listening at http://[::]:19224
    ✓ should handle health check requests


  2 passing (20ms)
```

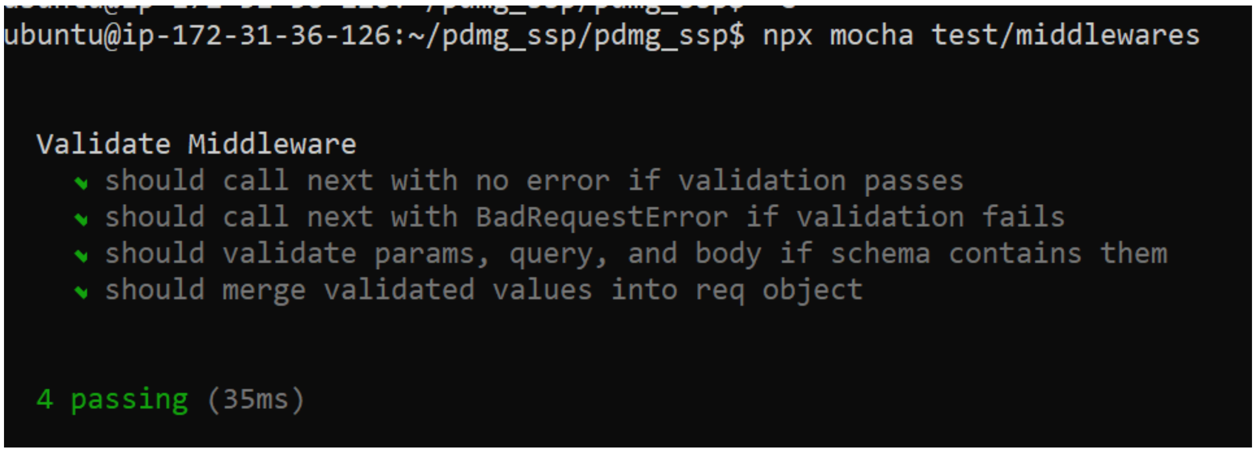
5.2.3 Test middlewares

- **npx mocha test/middlewares**

This tests all test files in middlewares' folder.

(you can test each file: npx mocha test/middlewares/file_name e.g. validate.middleware.test.js)

Expected results



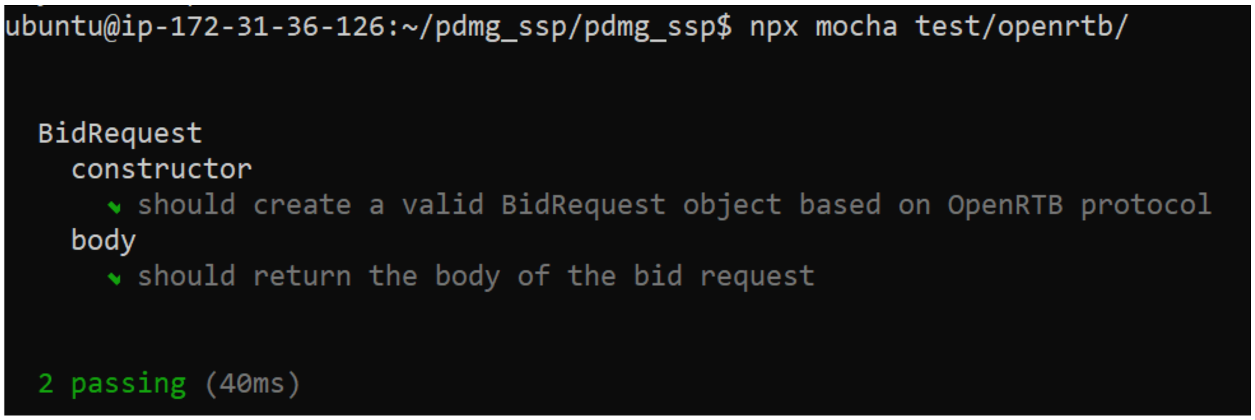
```
ubuntu@ip-172-31-36-126:~/pdmg_ssp/pdmg_ssp$ npx mocha test/middlewares


  Validate Middleware
    ✓ should call next with no error if validation passes
    ✓ should call next with BadRequestError if validation fails
    ✓ should validate params, query, and body if schema contains them
    ✓ should merge validated values into req object


  4 passing (35ms)
```

5.2.4 Test openrtb support

- **npx mocha test/openrtb/**

This tests whether this system generates correct bid request based on OpenRTB support.

Expected results



```
ubuntu@ip-172-31-36-126:~/pdmg_ssp/pdmg_ssp$ npx mocha test/openrtb/


  BidRequest
    constructor
      ✓ should create a valid BidRequest object based on OpenRTB protocol
    body
      ✓ should return the body of the bid request


  2 passing (40ms)
```

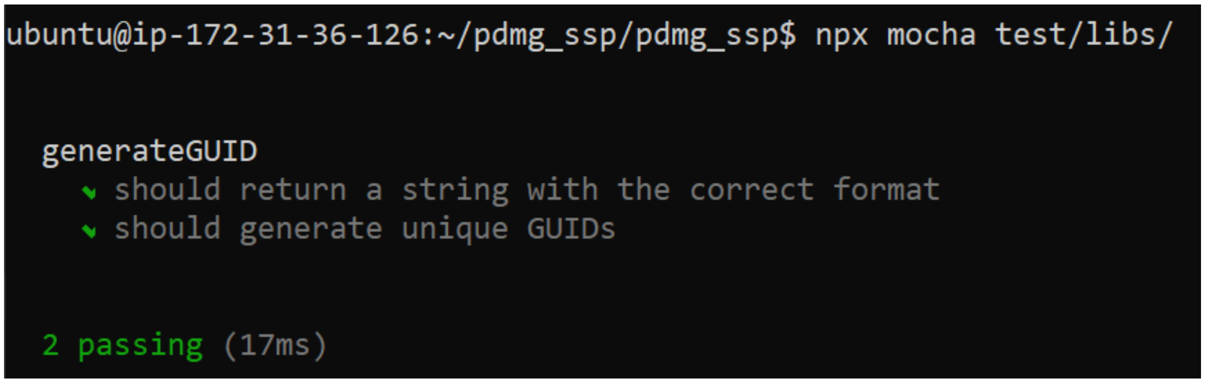
5.2.5 Test libs

- **npx mocha test/libs/**

This tests all files in libs folder (which is used as library in this system).

Expected results

generateGUID function

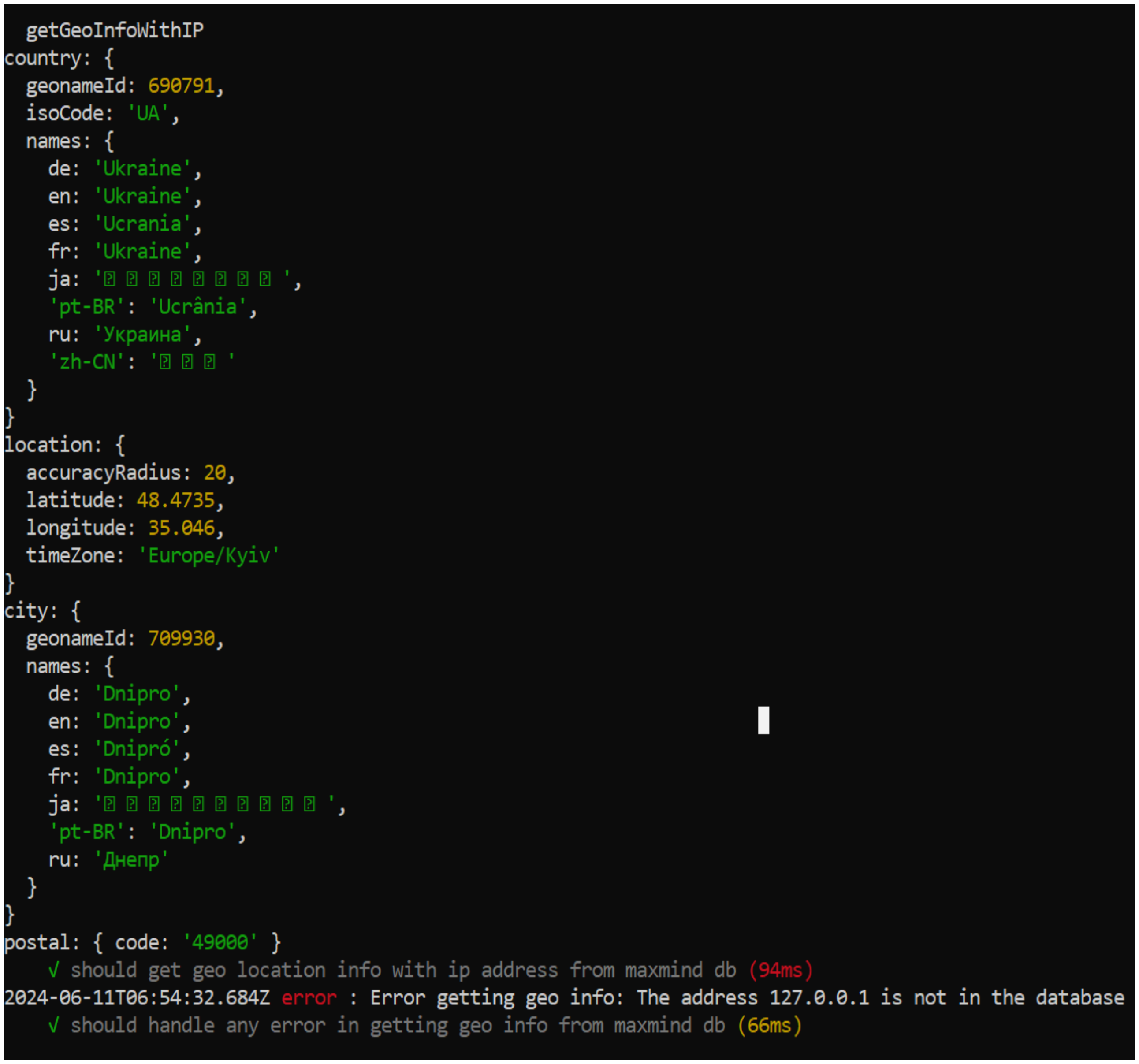


```
ubuntu@ip-172-31-36-126:~/pdmg_ssp/pdmg_ssp$ npx mocha test/libs/


  generateGUID
    ✓ should return a string with the correct format
    ✓ should generate unique GUIDs


  2 passing (17ms)
```

This is a test for the generateGUID function, which generates unique GUIDs that are used as GUIDs such as impression_id, publisher_id, device_id, etc.

This test ensures that the function generates a GUID in the correct format for the GUID and that the generated GUID is unique.

getGeoInfoWithIp function

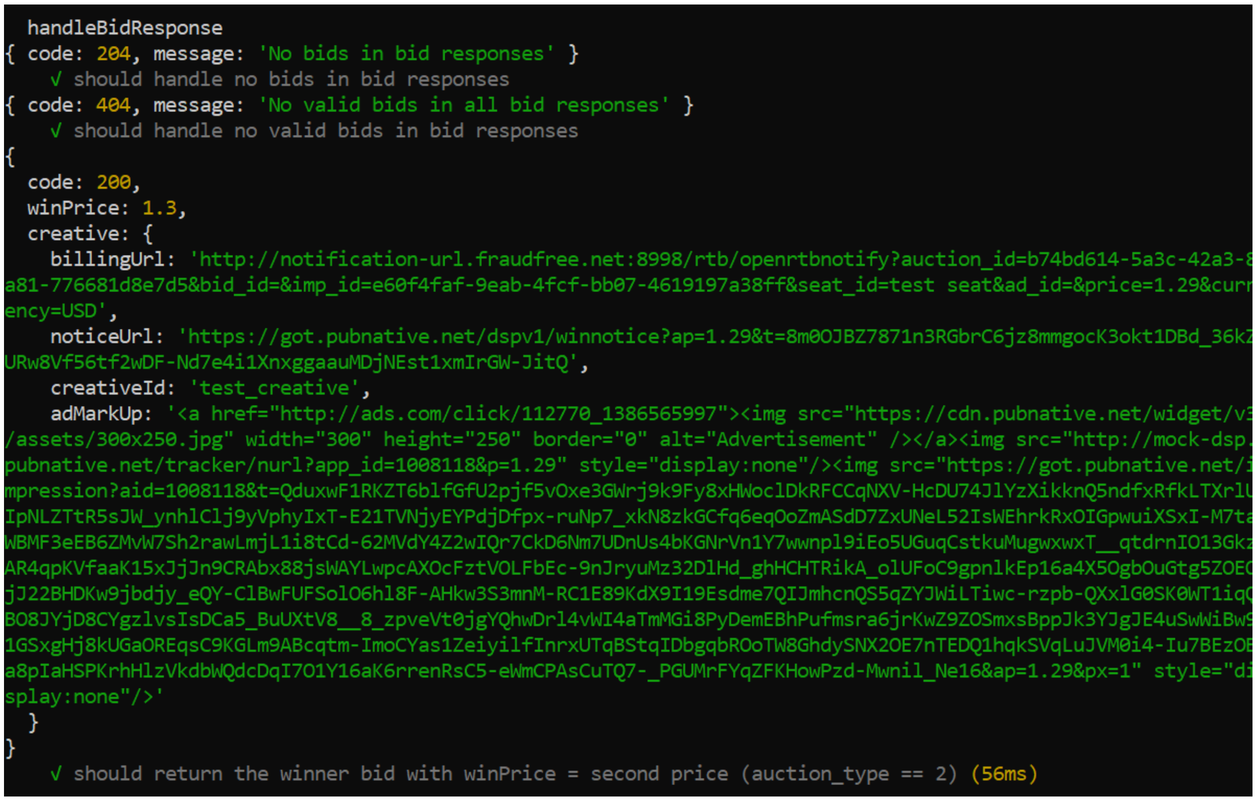


```
  getGeoInfoWithIP
country: {
  geonameId: 690791,
  isoCode: 'UA',
  names: {
    de: 'Ukraine',
    en: 'Ukraine',
    es: 'Ucrania',
    fr: 'Ukraine',
    ja: '□ □ □ □ □ □ □ □ ',
    'pt-BR': 'Ucrânia',
    ru: 'Украина',
    'zh-CN': '□ □ □ '
  }
}
location: {
  accuracyRadius: 20,
  latitude: 48.4735,
  longitude: 35.046,
  timeZone: 'Europe/Kyiv'
}
city: {
  geonameId: 709930,
  names: {
    de: 'Dnipro',
    en: 'Dnipro',
    es: 'Dnipró',
    fr: 'Dnipro',
    ja: '□ □ □ □ □ □ □ □ □ □ ',
    'pt-BR': 'Dnipro',
    ru: 'Днепр'
  }
}
postal: { code: '49000' }
    √ should get geo location info with ip address from maxmind db (94ms)
2024-06-11T06:54:32.684Z error : Error getting geo info: The address 127.0.0.1 is not in the database
    √ should handle any error in getting geo info from maxmind db (66ms)
```

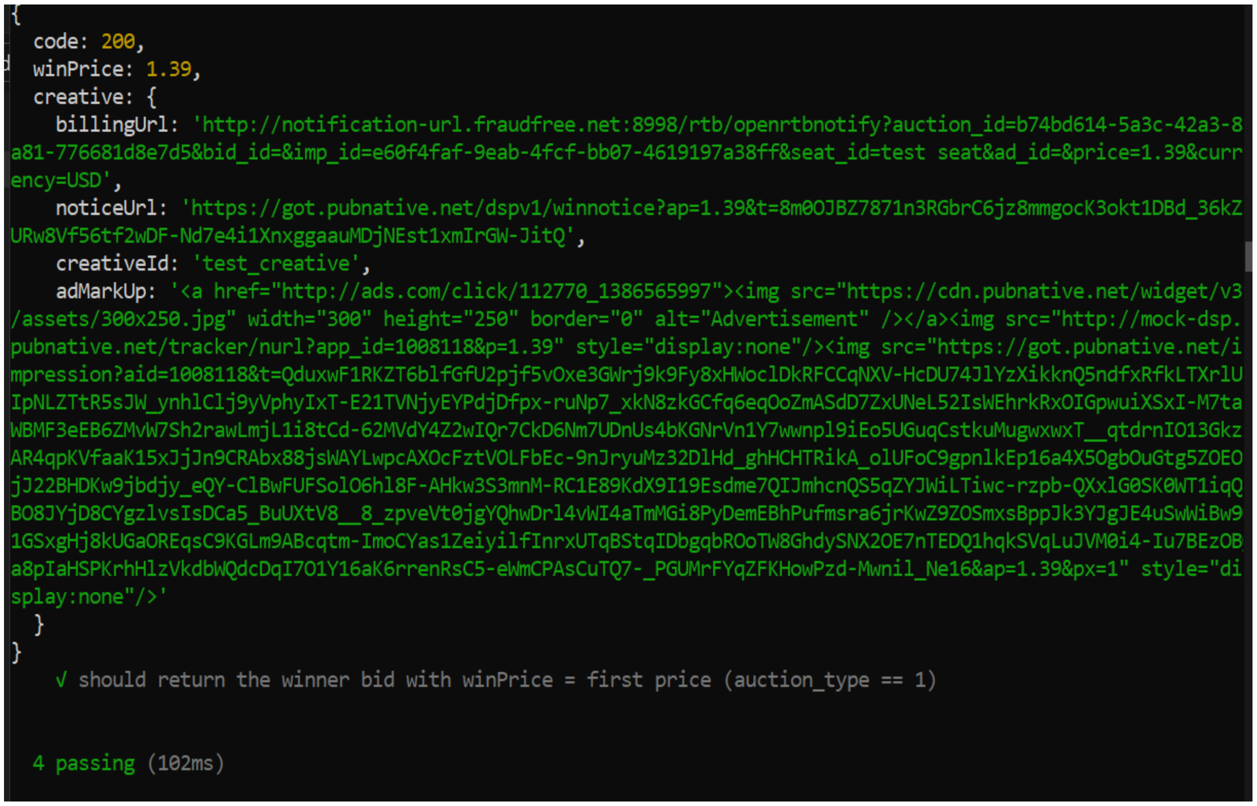
5.2.6 Test auction processing

- **npx mocha test/controllers/auction.controller.test.js**

Expected Results



```
  handleBidResponse
{ code: 204, message: 'No bids in bid responses' }
    √ should handle no bids in bid responses
{ code: 404, message: 'No valid bids in all bid responses' }
    √ should handle no valid bids in bid responses
{
  code: 200,
  winPrice: 1.3,
  creative: {
    billingUrl: 'http://notification-url.fraudfree.net:8998/rtb/openrtbnotify?auction_id=b74bd614-5a3c-42a3-
a81-776681d8e7d5&bid_id=&imp_id=e60f4faf-9eab-4fcf-bb07-4619197a38ff&seat_id=test seat&ad_id=&price=1.29&curr
ency=USD',
    noticeUrl: 'https://got.pubnative.net/dspv1/winnotice?ap=1.29&t=8m0OJBZ7871n3RGbrC6jz8mmgocK3okt1DBd_36k
URw8Vf56tf2wDF-Nd7e4i1XnxggaauMDjNEst1xmIrGW-JitQ',
    creativeId: 'test_creative',
    adMarkUp: '<a href="http://ads.com/click/112770_1386565997"><img src="https://cdn.pubnative.net/widget/v
/assets/300x250.jpg" width="300" height="250" border="0" alt="Advertisement" /></a><img src="http://mock-dsp
pubnative.net/tracker/nurl?app_id=1008118&p=1.29" style="display:none"/><img src="https://got.pubnative.net/
mpression?aid=1008118&t=QduxwF1RKZT6blfGfU2pjf5vOxe3GWrj9k9Fy8xHWoclDkRFCCqNXV-HcDU74JlYzXikknQ5ndfxRfkLTXrl
IpNLZTtR5sJW_ynhlClj9yVphyIxT-E21TVNjyEYPdjDfpx-ruNp7_xkN8zkGCfq6eqOoZmASdD7ZxUNeL52IsWEhrkRxOIGpwuiXSxI-M7t
WBMF3eEB6ZMvW7Sh2rawLmjL1i8tCd-62MVdY4Z2wIQr7CkD6Nm7UDnUs4bKGNrVn1Y7wwnpl9iEo5UGuqCstkuMugwxwxT__qtdrnIO13Gk
AR4qpKVfaaK15xJjJn9CRAbx88jsWAYLwpcAXOcFztVOLFbEc-9nJryuMz32DlHd_ghHCHTRikA_olUFoC9gpnlkEp16a4X5OgbOuGtg5ZOE
jJ22BHDKw9jbdjy_eQY-ClBwFUFSolO6hl8F-AHkw3S3mnM-RC1E89KdX9I19Esdme7QIJmhcnQS5qZYJWiLTiwc-rzpb-QXxlG0SK0WT1iq
BO8JYjD8CYgzlvsIsDCa5_BuUXtV8__8_zpveVt0jgYQhwDrl4vWI4aTmMGi8PyDemEBhPufmsra6jrKwZ9ZOSmxsBppJk3YJgJE4uSwWiBw
1GSxgHj8kUGaOREqsC9KGLm9ABcqtm-ImoCYas1ZeiyilfInrxUTqBStqIDbgqbROoTW8GhdySNX2OE7nTEDQ1hqkSVqLuJVM0i4-Iu7BEzO
a8pIaHSPKrhHlzVkdbWQdcDqI7O1Y16aK6rrenRsC5-eWmCPAsCuTQ7-_PGUMrFYqZFKHowPzd-Mwnil_Ne16&ap=1.29&px=1" style="d
splay:none"/>'
  }
}
    √ should return the winner bid with winPrice = second price (auction_type == 2) (56ms)
```



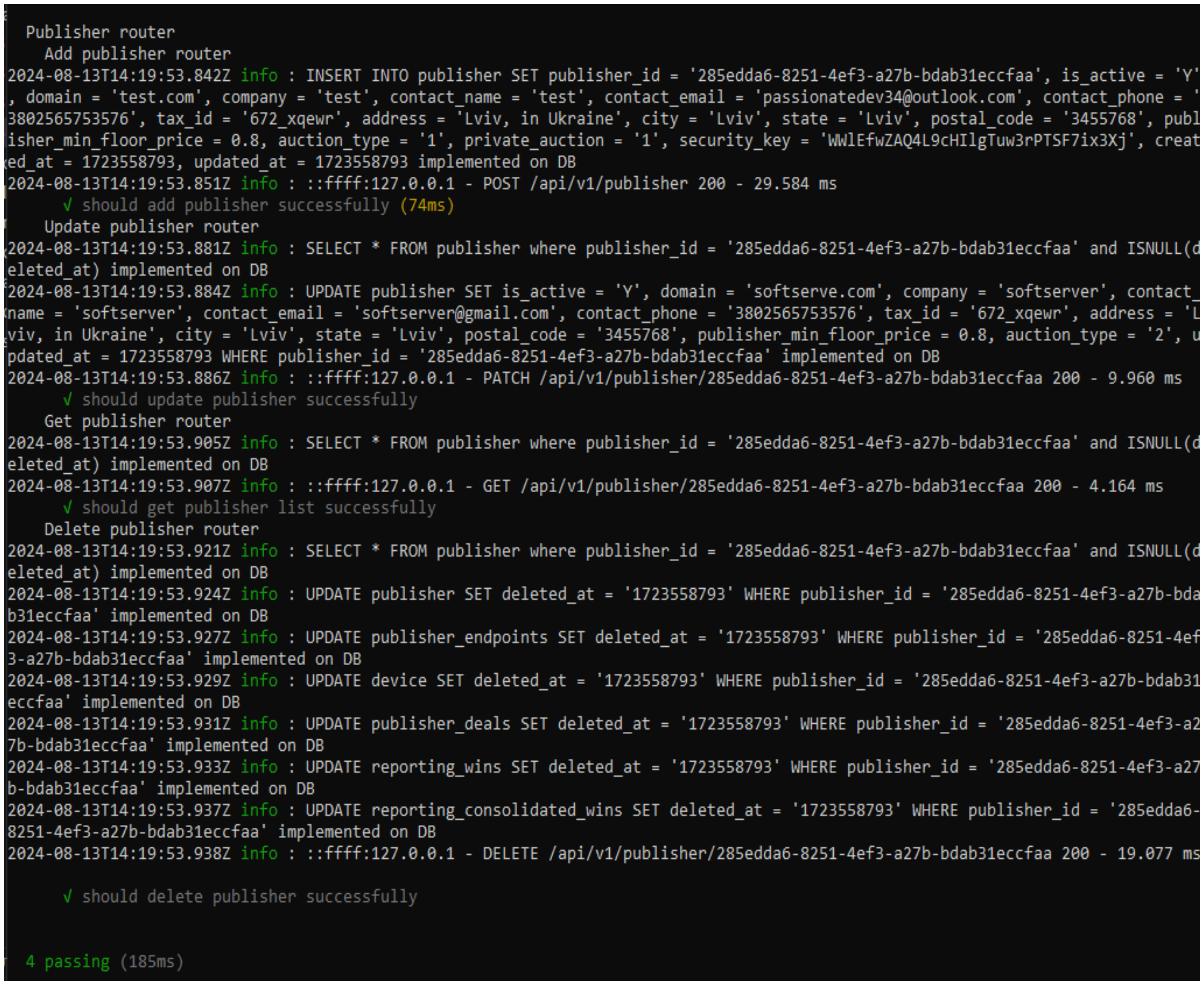
```
{
  code: 200,
  winPrice: 1.39,
  creative: {
    billingUrl: 'http://notification-url.fraudfree.net:8998/rtb/openrtbnotify?auction_id=b74bd614-5a3c-42a3-8
a81-776681d8e7d5&bid_id=&imp_id=e60f4faf-9eab-4fcf-bb07-4619197a38ff&seat_id=test seat&ad_id=&price=1.39&curr
ency=USD',
    noticeUrl: 'https://got.pubnative.net/dspv1/winnotice?ap=1.39&t=8m0OJBZ7871n3RGbrC6jz8mmgocK3okt1DBd_36kZ
URw8Vf56tf2wDF-Nd7e4i1XnxggaauMDjNEst1xmIrGW-JitQ',
    creativeId: 'test_creative',
    adMarkUp: '<a href="http://ads.com/click/112770_1386565997"><img src="https://cdn.pubnative.net/widget/v3
/assets/300x250.jpg" width="300" height="250" border="0" alt="Advertisement" /></a><img src="http://mock-dsp.
pubnative.net/tracker/nurl?app_id=1008118&p=1.39" style="display:none"/><img src="https://got.pubnative.net/i
mpression?aid=1008118&t=QduxwF1RKZT6blfGfU2pjf5vOxe3GWrj9k9Fy8xHWoclDkRFCCqNXV-HcDU74JlYzXikknQ5ndfxRfkLTXrlU
IpNLZTtR5sJW_ynhlClj9yVphyIxT-E21TVNjyEYPdjDfpx-ruNp7_xkN8zkGCfq6eqOoZmASdD7ZxUNeL52IsWEhrkRxOIGpwuiXSxI-M7ta
WBMF3eEB6ZMvW7Sh2rawLmjL1i8tCd-62MVdY4Z2wIQr7CkD6Nm7UDnUs4bKGNrVn1Y7wwnpl9iEo5UGuqCstkuMugwxwxT__qtdrnIO13Gkz
AR4qpKVfaaK15xJjJn9CRAbx88jsWAYLwpcAXOcFztVOLFbEc-9nJryuMz32DlHd_ghHCHTRikA_olUFoC9gpnlkEp16a4X5OgbOuGtg5ZOEO
jJ22BHDKw9jbdjy_eQY-ClBwFUFSolO6hl8F-AHkw3S3mnM-RC1E89KdX9I19Esdme7QIJmhcnQS5qZYJWiLTiwc-rzpb-QXxlG0SK0WT1iqQ
BO8JYjD8CYgzlvsIsDCa5_BuUXtV8__8_zpveVt0jgYQhwDrl4vWI4aTmMGi8PyDemEBhPufmsra6jrKwZ9ZOSmxsBppJk3YJgJE4uSwWiBw9
1GSxgHj8kUGaOREqsC9KGLm9ABcqtm-ImoCYas1ZeiyilfInrxUTqBStqIDbgqbROoTW8GhdySNX2OE7nTEDQ1hqkSVqLuJVM0i4-Iu7BEzOB
a8pIaHSPKrhHlzVkdbWQdcDqI7O1Y16aK6rrenRsC5-eWmCPAsCuTQ7-_PGUMrFYqZFKHowPzd-Mwnil_Ne16&ap=1.39&px=1" style="di
splay:none"/>'
  }
}
    √ should return the winner bid with winPrice = first price (auction_type == 1)


  4 passing (102ms)
```

5.2.7 Test RESTful APIs

5.2.7.1 Publisher APIs

- **npx mocha test/apis/publisher.router.test.js**

  Expected Result

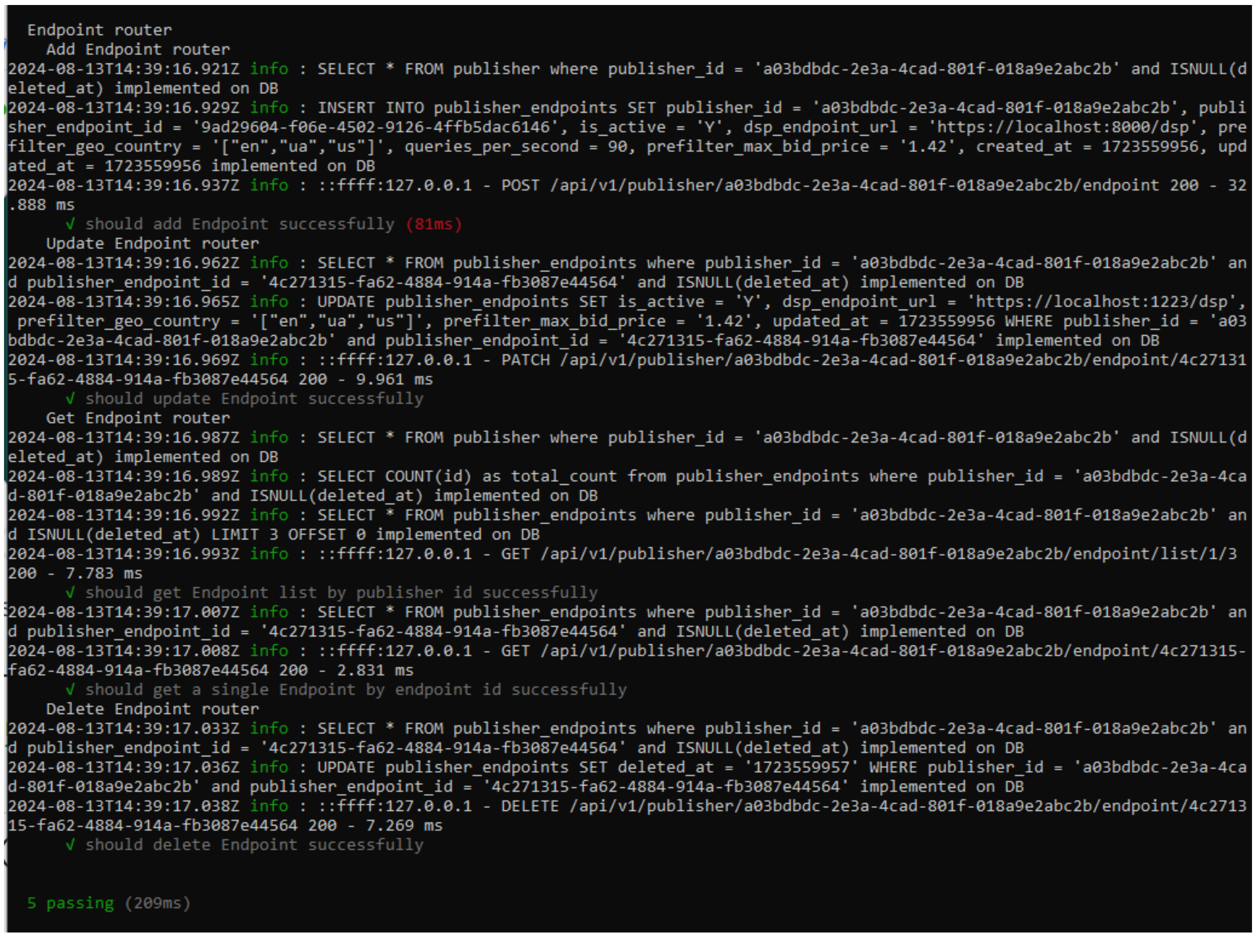


```
  Publisher router
    Add publisher router
2024-08-13T14:19:53.842Z info : INSERT INTO publisher SET publisher_id = '285edda6-8251-4ef3-a27b-bdab31eccfaa', is_active = 'Y'
, domain = 'test.com', company = 'test', contact_name = 'test', contact_email = 'passionatedev34@outlook.com', contact_phone = '
3802565753576', tax_id = '672_xqewr', address = 'Lviv, in Ukraine', city = 'Lviv', state = 'Lviv', postal_code = '3455768', publ
isher_min_floor_price = 0.8, auction_type = '1', private_auction = '1', security_key = 'WWlEfwZAQ4L9cHIlgTuw3rPTSF7ix3Xj', creat
ed_at = 1723558793, updated_at = 1723558793 implemented on DB
2024-08-13T14:19:53.851Z info : ::ffff:127.0.0.1 - POST /api/v1/publisher 200 - 29.584 ms
      √ should add publisher successfully (74ms)
    Update publisher router
2024-08-13T14:19:53.881Z info : SELECT * FROM publisher where publisher_id = '285edda6-8251-4ef3-a27b-bdab31eccfaa' and ISNULL(d
eleted_at) implemented on DB
2024-08-13T14:19:53.884Z info : UPDATE publisher SET is_active = 'Y', domain = 'softserve.com', company = 'softserver', contact_
name = 'softserver', contact_email = 'softserver@gmail.com', contact_phone = '3802565753576', tax_id = '672_xqewr', address = 'L
viv, in Ukraine', city = 'Lviv', state = 'Lviv', postal_code = '3455768', publisher_min_floor_price = 0.8, auction_type = '2', u
pdated_at = 1723558793 WHERE publisher_id = '285edda6-8251-4ef3-a27b-bdab31eccfaa' implemented on DB
2024-08-13T14:19:53.886Z info : ::ffff:127.0.0.1 - PATCH /api/v1/publisher/285edda6-8251-4ef3-a27b-bdab31eccfaa 200 - 9.960 ms
      √ should update publisher successfully
    Get publisher router
2024-08-13T14:19:53.905Z info : SELECT * FROM publisher where publisher_id = '285edda6-8251-4ef3-a27b-bdab31eccfaa' and ISNULL(d
eleted_at) implemented on DB
2024-08-13T14:19:53.907Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/285edda6-8251-4ef3-a27b-bdab31eccfaa 200 - 4.164 ms
      √ should get publisher list successfully
    Delete publisher router
2024-08-13T14:19:53.921Z info : SELECT * FROM publisher where publisher_id = '285edda6-8251-4ef3-a27b-bdab31eccfaa' and ISNULL(d
eleted_at) implemented on DB
2024-08-13T14:19:53.924Z info : UPDATE publisher SET deleted_at = '1723558793' WHERE publisher_id = '285edda6-8251-4ef3-a27b-bda
b31eccfaa' implemented on DB
2024-08-13T14:19:53.927Z info : UPDATE publisher_endpoints SET deleted_at = '1723558793' WHERE publisher_id = '285edda6-8251-4ef
3-a27b-bdab31eccfaa' implemented on DB
2024-08-13T14:19:53.929Z info : UPDATE device SET deleted_at = '1723558793' WHERE publisher_id = '285edda6-8251-4ef3-a27b-bdab31
eccfaa' implemented on DB
2024-08-13T14:19:53.931Z info : UPDATE publisher_deals SET deleted_at = '1723558793' WHERE publisher_id = '285edda6-8251-4ef3-a2
7b-bdab31eccfaa' implemented on DB
2024-08-13T14:19:53.933Z info : UPDATE reporting_wins SET deleted_at = '1723558793' WHERE publisher_id = '285edda6-8251-4ef3-a27
b-bdab31eccfaa' implemented on DB
2024-08-13T14:19:53.937Z info : UPDATE reporting_consolidated_wins SET deleted_at = '1723558793' WHERE publisher_id = '285edda6-
8251-4ef3-a27b-bdab31eccfaa' implemented on DB
2024-08-13T14:19:53.938Z info : ::ffff:127.0.0.1 - DELETE /api/v1/publisher/285edda6-8251-4ef3-a27b-bdab31eccfaa 200 - 19.077 ms

      √ should delete publisher successfully


  4 passing (185ms)
```

5.2.7.2 Publisher Endpoint Apis

- **npx mocha test/apis/endpoint.router.test.js**

  Expected Result

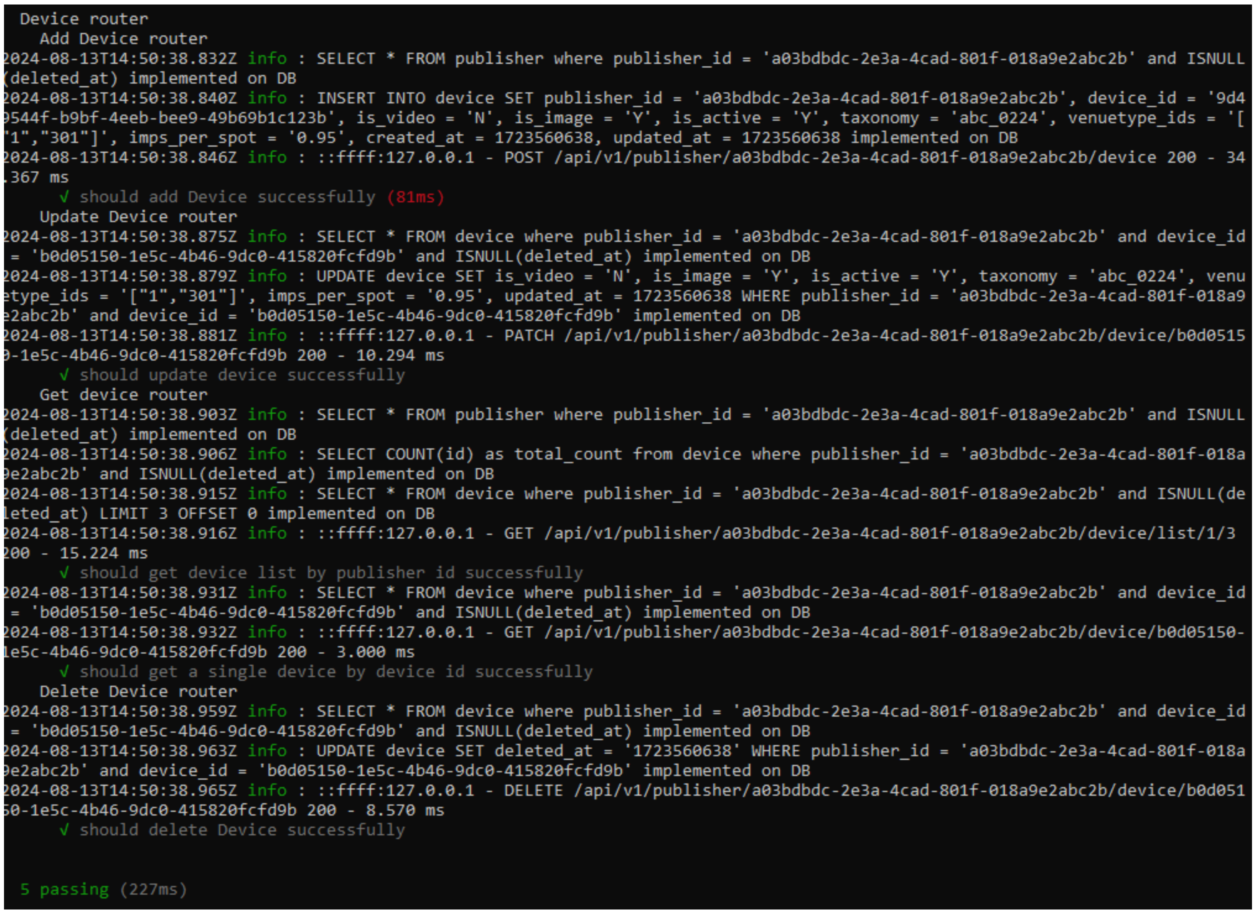


```
  Endpoint router
    Add Endpoint router
2024-08-13T14:39:16.921Z info : SELECT * FROM publisher where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL(d
eleted_at) implemented on DB
2024-08-13T14:39:16.929Z info : INSERT INTO publisher_endpoints SET publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b', publi
sher_endpoint_id = '9ad29604-f06e-4502-9126-4ffb5dac6146', is_active = 'Y', dsp_endpoint_url = 'https://localhost:8000/dsp', pre
filter_geo_country = '["en","ua","us"]', queries_per_second = 90, prefilter_max_bid_price = '1.42', created_at = 1723559956, upd
ated_at = 1723559956 implemented on DB
2024-08-13T14:39:16.937Z info : ::ffff:127.0.0.1 - POST /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/endpoint 200 - 32
.888 ms
      √ should add Endpoint successfully (81ms)
    Update Endpoint router
2024-08-13T14:39:16.962Z info : SELECT * FROM publisher_endpoints where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' an
d publisher_endpoint_id = '4c271315-fa62-4884-914a-fb3087e44564' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:39:16.965Z info : UPDATE publisher_endpoints SET is_active = 'Y', dsp_endpoint_url = 'https://localhost:1223/dsp',
 prefilter_geo_country = '["en","ua","us"]', prefilter_max_bid_price = '1.42', updated_at = 1723559956 WHERE publisher_id = 'a03
bdbdc-2e3a-4cad-801f-018a9e2abc2b' and publisher_endpoint_id = '4c271315-fa62-4884-914a-fb3087e44564' implemented on DB
2024-08-13T14:39:16.969Z info : ::ffff:127.0.0.1 - PATCH /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/endpoint/4c27131
5-fa62-4884-914a-fb3087e44564 200 - 9.961 ms
      √ should update Endpoint successfully
    Get Endpoint router
2024-08-13T14:39:16.987Z info : SELECT * FROM publisher where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL(d
eleted_at) implemented on DB
2024-08-13T14:39:16.989Z info : SELECT COUNT(id) as total_count from publisher_endpoints where publisher_id = 'a03bdbdc-2e3a-4ca
d-801f-018a9e2abc2b' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:39:16.992Z info : SELECT * FROM publisher_endpoints where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' an
d ISNULL(deleted_at) LIMIT 3 OFFSET 0 implemented on DB
2024-08-13T14:39:16.993Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/endpoint/list/1/3
200 - 7.783 ms
      √ should get Endpoint list by publisher id successfully
2024-08-13T14:39:17.007Z info : SELECT * FROM publisher_endpoints where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' an
d publisher_endpoint_id = '4c271315-fa62-4884-914a-fb3087e44564' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:39:17.008Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/endpoint/4c271315-
fa62-4884-914a-fb3087e44564 200 - 2.831 ms
      √ should get a single Endpoint by endpoint id successfully
    Delete Endpoint router
2024-08-13T14:39:17.033Z info : SELECT * FROM publisher_endpoints where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' an
d publisher_endpoint_id = '4c271315-fa62-4884-914a-fb3087e44564' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:39:17.036Z info : UPDATE publisher_endpoints SET deleted_at = '1723559957' WHERE publisher_id = 'a03bdbdc-2e3a-4ca
d-801f-018a9e2abc2b' and publisher_endpoint_id = '4c271315-fa62-4884-914a-fb3087e44564' implemented on DB
2024-08-13T14:39:17.038Z info : ::ffff:127.0.0.1 - DELETE /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/endpoint/4c2713
15-fa62-4884-914a-fb3087e44564 200 - 7.269 ms
      √ should delete Endpoint successfully


  5 passing (209ms)
```

5.2.7.3 Device Apis

- **npx mocha test/apis/device.router.test.js**

  Expected Result

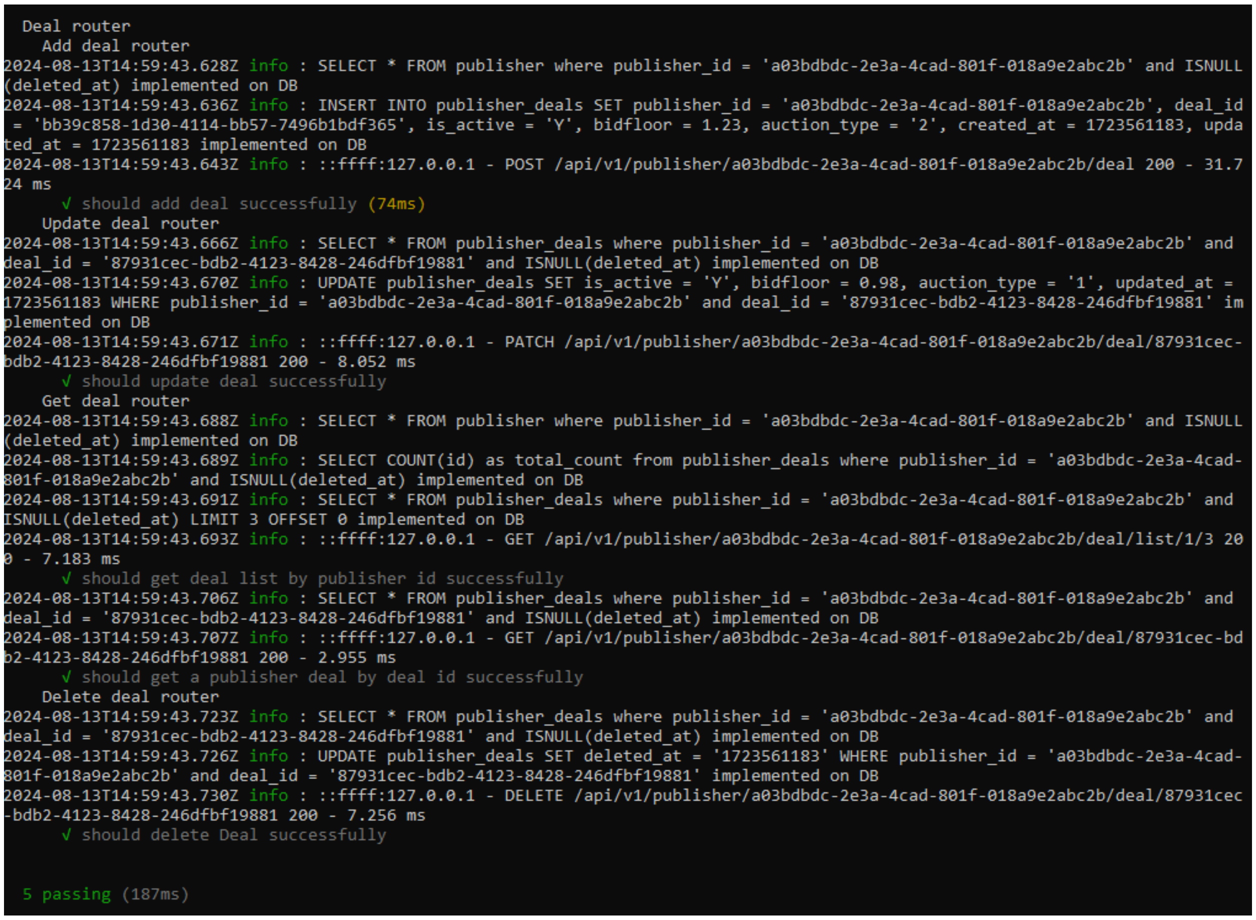


```
 Device router
   Add Device router
2024-08-13T14:50:38.832Z info : SELECT * FROM publisher where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL
(deleted_at) implemented on DB
2024-08-13T14:50:38.840Z info : INSERT INTO device SET publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b', device_id = '9d4
9544f-b9bf-4eeb-bee9-49b69b1c123b', is_video = 'N', is_image = 'Y', is_active = 'Y', taxonomy = 'abc_0224', venuetype_ids = '[
"1","301"]', imps_per_spot = '0.95', created_at = 1723560638, updated_at = 1723560638 implemented on DB
2024-08-13T14:50:38.846Z info : ::ffff:127.0.0.1 - POST /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/device 200 - 34
.367 ms
     √ should add Device successfully (81ms)
   Update Device router
2024-08-13T14:50:38.875Z info : SELECT * FROM device where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and device_id
 = 'b0d05150-1e5c-4b46-9dc0-415820fcfd9b' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:50:38.879Z info : UPDATE device SET is_video = 'N', is_image = 'Y', is_active = 'Y', taxonomy = 'abc_0224', venu
etype_ids = '["1","301"]', imps_per_spot = '0.95', updated_at = 1723560638 WHERE publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9
e2abc2b' and device_id = 'b0d05150-1e5c-4b46-9dc0-415820fcfd9b' implemented on DB
2024-08-13T14:50:38.881Z info : ::ffff:127.0.0.1 - PATCH /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/device/b0d0515
0-1e5c-4b46-9dc0-415820fcfd9b 200 - 10.294 ms
     √ should update device successfully
   Get device router
2024-08-13T14:50:38.903Z info : SELECT * FROM publisher where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL
(deleted_at) implemented on DB
2024-08-13T14:50:38.906Z info : SELECT COUNT(id) as total_count from device where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a
9e2abc2b' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:50:38.915Z info : SELECT * FROM device where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL(de
leted_at) LIMIT 3 OFFSET 0 implemented on DB
2024-08-13T14:50:38.916Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/device/list/1/3
200 - 15.224 ms
     √ should get device list by publisher id successfully
2024-08-13T14:50:38.931Z info : SELECT * FROM device where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and device_id
 = 'b0d05150-1e5c-4b46-9dc0-415820fcfd9b' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:50:38.932Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/device/b0d05150-
1e5c-4b46-9dc0-415820fcfd9b 200 - 3.000 ms
     √ should get a single device by device id successfully
   Delete Device router
2024-08-13T14:50:38.959Z info : SELECT * FROM device where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and device_id
 = 'b0d05150-1e5c-4b46-9dc0-415820fcfd9b' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:50:38.963Z info : UPDATE device SET deleted_at = '1723560638' WHERE publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a
9e2abc2b' and device_id = 'b0d05150-1e5c-4b46-9dc0-415820fcfd9b' implemented on DB
2024-08-13T14:50:38.965Z info : ::ffff:127.0.0.1 - DELETE /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/device/b0d051
50-1e5c-4b46-9dc0-415820fcfd9b 200 - 8.570 ms
     √ should delete Device successfully


 5 passing (227ms)
```

5.2.7.4 Publisher deal Apis

- **npx mocha test/apis/deal.router.test.js**

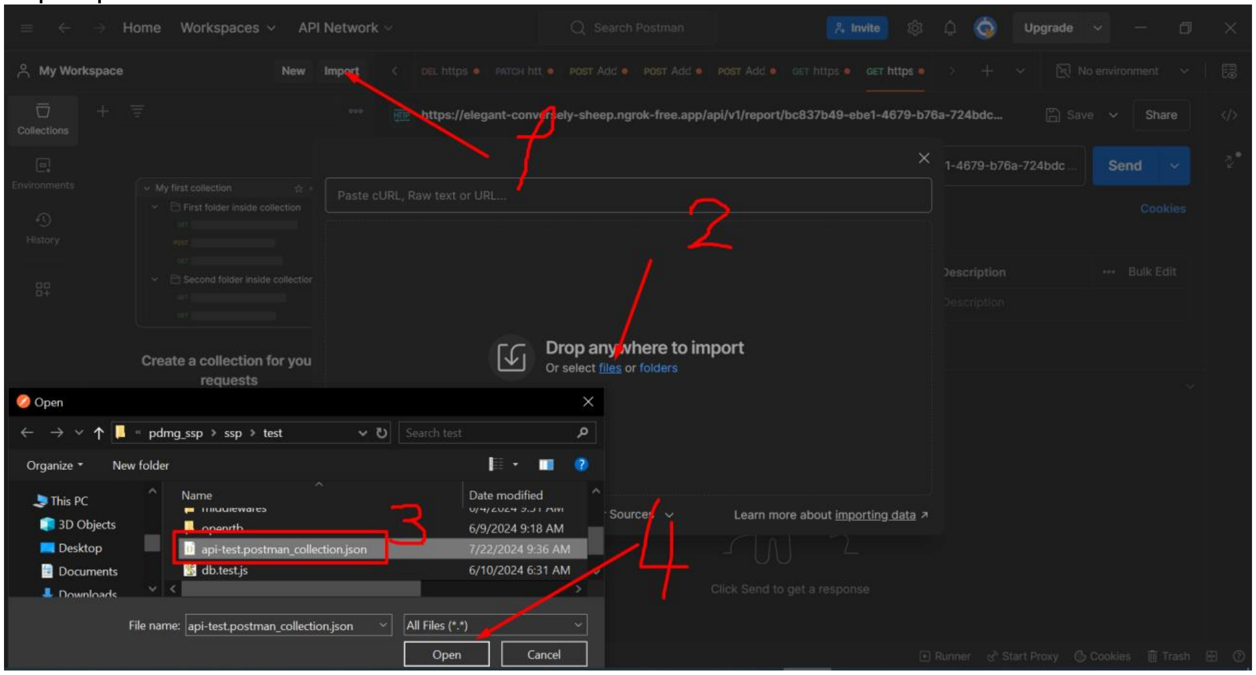
Expected Result



```
  Deal router
    Add deal router
2024-08-13T14:59:43.628Z info : SELECT * FROM publisher where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL
(deleted_at) implemented on DB
2024-08-13T14:59:43.636Z info : INSERT INTO publisher_deals SET publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b', deal_id
 = 'bb39c858-1d30-4114-bb57-7496b1bdf365', is_active = 'Y', bidfloor = 1.23, auction_type = '2', created_at = 1723561183, upda
ted_at = 1723561183 implemented on DB
2024-08-13T14:59:43.643Z info : ::ffff:127.0.0.1 - POST /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/deal 200 - 31.7
24 ms
      √ should add deal successfully (74ms)
    Update deal router
2024-08-13T14:59:43.666Z info : SELECT * FROM publisher_deals where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and
deal_id = '87931cec-bdb2-4123-8428-246dfbf19881' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:59:43.670Z info : UPDATE publisher_deals SET is_active = 'Y', bidfloor = 0.98, auction_type = '1', updated_at =
1723561183 WHERE publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and deal_id = '87931cec-bdb2-4123-8428-246dfbf19881' im
plemented on DB
2024-08-13T14:59:43.671Z info : ::ffff:127.0.0.1 - PATCH /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/deal/87931cec-
bdb2-4123-8428-246dfbf19881 200 - 8.052 ms
      √ should update deal successfully
    Get deal router
2024-08-13T14:59:43.688Z info : SELECT * FROM publisher where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and ISNULL
(deleted_at) implemented on DB
2024-08-13T14:59:43.689Z info : SELECT COUNT(id) as total_count from publisher_deals where publisher_id = 'a03bdbdc-2e3a-4cad-
801f-018a9e2abc2b' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:59:43.691Z info : SELECT * FROM publisher_deals where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and
ISNULL(deleted_at) LIMIT 3 OFFSET 0 implemented on DB
2024-08-13T14:59:43.693Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/deal/list/1/3 20
0 - 7.183 ms
      √ should get deal list by publisher id successfully
2024-08-13T14:59:43.706Z info : SELECT * FROM publisher_deals where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and
deal_id = '87931cec-bdb2-4123-8428-246dfbf19881' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:59:43.707Z info : ::ffff:127.0.0.1 - GET /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/deal/87931cec-bd
b2-4123-8428-246dfbf19881 200 - 2.955 ms
      √ should get a publisher deal by deal id successfully
    Delete deal router
2024-08-13T14:59:43.723Z info : SELECT * FROM publisher_deals where publisher_id = 'a03bdbdc-2e3a-4cad-801f-018a9e2abc2b' and
deal_id = '87931cec-bdb2-4123-8428-246dfbf19881' and ISNULL(deleted_at) implemented on DB
2024-08-13T14:59:43.726Z info : UPDATE publisher_deals SET deleted_at = '1723561183' WHERE publisher_id = 'a03bdbdc-2e3a-4cad-
801f-018a9e2abc2b' and deal_id = '87931cec-bdb2-4123-8428-246dfbf19881' implemented on DB
2024-08-13T14:59:43.730Z info : ::ffff:127.0.0.1 - DELETE /api/v1/publisher/a03bdbdc-2e3a-4cad-801f-018a9e2abc2b/deal/87931cec
-bdb2-4123-8428-246dfbf19881 200 - 7.256 ms
      √ should delete Deal successfully


  5 passing (187ms)
```

6. Test RESTful APIs using Postman

- Open Postman
- Import postman file


Import
1
2
Drop anywhere to import
Or select files or folders
Paste cURL, Raw text or URL...
3
4
api-test.postman_collection.json
Open
Cancel

- Run the collection

NOTE: All APIs MUST include access key (which can be either Admin key or API key for a specific publisher) at headers info.

Explanation for each API is as next Chapter.

7. APIs

7.1 Publisher APIs

7.1.1 Add publisher – add a new publisher with admin access and retrieve publisher id and access key of the publisher.

- router: /api/v1/publisher

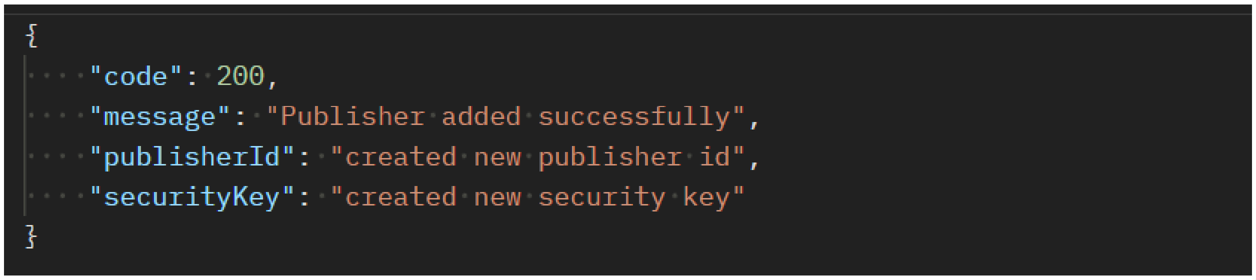
- Method: POST

- Headers: [{ “x-api-key”: “Admin access key” }]

- body: {

```
    "isActive": "Y",
    "domain": "softserve.com",
    "company": "softserver",
    "contactName": "softserver",
    "contactEmail": "passionatedev34@outlook.com",
    "contactPhone": "3802565753576",
    "taxId": "672_xqewr",
    "address": "Lviv, in Ukraine",
    "city": "Lviv",
    "state": "Lviv",
    "postalCode": "3455768",
    "publisherMinFloorPrice": "0.8",
    "auctionType": "1",
    "privateAuction": 1
}
```

- Successful Response



```
{
    "code": 200,
    "message": "Publisher added successfully",
    "publisherId": "created new publisher id",
    "securityKey": "created new security key"
}
```

**** The security key can be used in all apis just for the publisher on our system.***

7.1.2 Update publisher – update a publisher by publisher id

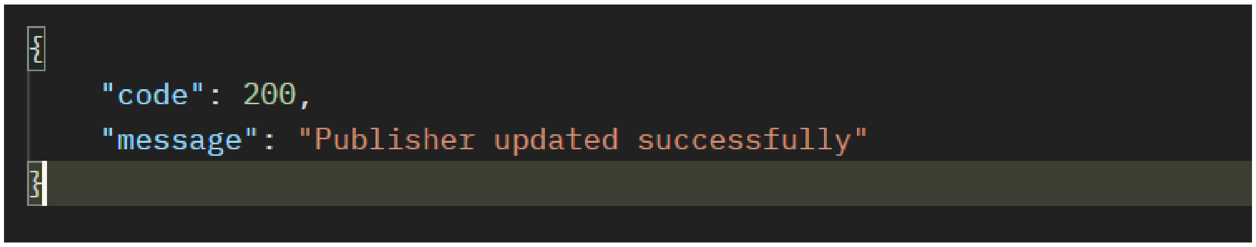
- router: /api/v1/publisher/:publisherId

- method: PATCH

- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]

- body: {
    "isActive": "Y",
    "domain": "softserve.com",
    "company": "softserver",
    "contactName": "softserver",
    "contactEmail": "softserver@gmail.com",
    "contactPhone": "3802565753576",
    "taxId": "672_xqewr",
    "address": "Lviv, in Ukraine",
    "city": "Lviv",
    "state": "Lviv",
    "postalCode": "3455768",
    "publisherMinFloorPrice": "0.8",
    "auctionType": "2"
  }

- Successful Response



```
{
    "code": 200,
    "message": "Publisher updated successfully"
}
```

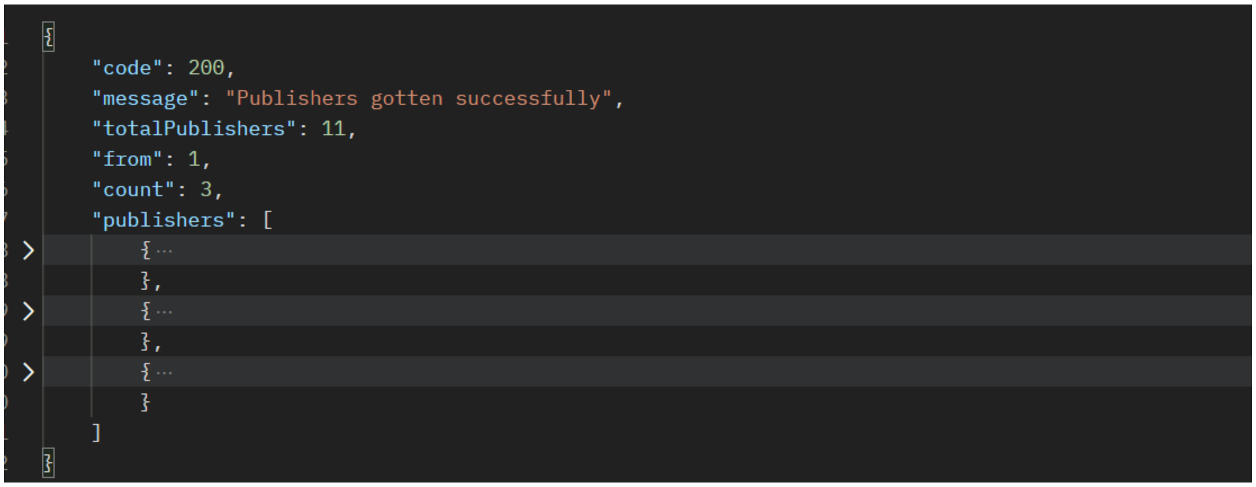
7.1.3 Get publisher list – retrieve desired number of publishers from specific position in our database

- router: /api/v1/publisher/list/:start/:limit

- method: GET

- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]

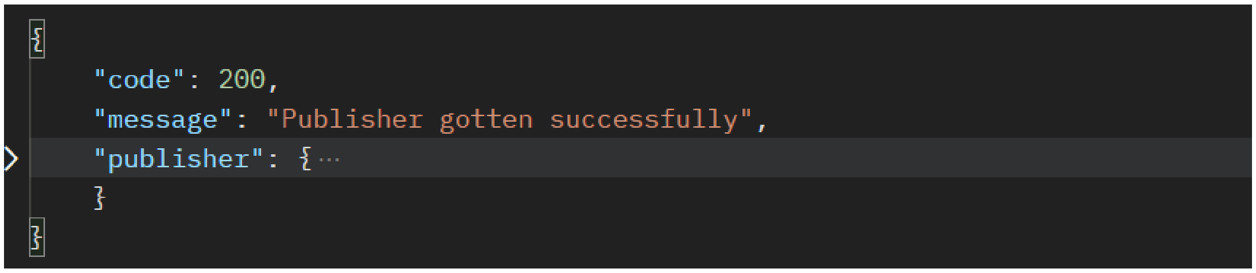
- Successful Response



```
{
    "code": 200,
    "message": "Publishers gotten successfully",
    "totalPublishers": 11,
    "from": 1,
    "count": 3,
    "publishers": [
        {…
        },
        {…
        },
        {…
        }
    ]
}
```

7.1.4 Get a publisher – retrieve a publisher by publisher id

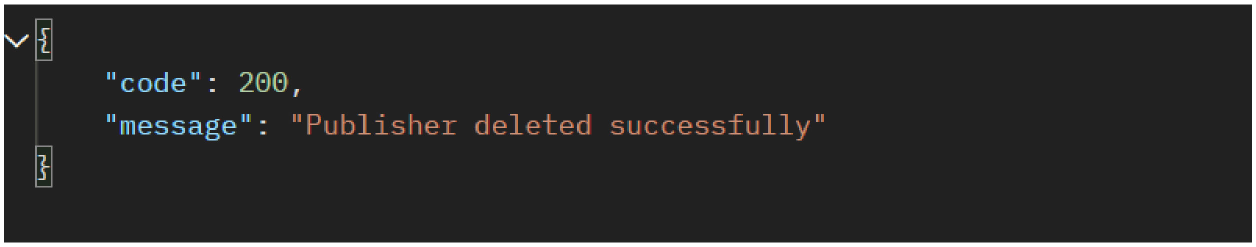
- router: /api/v1/publisher/:publisherId
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response



```
{
    "code": 200,
    "message": "Publisher gotten successfully",
    "publisher": {…
    }
}
```

7.1.5 Delete a publisher – delete a publisher by publisher id

- router: /api/v1/publisher/:publisherId
- method: DELETE
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response

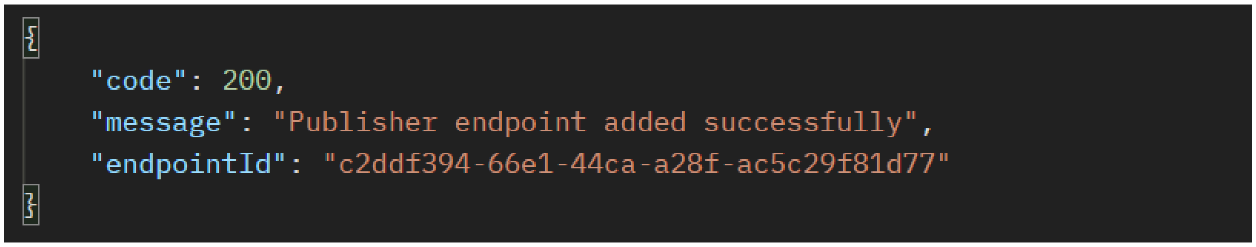


```
{
    "code": 200,
    "message": "Publisher deleted successfully"
}
```

7.2 Publisher Endpoint APIs

7.2.1 Add endpoint – add a publisher endpoint for a specific publisher and retrieve the endpoint id

- router: /api/v1/publisher/:publisherId/endpoint
- Method: POST
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- body: {
  "dspEndpointUrl": "http://localhost:8000 /dsp",
  "queriesPerSecond": 90,
  "prefilterGeoCountry": ["en","ua", "us"],
  "prefilterMaxBidPrice": 1.42,
  "isActive": "Y"
  }
- Successful Response



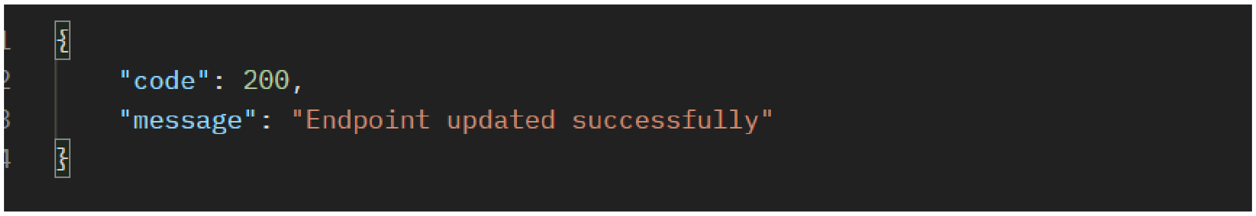
```
{
    "code": 200,
    "message": "Publisher endpoint added successfully",
    "endpointId": "c2ddf394-66e1-44ca-a28f-ac5c29f81d77"
}
```

7.2.2 Update endpoint – update a publisher endpoint by endpoint id for a specific publisher

- router: /api/v1/publisher/:publisherId/endpoint/:endpointId
- method: PATCH
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- body: {
  "dspEndpointUrl": "http://localhost:8001 /dsp",

```
    "queriesPerSecond": 100,
    "prefilterGeoCountry": ["en","ua", "us"],
    "prefilterMaxBidPrice": 1.42,
    "isActive": "Y"
}
```

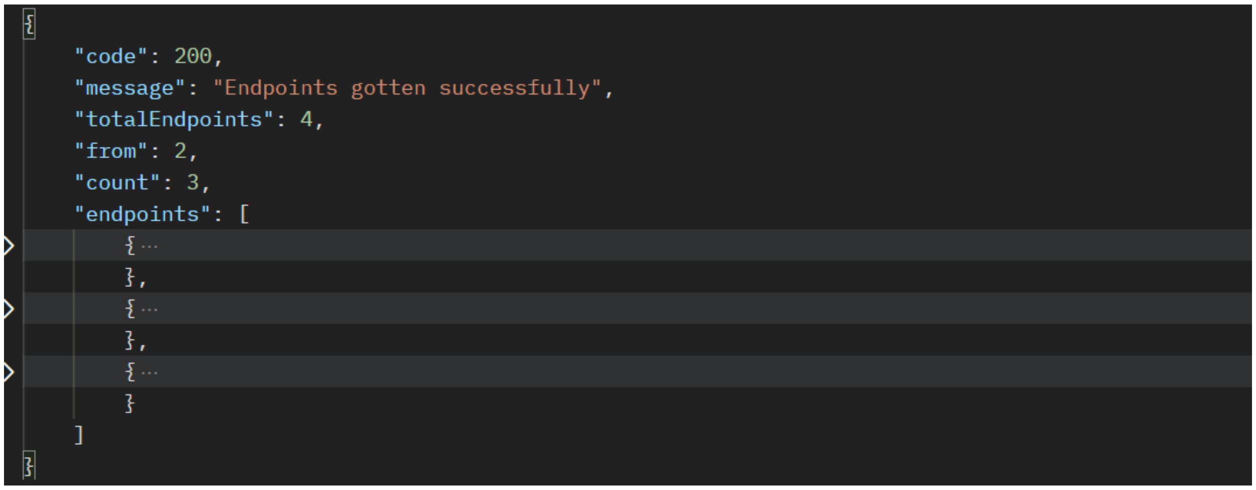
- Successful Response



```
{
    "code": 200,
    "message": "Endpoint updated successfully"
}
```

7.2.3 Get endpoints list by publisher id – retrieve desired number of endpoints specific position for a specific publisher

- router: /api/v1/publisher/:publisherId/endpoint/list/:start/:limit
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response

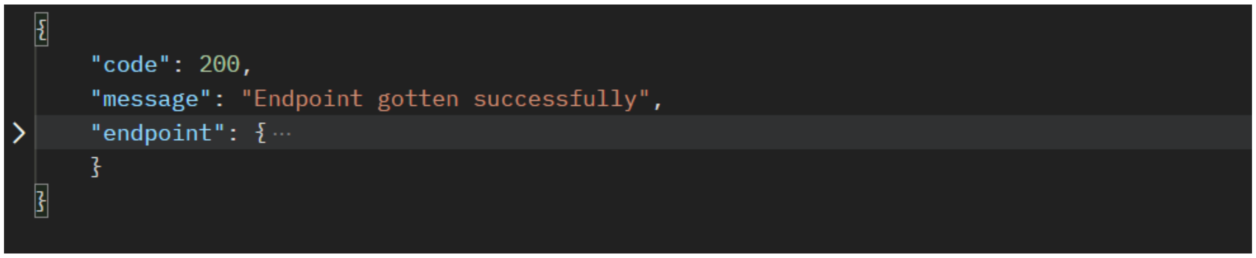


```
{
    "code": 200,
    "message": "Endpoints gotten successfully",
    "totalEndpoints": 4,
    "from": 2,
    "count": 3,
    "endpoints": [
        {…
        },
        {…
        },
        {…
        }
    ]
}
```

7.2.4 Get a publisher endpoint – retrieve a publisher endpoint by endpoint id for a specific publisher

- router: /api/v1/publisher/:publisherId/endpoint/:endpointId
- method: GET

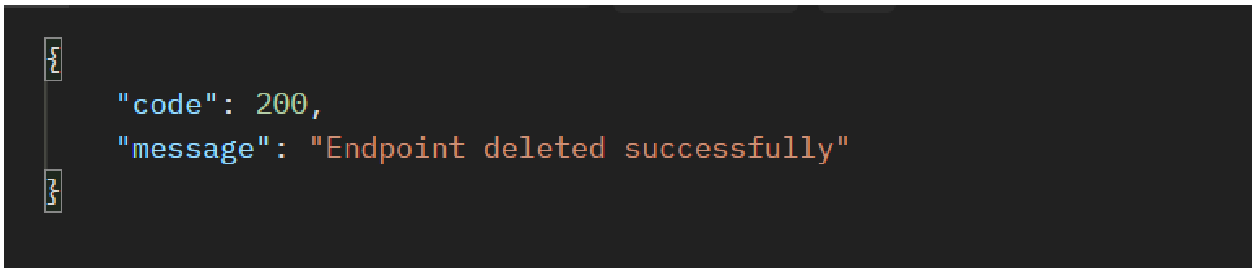
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response



```
{
    "code": 200,
    "message": "Endpoint gotten successfully",
    "endpoint": {⋯
    }
}
```

7.2.5 Delete a publisher endpoint – delete a publisher endpoint by endpoint id for a specific publisher

- router: /api/v1/publisher/:publisherId/endpoint/:endpointId
- method: DELETE
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response



```
{
    "code": 200,
    "message": "Endpoint deleted successfully"
}
```

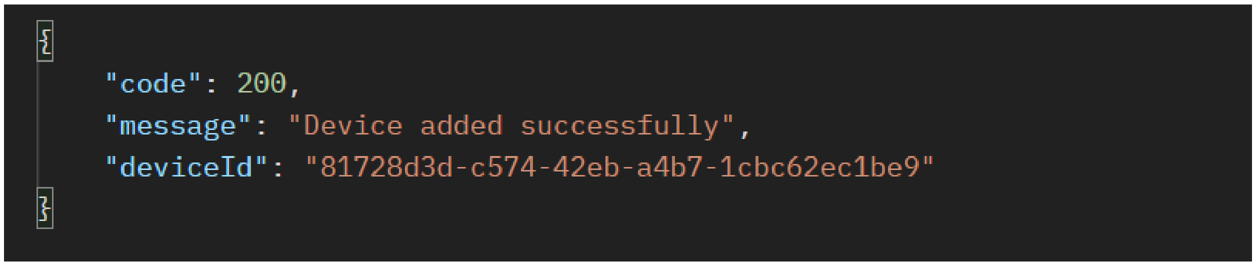
7.3 Device APIs

7.3.1 Add device – add a new device and retrieve the device id

- router: /api/v1/publisher/:publisherId/device
- Method: POST
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- body: {
    "isVideo": "N",
    "isImage": "Y",
    "isActive": "Y",
    "taxonomy": "abc_0224",
    "venuetypeIds": [

```
        "1",
        "301"
    ],
    "impsPerSpot": 0.95
}
```

- Successful Response



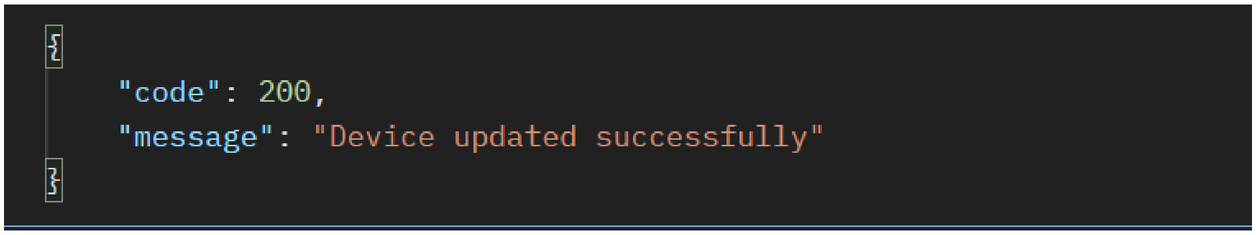
```
{
    "code": 200,
    "message": "Device added successfully",
    "deviceId": "81728d3d-c574-42eb-a4b7-1cbc62ec1be9"
}
```

7.3.2 Update device – update a device by device id for a specific publisher

- router: /api/v1/publisher/:publisherId/device/:deviceId
- method: PATCH
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- body:

```
{
    "isVideo": "N",
    "isImage": "Y",
    "isActive": "Y",
    "taxonomy": "abc_0224",
    "venuetypeIds": [
        "1",
        "301"
    ],
    "impsPerSpot": 0.98
}
```

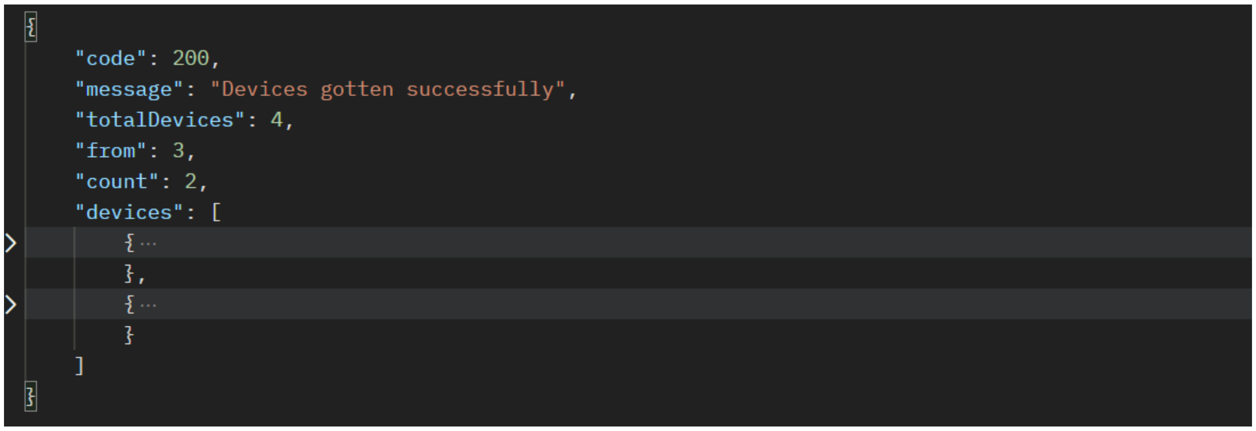
- Successful Response



```
{
    "code": 200,
    "message": "Device updated successfully"
}
```

7.3.3 Get device list by publisher id – retrieve desired number of devices from specific position for a specific publisher

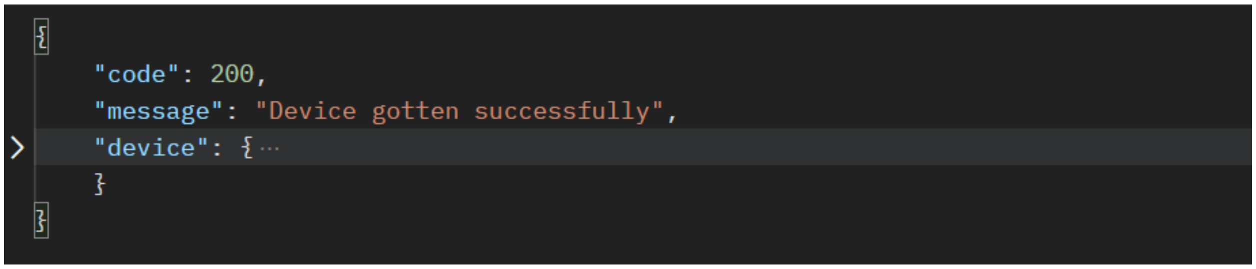
- router: /api/v1/publisher/:publisherId/device/list/:start/:limit
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response



```
{
    "code": 200,
    "message": "Devices gotten successfully",
    "totalDevices": 4,
    "from": 3,
    "count": 2,
    "devices": [
        {…
        },
        {…
        }
    ]
}
```

7.3.4 Get a device – retrieve a device by device id for a specific publisher

- router: /api/v1/publisher/:publisherId/device/:deviceId
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response

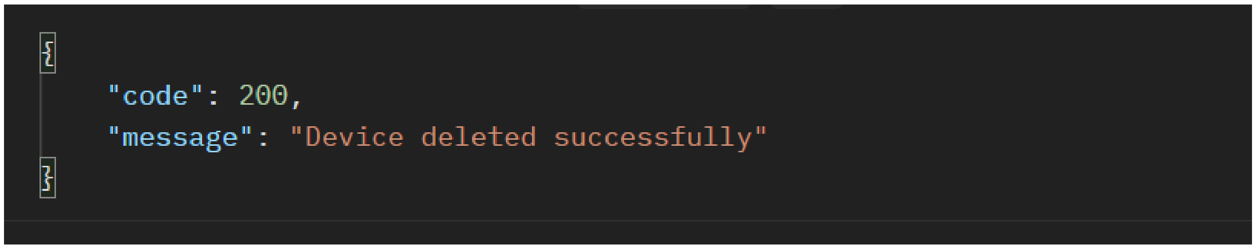


```
{
    "code": 200,
    "message": "Device gotten successfully",
    "device": {…
    }
}
```

7.3.5 Delete a device – delete a device by device id for a specific publisher

- router: /api/v1/publisher/:publisherId/device/:deviceId
- method: DELETE

- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response

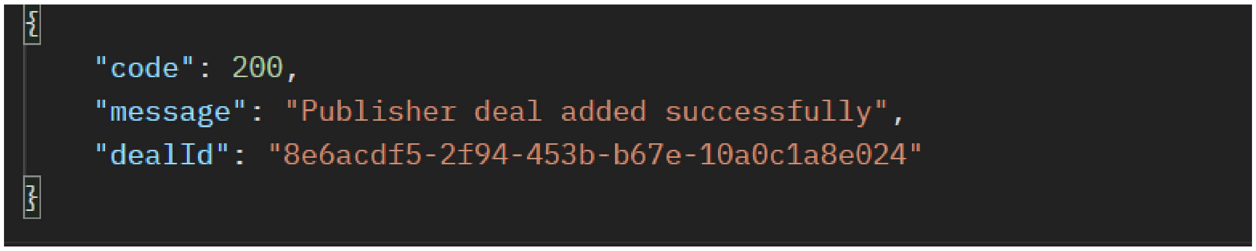


```
{
    "code": 200,
    "message": "Device deleted successfully"
}
```

7.4 Publisher Deal APIs

7.4.1 Add a publisher deal – add a new publisher deal and retrieve the deal id

- router: /api/v1/publisher/:publisherId/deal
- Method: POST
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- body: {
    "isActive": "Y",
    "bidfloor": 1.23,
    "auctionType": "2"
  }
- Successful Response



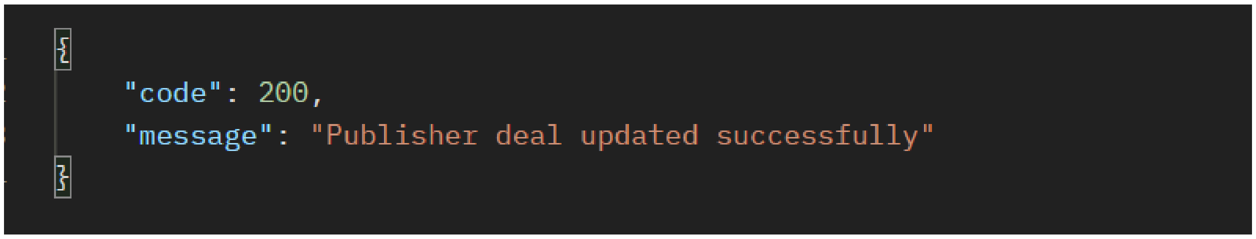
```
{
    "code": 200,
    "message": "Publisher deal added successfully",
    "dealId": "8e6acdf5-2f94-453b-b67e-10a0c1a8e024"
}
```

7.4.2 Update publisher deal – update a publisher deal by deal id for a specific publisher

- router: /api/v1/publisher/:publisherId/deal/:dealId
- method: PATCH
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- body: {

```
    "isActive": "Y",
    "bidfloor": 0.98,
    "auctionType": "2"
}
```

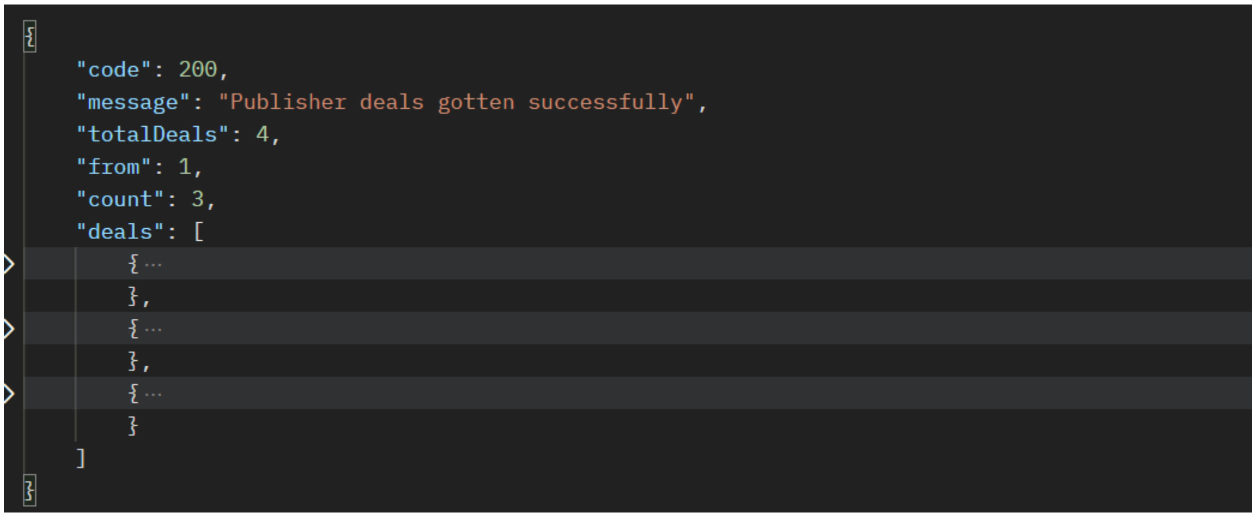
- Successful Response



```
{
    "code": 200,
    "message": "Publisher deal updated successfully"
}
```

7.4.3 Get publisher deal list by publisher id – retrieve desired number of publisher deals from specific position in our database

- router: /api/v1/publisher/:publisherId/deal/list/:start/:limit
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response

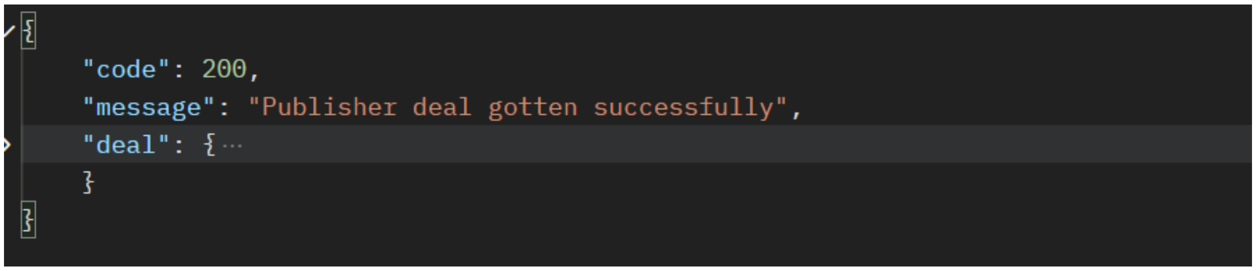


```
{
    "code": 200,
    "message": "Publisher deals gotten successfully",
    "totalDeals": 4,
    "from": 1,
    "count": 3,
    "deals": [
        {…
        },
        {…
        },
        {…
        }
    ]
}
```

7.4.4 Get a publisher deal – retrieve a publisher deal by deal id for a specific publisher

- router: /api/v1/publisher/:publisherId/deal/:dealId
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]

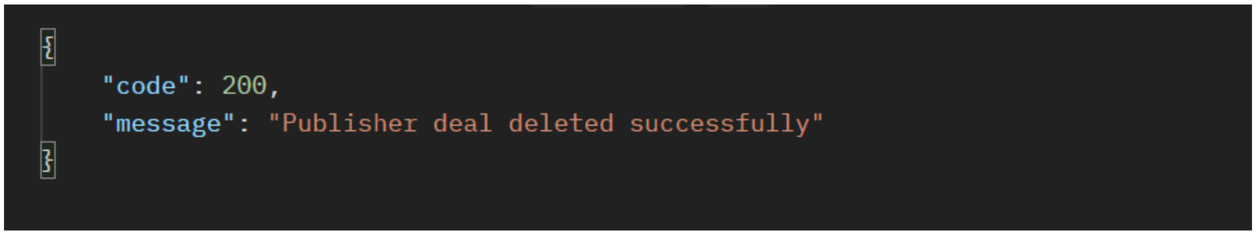
- Successful Response



```
{
    "code": 200,
    "message": "Publisher deal gotten successfully",
    "deal": {…
    }
}
```

7.4.5 Delete a publisher deal – delete a publisher deal by deal id for a specific publisher

- router: /api/v1/publisher/:publisherId/deal/:dealId
- method: DELETE
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response



```
{
    "code": 200,
    "message": "Publisher deal deleted successfully"
}
```

7.5 Report APIs

These APIs send an email to a specific publisher by publisher id along with a csv file included analytic data until current date

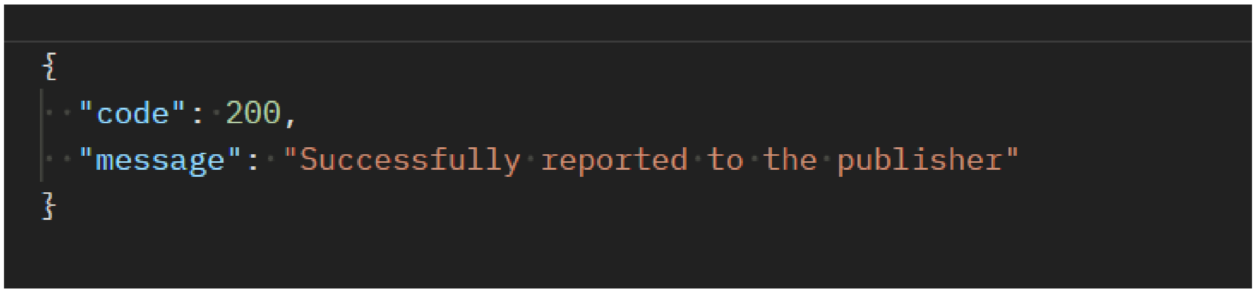
1) **Router1: /api/v1/report/:publisherId**

   This sends a report to a publisher for all publisher endpoints

2) **Router2: /api/v1/report/:publisherId/:endpointId**

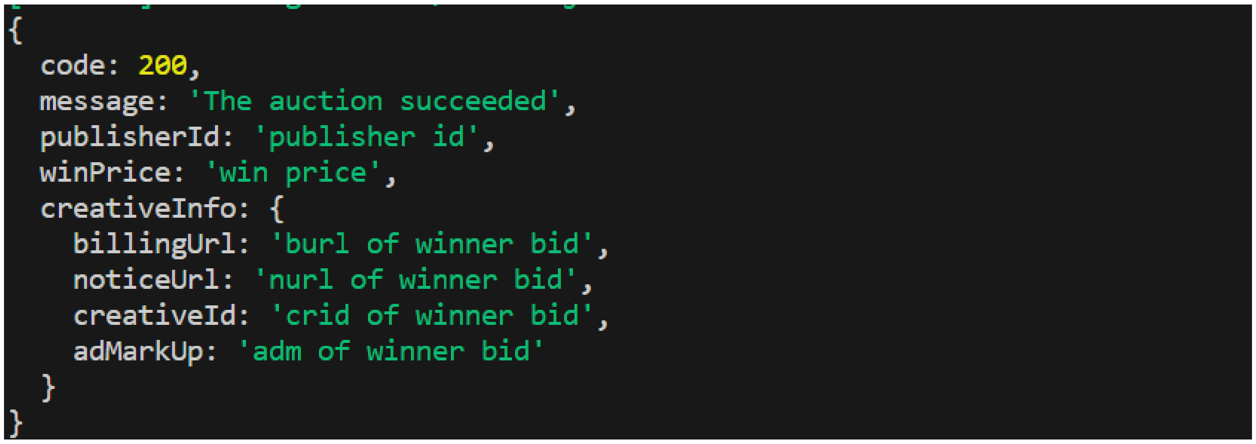
   This sends a report to a publisher for a publisher endpoint

- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- Successful Response


```
{
  "code": 200,
  "message": "Successfully reported to the publisher"
}
```

7.6 Auction API – initiate an auction and retrieve auction result

- router: /api/v1/auction
- method: GET
- Headers: [{ “x-api-key”: “Security key of the publisher or Admin access key”}]
- query: {
  “publisher_id”: publisherId,
  “device_id”: deviceId,
  “ad_unit_id”: 1
  }
- Successful Response


```
{
  code: 200,
  message: 'The auction succeeded',
  publisherId: 'publisher id',
  winPrice: 'win price',
  creativeInfo: {
    billingUrl: 'burl of winner bid',
    noticeUrl: 'nurl of winner bid',
    creativeId: 'crid of winner bid',
    adMarkUp: 'adm of winner bid'
  }
}
```

8. High Traffic Testing using Apache bench

8.1 Environment Setup

8.1.1 On Ubuntu

1. Update package database

   # apt-get update

2. Install apache2 utils package to get access to Apache Bench

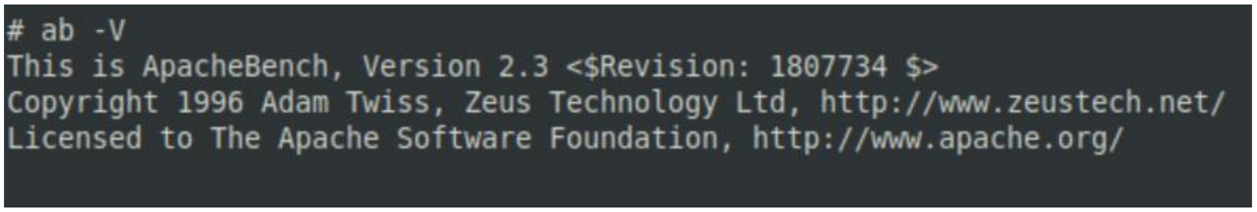
   # apt-get install apache2-utils

   Source: https://bobcares.com/blog/apache-benchmark-install-ubuntu

3. Verifying Apache Bench Installation

   # ab –V

   The command above produced on output as shown below:



```
# ab -V
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
```

8.1.2 On Windows

- Install using choco command

  chocho install apache-httpd

  Source: https://community.chocolatey.org/packages/apache-httpd

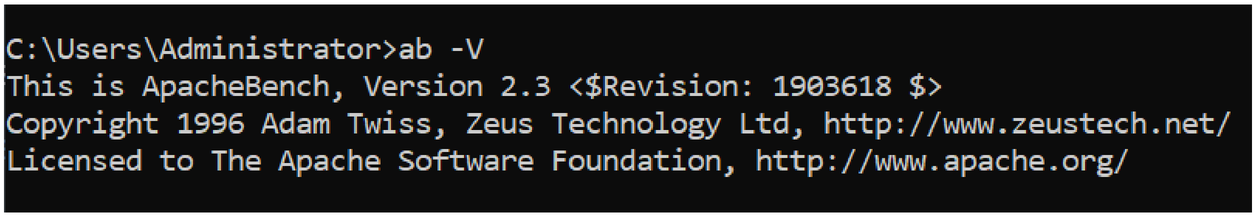
  ** how can use choco command on windows*

  Run the following command

  @"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

- Verifying Apache Bench Installation

  ab –V



```
C:\Users\Administrator>ab -V
This is ApacheBench, Version 2.3 <$Revision: 1903618 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
```

8.2 Test using ab command

ab -n 3000 -c 10 -H "x-api-key: Admin access key or security key of the publisher" "auction url"

eg **:** ab –n 3000 -c 10 -H "x-api-key: 3yT7jN9sBdR5fQgP2mW6uE4zA8cV1xX0" "http://localhost:80/api/v1/auction?publisher_id=bc837b49-ebe1-4679-b76a-724bdcb85ef5&&device_id=26cdd8d6-d84b-4374-b972-f6b1c799e4bc&&ad_unit_id=1"

This is used to perform a load test on our system with high traffic condition.

- ab: This is the Apache Bechmark tool used to perform the load testing

- -n 3000: This option specifies the total number of requests to perform.

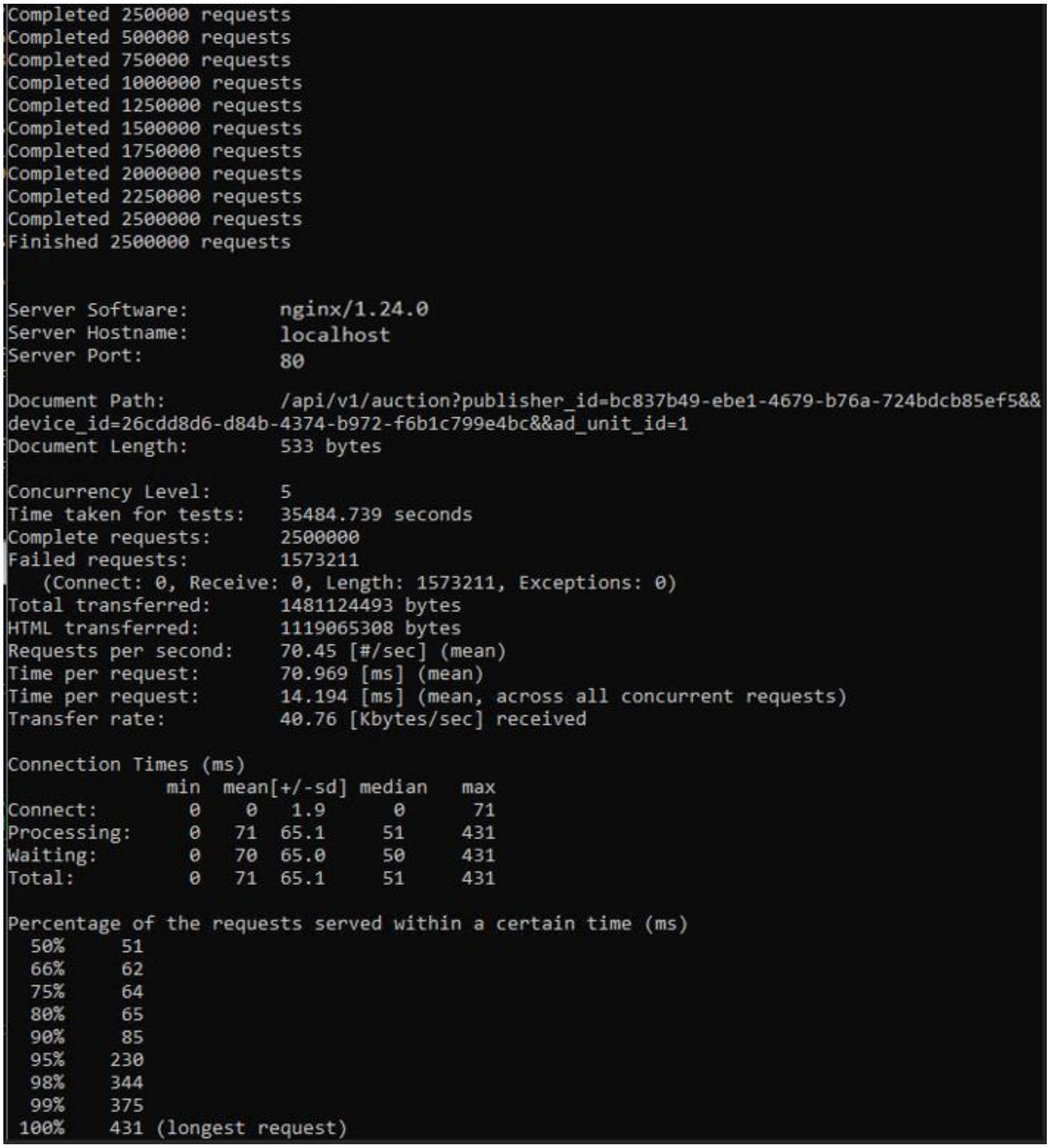
- -c 10: This option specifies the number of multiple requests to perform at a time (i.e. currently level). Here, it is set to 10, meaning 10 requests will be sent simultaneously.

- -H “x-api-key: 3yT7jN9sBdR5fQgP2mW6uE4zA8cV1xX0”: This option adds a custom header to authenticate the request.

- http://localhost:80/api/v1/auction?publisher_id=bc837b49-ebe1-4679-b76a-724bdcb85ef5&&device_id=26cdd8d6-d84b-4374-b972-f6b1c799e4bc&&ad_unit_id=1

  This is the URL of the endpoint being tested.

8.3 Testing result



```
Completed 250000 requests
Completed 500000 requests
Completed 750000 requests
Completed 1000000 requests
Completed 1250000 requests
Completed 1500000 requests
Completed 1750000 requests
Completed 2000000 requests
Completed 2250000 requests
Completed 2500000 requests
Finished 2500000 requests


Server Software:        nginx/1.24.0
Server Hostname:        localhost
Server Port:            80

Document Path:          /api/v1/auction?publisher_id=bc837b49-ebe1-4679-b76a-724bdcb85ef5&&
device_id=26cdd8d6-d84b-4374-b972-f6b1c799e4bc&&ad_unit_id=1
Document Length:        533 bytes

Concurrency Level:      5
Time taken for tests:   35484.739 seconds
Complete requests:      2500000
Failed requests:        1573211
   (Connect: 0, Receive: 0, Length: 1573211, Exceptions: 0)
Total transferred:      1481124493 bytes
HTML transferred:       1119065308 bytes
Requests per second:    70.45 [#/sec] (mean)
Time per request:       70.969 [ms] (mean)
Time per request:       14.194 [ms] (mean, across all concurrent requests)
Transfer rate:          40.76 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   1.9      0      71
Processing:     0   71  65.1     51     431
Waiting:        0   70  65.0     50     431
Total:          0   71  65.1     51     431

Percentage of the requests served within a certain time (ms)
  50%     51
  66%     62
  75%     64
  80%     65
  90%     85
  95%    230
  98%    344
  99%    375
 100%    431 (longest request)
```

Summary of Results

- Total Requests: 2 500 000
- Concurrency Level: 5 which sends 5 requests to the server concurrently.
- Time Taken for Tests: 35484.739 seconds equals around 10hrs
- Complete Requests: 2 500 000
- Failed Requests: 1 573 211

Length Errors: 1 573 211 (This doesn’t mean server error, just length error related to responses' length. don’t need to worry about this)

- Total Transferred: 1481124493 bytes
- HTML Transferred: 1119065308 bytes
- Requests per Second: 70.45 [#/sec] (mean)
- Time per Request: 70.969[ms] (mean)
- Time per Request (across all concurrent requests): 14.194[ms]
- Transfer Rate: 40.76 [Kbytes/sec] received

Percentage of Requests Served within a Certain Time (ms):

- 50%: 51ms
- 66%: 62ms
- 75%: 64ms
- 80%: 65ms
- 90%: 85ms
- 95%: 230ms
- 99%: 375ms
- 100%: 431ms (longest request)