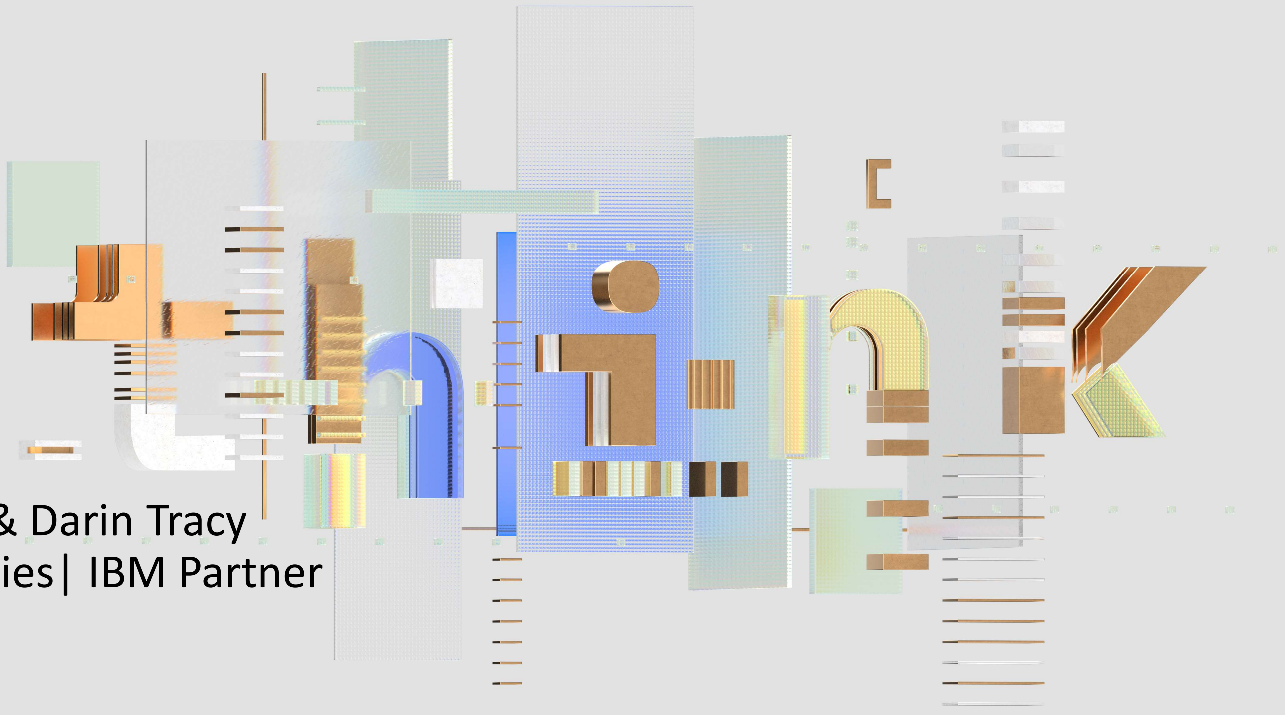
Developers – Get Up and Running with IBM Informix with Docker

think 2019



Brian Hughes & Darin Tracy
HCL Technologies| IBM Partner

Notices and disclaimers

Notices and disclaimers continued

Contents

Up & Running with Informix Database

Using the IBM Informix Docker image you can have a database system up and available in minutes.

Github Project based on docker-compose

Docker Compose

This repo contains the files to start and run the Sandbox project

https://github.com/informix/compose-developer-sandbox

```
$git clone https://github.com/informix/compose-developer-sandbox
```

Start The Sandbox

This will start a **server** docker container and a **client** docker container

The first time you run the docker-compose it will download the images from dockerhub

docker-compose

This command will run a docker container, setup connectivity ports, accept the license agreement and name the container '**server**'.

This command will also run a docker container, with client drivers and code for demo purposes. The container is named **'client'.**

```
$ cd compose-developer-sandbox
$ docker-compose up -d
```

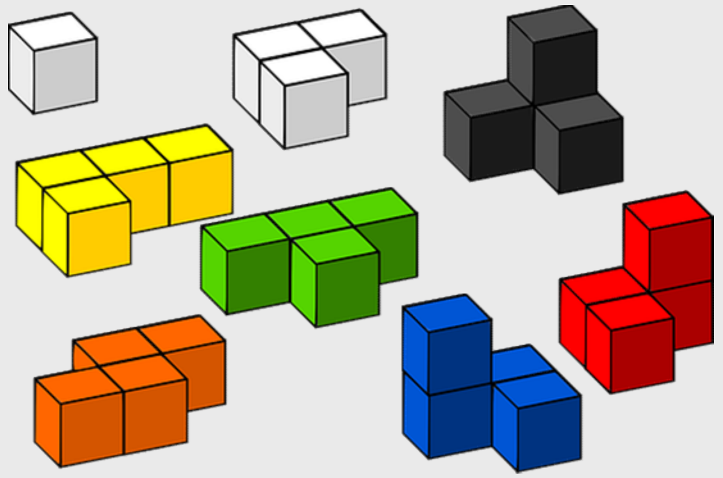
Up & Running with Informix Application Development

Using the IBM Informix Developer Sandbox Docker image you can have a development system up and running in minutes.

Building blocks

Client Sandbox includes,

- Java 1.8
- Informix JDBC
- Python 2.7
- NodeJS 8.10.0
- Informix Python Driver
- Informix NodeJS Driver
- Informix ODBC Driver



Information on the run options used

The –td starts the container as a daemon. You can then start a shell with the docker exec command and specify the name of the container to attach to.

The –name server, will name this container ‘**client**’

The –p options map the ports within the container to the host system. Port 9001 is needed for a demo needing http access.

```
$docker exec –it client bash
```

Application Development Examples

Using the IBM Informix Docker image you can have a database system up and available in minutes.

Java Examples – Initial setup

Setup Steps

1. Login to the client Docker container
   - (docker exec –it client bash)

2. cd informix-db-examples/sql/java

3. ./gradlew clean jar

`https://github.com/informix/informix-db-examples/tree/master/java`

Java Example – Create database

Run the following to create a database for use with the examples.

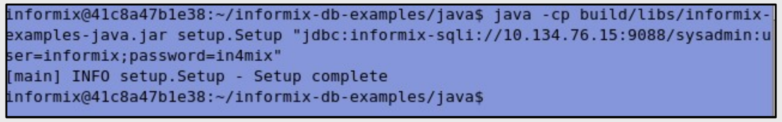
```
$java -cp build/libs/informix-examples-java.jar setup.Setup
"jdbc:informix-
sqli://server:9088/sysadmin:user=informix;password=in4mix"
```

- This example will create a database named **banktest**. This can be used for the other examples. And will be used for the Demo program referred to later on.
- **FYI**: informix-db-examples/sql/java/ex_create.sh

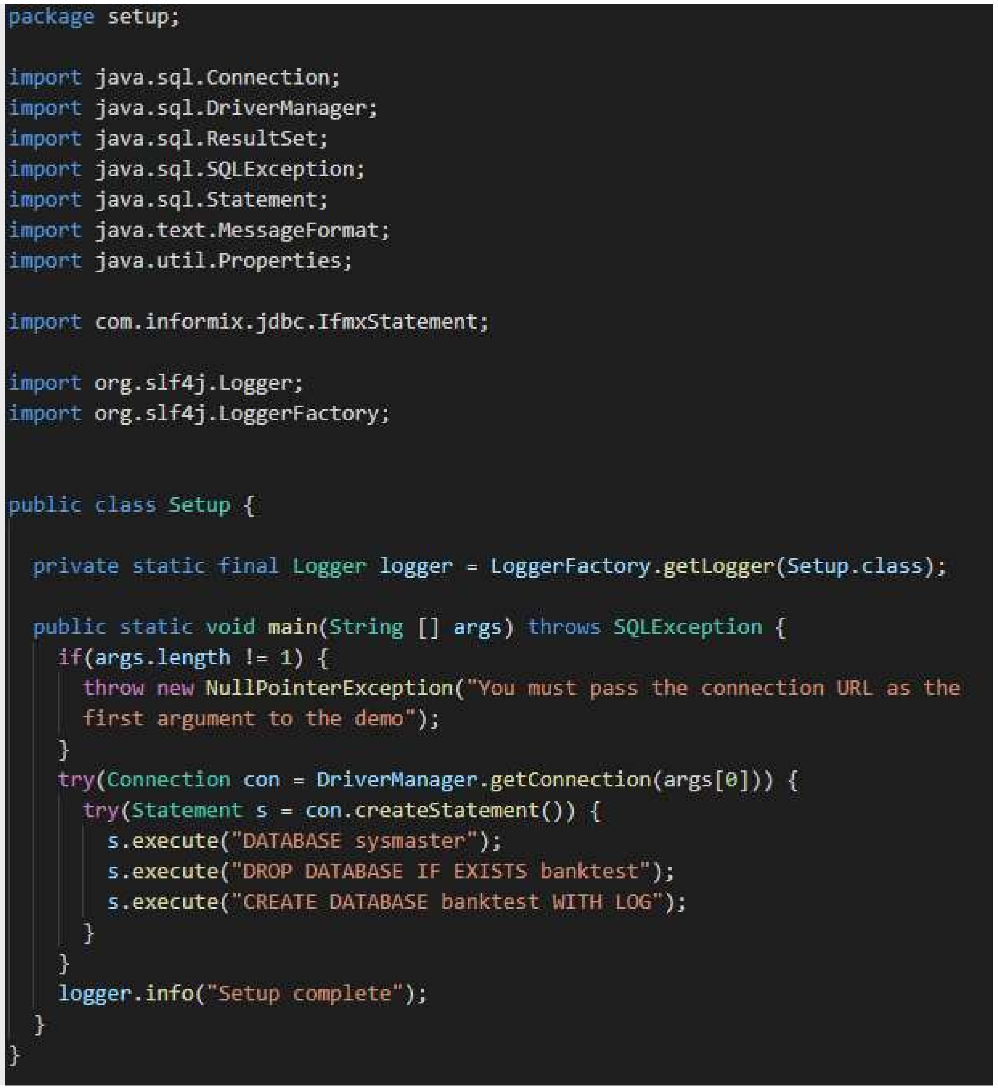
```
https://github.com/informix/informix-db-examples/tree/master/java
```

Java Example – Code - Create database

- src/main/java/setup/Setup.java



```
informix@41c8a47b1e38:~/informix-db-examples/java$ java -cp build/libs/informix-
examples-java.jar setup.Setup "jdbc:informix-sqli://10.134.76.15:9088/sysadmin:u
ser=informix;password=in4mix"
[main] INFO setup.Setup - Setup complete
informix@41c8a47b1e38:~/informix-db-examples/java$
```



```java
package setup;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.MessageFormat;
import java.util.Properties;

import com.informix.jdbc.IfmxStatement;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;


public class Setup {

  private static final Logger logger = LoggerFactory.getLogger(Setup.class);

  public static void main(String [] args) throws SQLException {
    if(args.length != 1) {
      throw new NullPointerException("You must pass the connection URL as the
      first argument to the demo");
    }
    try(Connection con = DriverManager.getConnection(args[0])) {
      try(Statement s = con.createStatement()) {
        s.execute("DATABASE sysmaster");
        s.execute("DROP DATABASE IF EXISTS banktest");
        s.execute("CREATE DATABASE banktest WITH LOG");
      }
    }
    logger.info("Setup complete");
  }
}
```

Java Example – BSON Example

Run the following to BSON example.

```
$java -cp build/libs/informix-examples-java.jar
dataTypes.jsonBson.BsonExample “jdbc:informix-
sqli://server:9088/banktest:user=informix;password=in4mix”
```

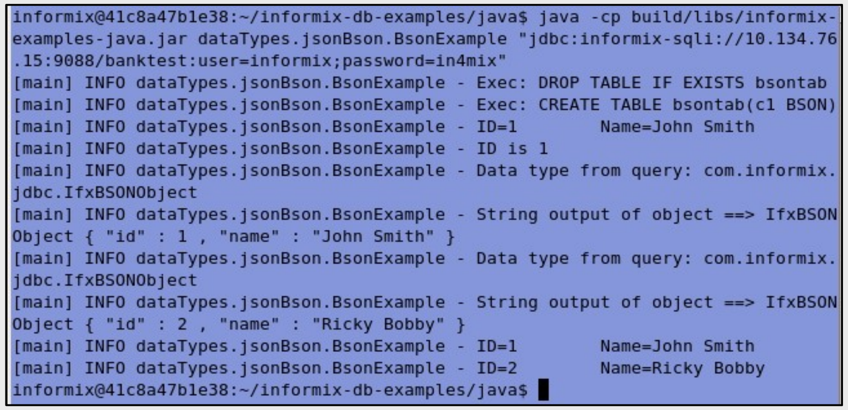
- This example will create a table named **bsonTab**. It creates a bson object and performs a couple inserts using different options.

- **FYI**: informix-db-examples/sql/java/ex_bson.sh

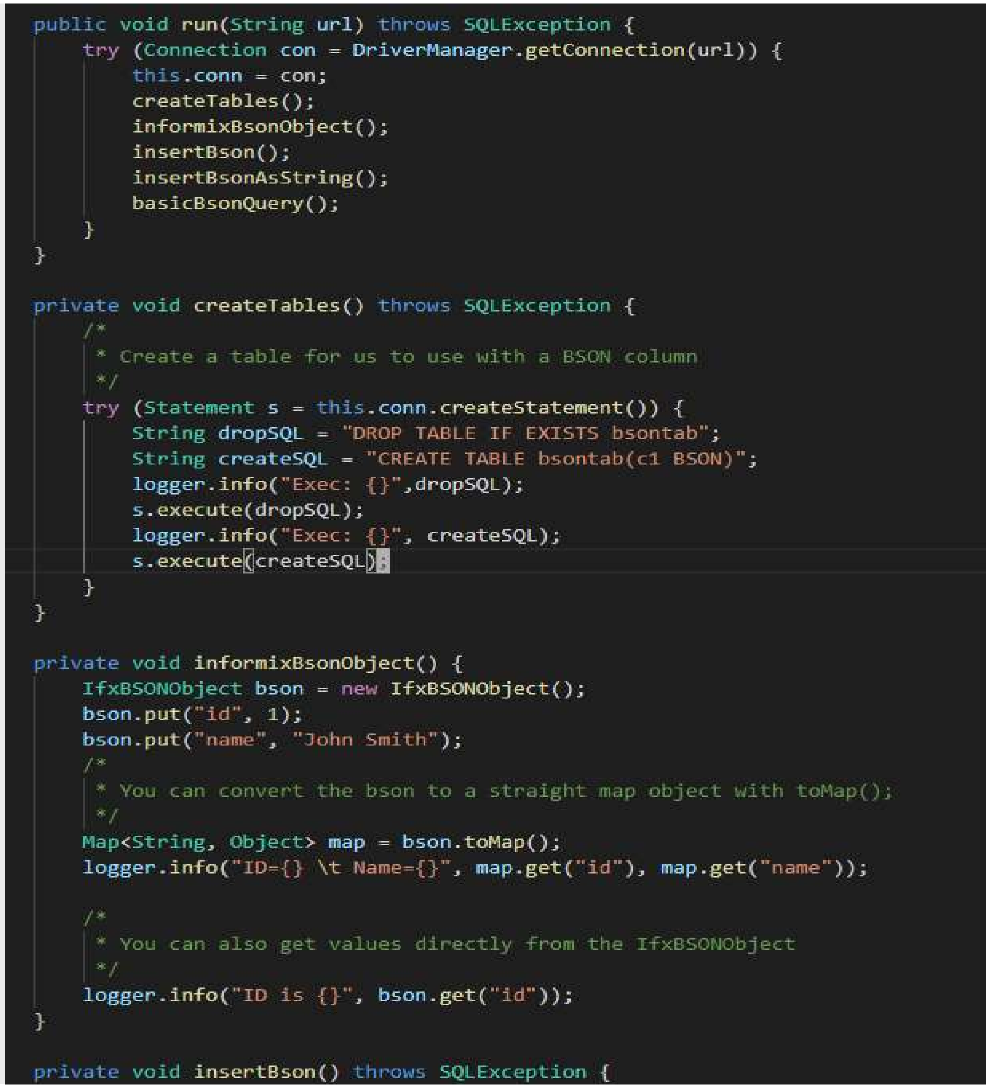
```
https://github.com/informix/informix-db-examples/tree/master/java
```

Java Example – Code – BSON example

- src/main/java/dataTypes/jsonBson/BsonExample.java



```
informix@41c8a47b1e38:~/informix-db-examples/java$ java -cp build/libs/informix-
examples-java.jar dataTypes.jsonBson.BsonExample "jdbc:informix-sqli://10.134.76
.15:9088/banktest:user=informix;password=in4mix"
[main] INFO dataTypes.jsonBson.BsonExample - Exec: DROP TABLE IF EXISTS bsontab
[main] INFO dataTypes.jsonBson.BsonExample - Exec: CREATE TABLE bsontab(c1 BSON)
[main] INFO dataTypes.jsonBson.BsonExample - ID=1        Name=John Smith
[main] INFO dataTypes.jsonBson.BsonExample - ID is 1
[main] INFO dataTypes.jsonBson.BsonExample - Data type from query: com.informix.
jdbc.IfxBSONObject
[main] INFO dataTypes.jsonBson.BsonExample - String output of object ==> IfxBSON
Object { "id" : 1 , "name" : "John Smith" }
[main] INFO dataTypes.jsonBson.BsonExample - Data type from query: com.informix.
jdbc.IfxBSONObject
[main] INFO dataTypes.jsonBson.BsonExample - String output of object ==> IfxBSON
Object { "id" : 2 , "name" : "Ricky Bobby" }
[main] INFO dataTypes.jsonBson.BsonExample - ID=1        Name=John Smith
[main] INFO dataTypes.jsonBson.BsonExample - ID=2        Name=Ricky Bobby
informix@41c8a47b1e38:~/informix-db-examples/java$ 
```



```java
public void run(String url) throws SQLException {
    try (Connection con = DriverManager.getConnection(url)) {
        this.conn = con;
        createTables();
        informixBsonObject();
        insertBson();
        insertBsonAsString();
        basicBsonQuery();
    }
}

private void createTables() throws SQLException {
    /*
     * Create a table for us to use with a BSON column
     */
    try (Statement s = this.conn.createStatement()) {
        String dropSQL = "DROP TABLE IF EXISTS bsontab";
        String createSQL = "CREATE TABLE bsontab(c1 BSON)";
        logger.info("Exec: {}",dropSQL);
        s.execute(dropSQL);
        logger.info("Exec: {}", createSQL);
        s.execute(createSQL);
    }
}

private void informixBsonObject() {
    IfxBSONObject bson = new IfxBSONObject();
    bson.put("id", 1);
    bson.put("name", "John Smith");
    /*
     * You can convert the bson to a straight map object with toMap();
     */
    Map<String, Object> map = bson.toMap();
    logger.info("ID={} \t Name={}", map.get("id"), map.get("name"));

    /*
     * You can also get values directly from the IfxBSONObject
     */
    logger.info("ID is {}", bson.get("id"));
}

private void insertBson() throws SQLException {
```

Java Example – SmartTrigger example

Run the following to SmartTrigger example.

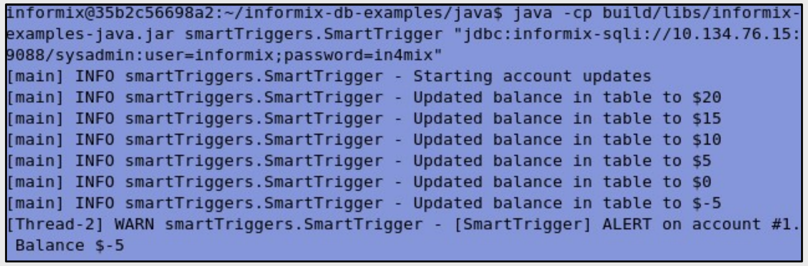
```
$java -cp build/libs/informix-examples-java.jar
smartTriggers.SmartTrigger "jdbc:informix-
sqli://server:9088/sysadmin:user=informix;password=in4mix"
```

- This example will create a smart Trigger on an account table. It will periodically update the account balance, decreasing the total. When the account goes below 0 an Alert will trigger on the update.
  - Important to use connect to **sysadmin** for this example
- **FYI**: informix-db-examples/sql/java/ex_smarttrigger.sh

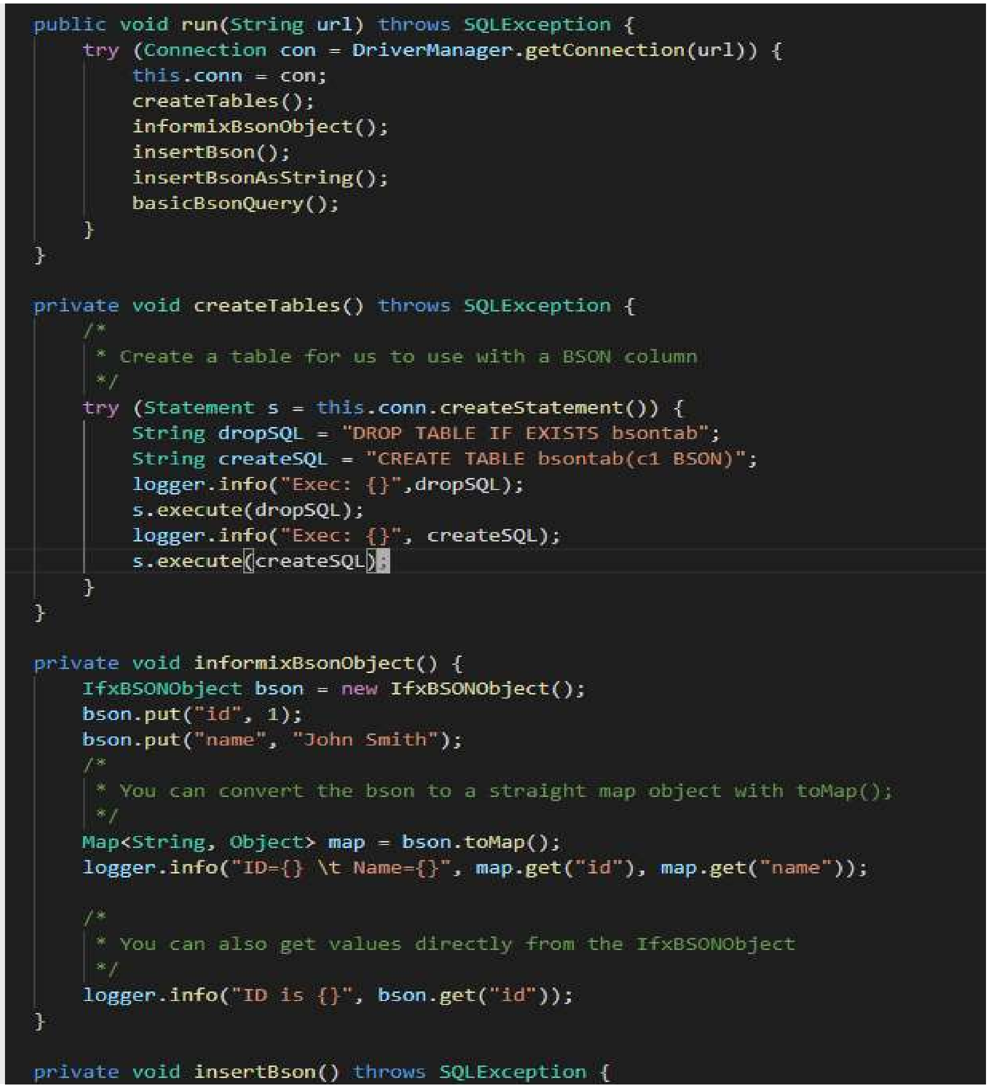
```
https://github.com/informix/informix-db-examples/tree/master/java
```

Java Example – Code – SmartTrigger example

- src/main/java/smartTriggers/SmartTrigger.java



```
informix@35b2c56698a2:~/informix-db-examples/java$ java -cp build/libs/informix-
examples-java.jar smartTriggers.SmartTrigger "jdbc:informix-sqli://10.134.76.15:
9088/sysadmin:user=informix;password=in4mix"
[main] INFO smartTriggers.SmartTrigger - Starting account updates
[main] INFO smartTriggers.SmartTrigger - Updated balance in table to $20
[main] INFO smartTriggers.SmartTrigger - Updated balance in table to $15
[main] INFO smartTriggers.SmartTrigger - Updated balance in table to $10
[main] INFO smartTriggers.SmartTrigger - Updated balance in table to $5
[main] INFO smartTriggers.SmartTrigger - Updated balance in table to $0
[main] INFO smartTriggers.SmartTrigger - Updated balance in table to $-5
[Thread-2] WARN smartTriggers.SmartTrigger - [SmartTrigger] ALERT on account #1.
 Balance $-5
```



```java
public void run(String url) throws SQLException {
    try (Connection con = DriverManager.getConnection(url)) {
        this.conn = con;
        createTables();
        informixBsonObject();
        insertBson();
        insertBsonAsString();
        basicBsonQuery();
    }
}

private void createTables() throws SQLException {
    /*
     * Create a table for us to use with a BSON column
     */
    try (Statement s = this.conn.createStatement()) {
        String dropSQL = "DROP TABLE IF EXISTS bsontab";
        String createSQL = "CREATE TABLE bsontab(c1 BSON)";
        logger.info("Exec: {}",dropSQL);
        s.execute(dropSQL);
        logger.info("Exec: {}", createSQL);
        s.execute(createSQL);
    }
}

private void informixBsonObject() {
    IfxBSONObject bson = new IfxBSONObject();
    bson.put("id", 1);
    bson.put("name", "John Smith");
    /*
     * You can convert the bson to a straight map object with toMap();
     */
    Map<String, Object> map = bson.toMap();
    logger.info("ID={} \t Name={}", map.get("id"), map.get("name"));

    /*
     * You can also get values directly from the IfxBSONObject
     */
    logger.info("ID is {}", bson.get("id"));
}

private void insertBson() throws SQLException {
```

Python Examples – Initial setup

Initial setup

1. Login to the client Docker container
   – (docker exec –it client bash)

2. cd informix-db-examples

3. Review connections.json (used by python programs)

4. cd informix-db-examples/sql/python

```
{
  “host” : “server”,
  “port” : “9088”,
  “user” : “informix”,
  “password” : “in4mix”,
  “database” : “banktest”,
  “server” : “informix”
}
```

`https://github.com/informix/informix-db-examples/tree/master/python`

Python Examples – basicQuery example

Run basicQuery.py

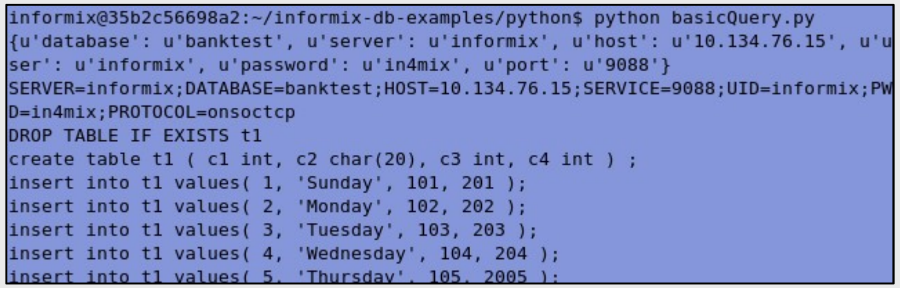
```
$python basicQuery.py
```

- This example will create a table named **t1**, insert some data and query the data.

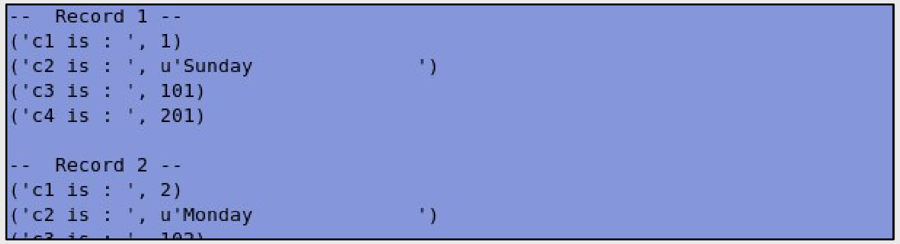
```
https://github.com/informix/informix-db-examples/tree/master/python
```

Python Example – Code – basicQuery example

- basicQuery.py

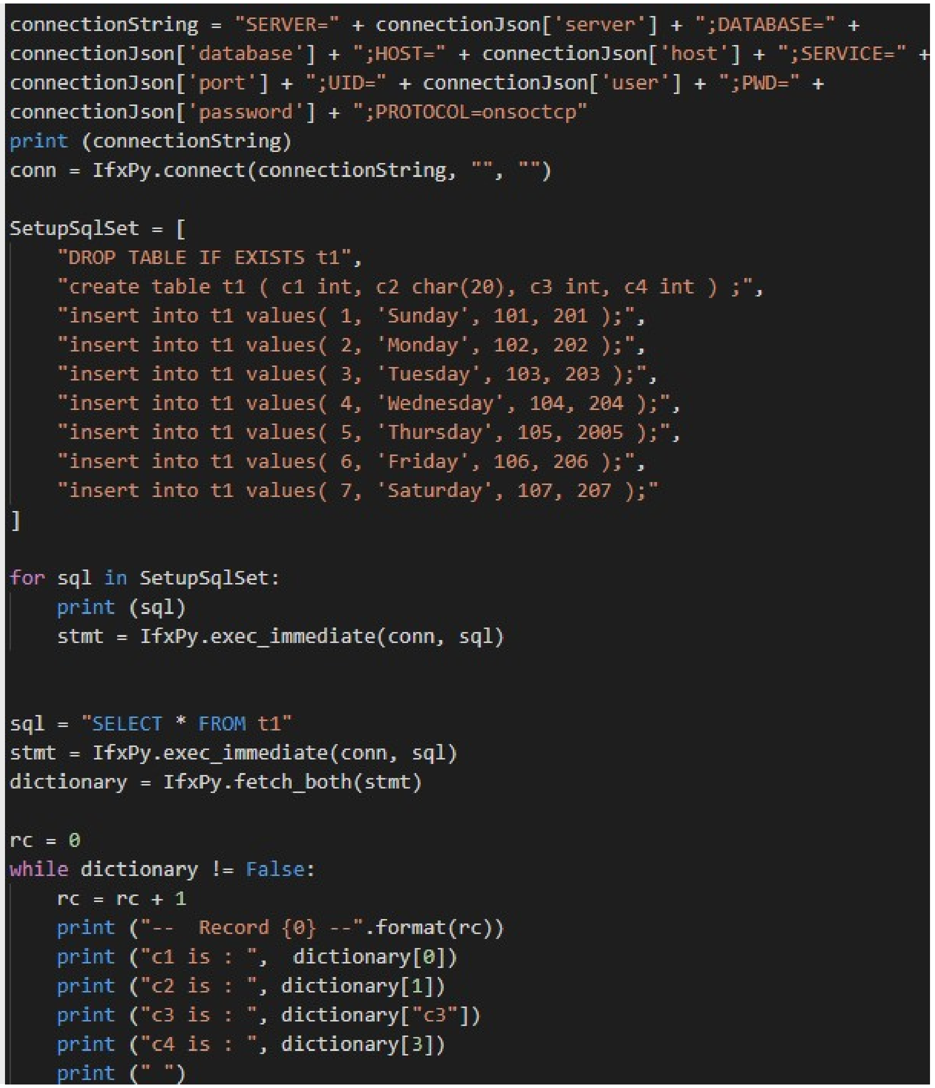


```
informix@35b2c56698a2:~/informix-db-examples/python$ python basicQuery.py
{u'database': u'banktest', u'server': u'informix', u'host': u'10.134.76.15', u'u
ser': u'informix', u'password': u'in4mix', u'port': u'9088'}
SERVER=informix;DATABASE=banktest;HOST=10.134.76.15;SERVICE=9088;UID=informix;PW
D=in4mix;PROTOCOL=onsoctcp
DROP TABLE IF EXISTS t1
create table t1 ( c1 int, c2 char(20), c3 int, c4 int ) ;
insert into t1 values( 1, 'Sunday', 101, 201 );
insert into t1 values( 2, 'Monday', 102, 202 );
insert into t1 values( 3, 'Tuesday', 103, 203 );
insert into t1 values( 4, 'Wednesday', 104, 204 );
insert into t1 values( 5, 'Thursday', 105, 2005 );
```



```
--  Record 1 --
('c1 is : ', 1)
('c2 is : ', u'Sunday              ')
('c3 is : ', 101)
('c4 is : ', 201)

--  Record 2 --
('c1 is : ', 2)
('c2 is : ', u'Monday              ')
```



```python
connectionString = "SERVER=" + connectionJson['server'] + ";DATABASE=" +
connectionJson['database'] + ";HOST=" + connectionJson['host'] + ";SERVICE=" +
connectionJson['port'] + ";UID=" + connectionJson['user'] + ";PWD=" +
connectionJson['password'] + ";PROTOCOL=onsoctcp"
print (connectionString)
conn = IfxPy.connect(connectionString, "", "")

SetupSqlSet = [
    "DROP TABLE IF EXISTS t1",
    "create table t1 ( c1 int, c2 char(20), c3 int, c4 int ) ;",
    "insert into t1 values( 1, 'Sunday', 101, 201 );",
    "insert into t1 values( 2, 'Monday', 102, 202 );",
    "insert into t1 values( 3, 'Tuesday', 103, 203 );",
    "insert into t1 values( 4, 'Wednesday', 104, 204 );",
    "insert into t1 values( 5, 'Thursday', 105, 2005 );",
    "insert into t1 values( 6, 'Friday', 106, 206 );",
    "insert into t1 values( 7, 'Saturday', 107, 207 );"
]

for sql in SetupSqlSet:
    print (sql)
    stmt = IfxPy.exec_immediate(conn, sql)


sql = "SELECT * FROM t1"
stmt = IfxPy.exec_immediate(conn, sql)
dictionary = IfxPy.fetch_both(stmt)

rc = 0
while dictionary != False:
    rc = rc + 1
    print ("--  Record {0} --".format(rc))
    print ("c1 is : ",  dictionary[0])
    print ("c2 is : ", dictionary[1])
    print ("c3 is : ", dictionary["c3"])
    print ("c4 is : ", dictionary[3])
    print (" ")
```

NodeJS Examples – Under construction

NodeJS Examples – Initial setup

1. Initial setup
   - Login to the client Docker container
     - (docker exec –it client bash)
   - cd informix-db-examples
   - Modify connections.json accordingly. Set the host in the connections.json to your IP address of the HOST where the docker containers are running
   - cd nodejs

```
{
  "host": "localhost",
  "port": "9088",
  "user": "informix",
  "password": "in4mix",
  "database": "banktest",
  "server": "informix"
}
```

https://github.com/informix/informix-db-examples/tree/master/nodejs

NodeJS Examples – basicQuery example

2. Run basicQuery.js

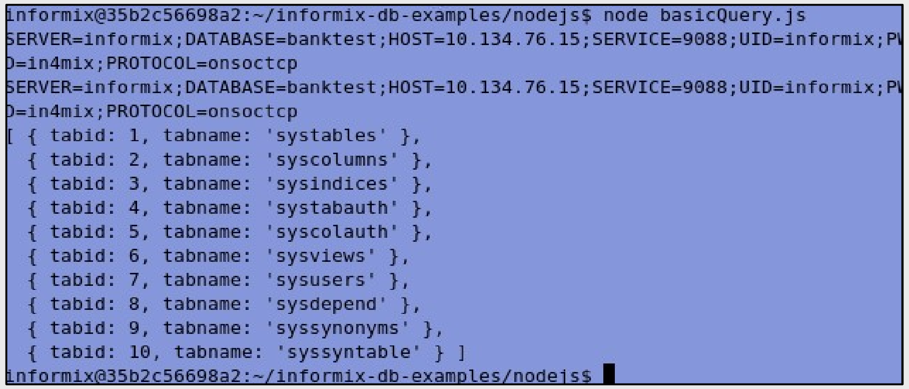
```
$node basicQuery.js
```

- This example will query the first 10 rows from systables.

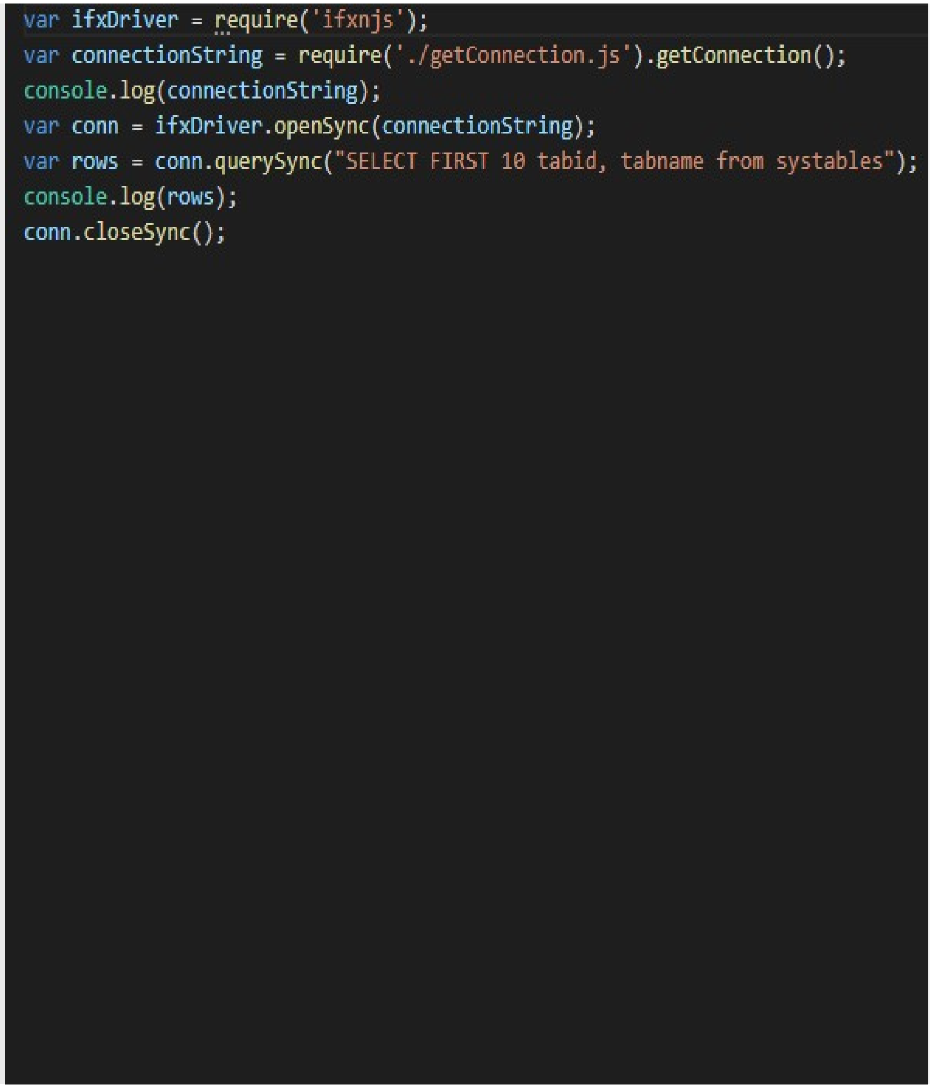
```
https://github.com/informix/informix-db-examples/tree/master/nodejs
```

NodeJS Example – Code – basicQuery example

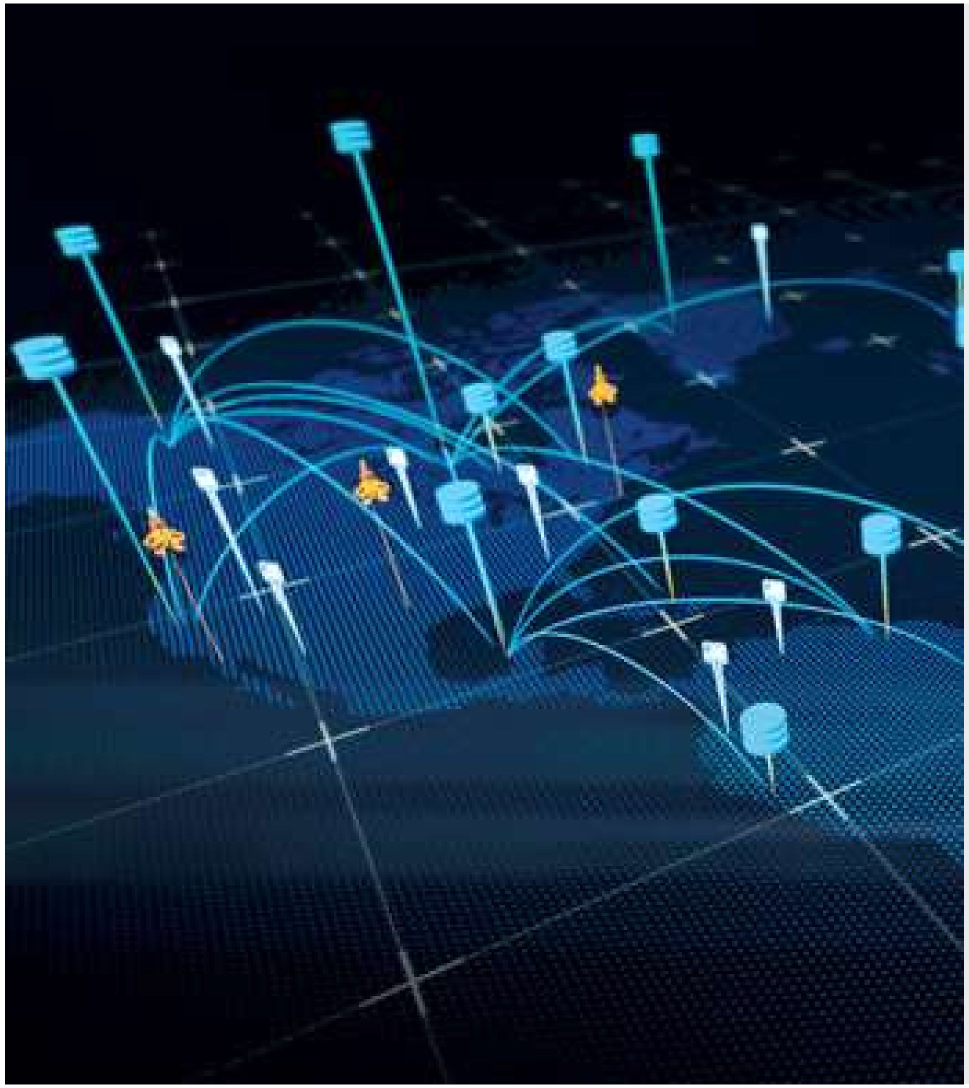
- basicQuery.js



```
informix@35b2c56698a2:~/informix-db-examples/nodejs$ node basicQuery.js
SERVER=informix;DATABASE=banktest;HOST=10.134.76.15;SERVICE=9088;UID=informix;PW
D=in4mix;PROTOCOL=onsoctcp
SERVER=informix;DATABASE=banktest;HOST=10.134.76.15;SERVICE=9088;UID=informix;PW
D=in4mix;PROTOCOL=onsoctcp
[ { tabid: 1, tabname: 'systables' },
  { tabid: 2, tabname: 'syscolumns' },
  { tabid: 3, tabname: 'sysindices' },
  { tabid: 4, tabname: 'systabauth' },
  { tabid: 5, tabname: 'syscolauth' },
  { tabid: 6, tabname: 'sysviews' },
  { tabid: 7, tabname: 'sysusers' },
  { tabid: 8, tabname: 'sysdepend' },
  { tabid: 9, tabname: 'syssynonyms' },
  { tabid: 10, tabname: 'syssyntable' } ]
informix@35b2c56698a2:~/informix-db-examples/nodejs$
```



```
var ifxDriver = require('ifxnjs');
var connectionString = require('./getConnection.js').getConnection();
console.log(connectionString);
var conn = ifxDriver.openSync(connectionString);
var rows = conn.querySync("SELECT FIRST 10 tabid, tabname from systables");
console.log(rows);
conn.closeSync();
```



Demo

Using the IBM Informix Docker image you can have a database system up and available in minutes.

GeoSpatial – SmartTrigger Demo

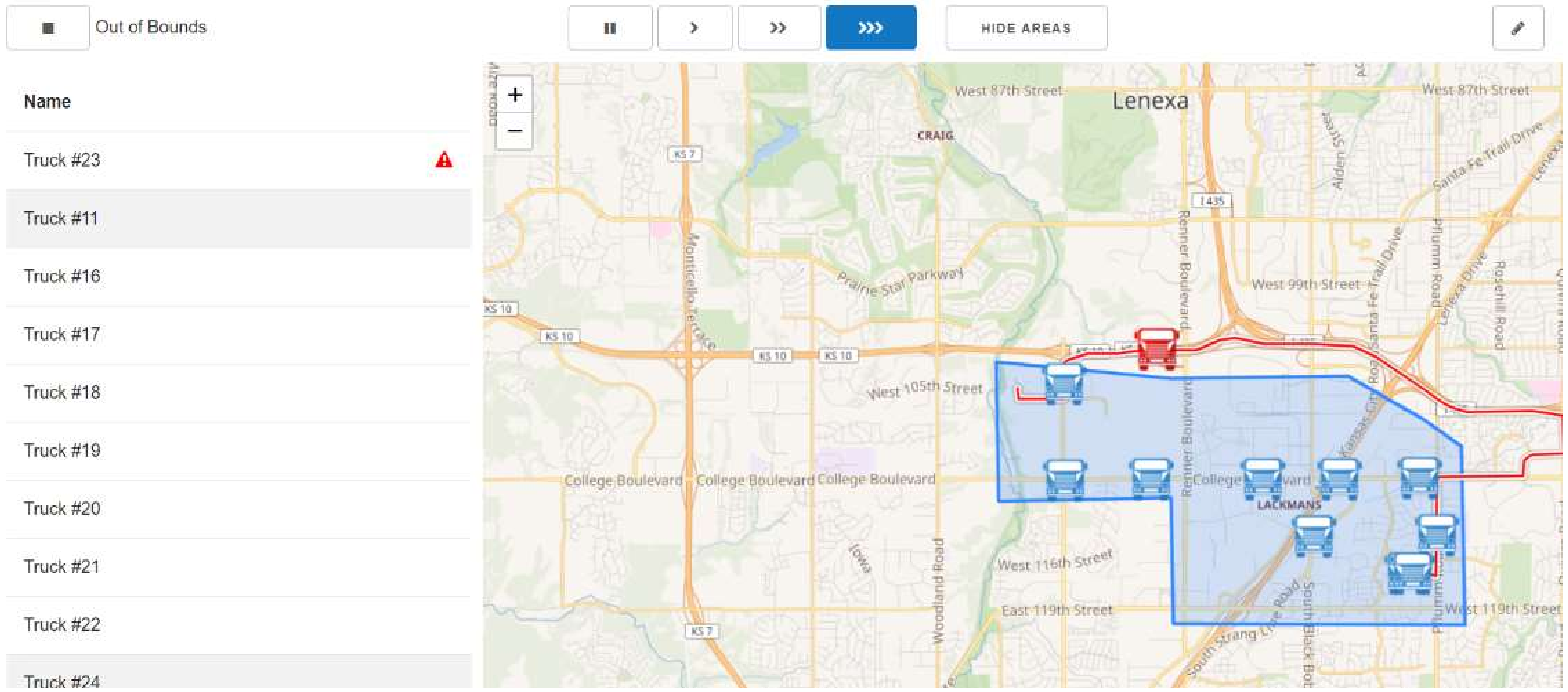
1. Initial setup
   - Login to the client Docker container
     – (docker exec –it client bash)
   - cd demo
   - Modify config.json accordingly. Set the host in the connections.json to your IP address of the HOST where the docker containers are running

GeoSpatial – SmartTrigger Demo

2. Run the demo
   - Java –jar geotriggers.jar

   - This Demo simulates fleet management. A random number of trucks are tracked on their delivery routes.
   - When a truck goes outside of its geo-fenced area an Alert will be triggered.
   - This Demo uses Smart Triggers for the alerting, and Geospatial data for the tracking of the truck's routes.

GeoSpatial – SmartTrigger Demo



Out of Bounds

Name
Truck #23 ⚠
Truck #11
Truck #16
Truck #17
Truck #18
Truck #19
Truck #20
Truck #21
Truck #22
Truck #24