

Home PUBLIC Questions Tags Users Companies COLLECTIVES Explore Collectives Stack Overflow for Teams

Unable to connect to Postgres DB due to the authentication type 10 is not supported

Ask Question

Asked 2 years, 3 months ago Modified 3 months ago Viewed 132k times

I have recently tried my java on Postgres. Installed it on local (PostgreSQL 13.0). Created a maven project and used Spring Data JPA, works just fine. Whereas when I tried using Gradle project, I am not able to connect to the DB and keep getting the following error.

```
org.postgresql.util.PSQLException: The authentication type 10 is not supported. Check that you have configured the pg_hba.conf file to include the client's IP address or subnet, and that it is using an authentication scheme supported by the driver. at org.postgresql.core.v3.ConnectionFactoryImpl.doAuthentication(ConnectionFactoryImpl.java:614) ~[postgresql-42.1.4.jar:42.1.4] at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:222) ~[postgresql-42.1.4.jar:42.1.4] at org.postgresql.core.ConnectionFactory.openConnection(ConnectionFactory.java:49) ~[postgresql-42.1.4.jar:42.1.4] at org.postgresql.jdbc.PgConnection.(PgConnection.java:194) ~[postgresql-42.1.4.jar:42.1.4] at org.postgresql.Driver.makeConnection(Driver.java:450) ~[postgresql-42.1.4.jar:42.1.4] at org.postgresql.Driver.connect(Driver.java:252) ~[postgresql-42.1.4.jar:42.1.4] at java.sql.DriverManager.getConnection(Unknown Source) [na:1.8.0_261] at org.postgresql.ds.common.BaseDataSource.getConnection(BaseDataSource.java:94) ~[postgresql-42.1.4.jar:42.1.4] at org.postgresql.ds.common.BaseDataSource.getConnection(BaseDataSource.java:79) ~[postgresql-42.1.4.jar:42.1.4]
```

I tried using JDBCTemplate as well. Doesn't work.

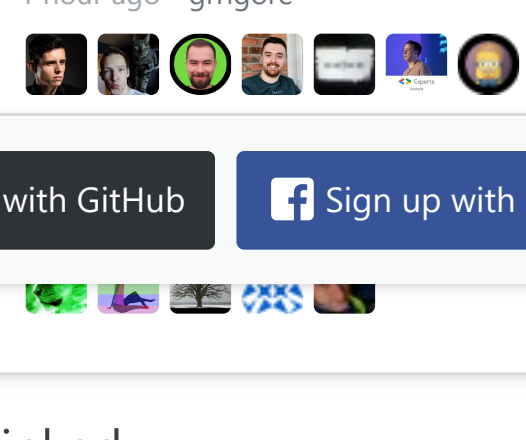
The Overflow Blog

- Comparing tag trends with our Most Loved programming languages
- The less JavaScript, the better (Ep. 532)

Featured on Meta

- Accessibility Update: Colors
- Introducing a new close reason specifically for non-English questions
- We're bringing advertisements for technology courses to Stack Overflow
- 2022 a year in moderation
- Temporary policy: ChatGPT is banned

46 people chatting



Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email

Sign up with Google

Sign up with GitHub

Sign up with Facebook

Used the deprecated Lib of - Doesn't Work either.

Please Suggest me a solution for this problem.

My code and Config:

```
@Autowired
ApplicationContext context;

public void setDataSource() {
    //Getting Bean by Class
    DriverManagerDataSource dataSource = context.getBean(DriverManagerDataSource.class);
    this.dataSource = dataSource;
    this.jdbcTemplateObject = new JdbcTemplate(this.dataSource);
}

@Override
public Customer create(Customer customer) {
    setDataSource();
    String sql = "insert into CustomerOrder (customerType, customerPayment) values (?, ?)";
    jdbcTemplateObject.update(sql, customerOrder.getCustomerOrderType(), customerOrder.getCustomerOrderPayment());

    KeyHolder holder = new GeneratedKeyHolder();
    jdbcTemplateObject.update(new PreparedStatementCreator() {
        @Override
        public PreparedStatement createPreparedStatement(Connection connection) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS);
            ps.setString(1, customer.getType());
            return ps;
        }
    }, holder);

    long customerId = holder.getKey().longValue();
    customer.setCustomerId(customerId);
    return customer;
}

}
```

dependencies

```
implementation('org.springframework.boot:spring-boot-starter-web')
compile('org.springframework.boot:spring-boot-devtools')
compile(group: 'org.postgresql', name: 'postgresql', version: '42.1.4')
compile('org.springframework.boot:spring-boot-starter-jdbc')
compile('org.springframework:spring-jdbc:5.2.5.RELEASE')
```

password_encryption is set like this:

```
postgres=# show password_encryption;
password_encryption
-----
scram-sha-256
(1 row)
```

java postgresql spring-boot gradle postgresql-13

Share Improve this question Follow

asked Apr 7, 2020 at 18:04 asked Oct 5, 2020 at 14:01

asked Oct 5, 2020 at 14:43

3 What does `show password_encryption` give you? If that is set to `SCRAM-SHA-256` then you will need a newer JDBC driver - `a_horse_with_no_name` Oct 5, 2020 at 14:43

Yes, `show password_encryption`; `password_encryption` ----- `scram-sha-256` (1 row)

In my case, the only way to solve this problem in a Docker image was to downgrade Postgres version, as per my colleague's advice. So I created a new container with PostgreSQL v13.2 and it solved the issue. It is a test server so this solution works for me. - `gipe` Jan 20, 2022 at 12:50

Add a comment

17 Answers

Sorted by: Highest score (default)

He назвали ответ? Задайте вопрос на Stack Overflow на русском.

I solved similar issue by applying below steps in PostgreSQL Version 13:

1. Change `password_encryption` to `md5` in `postgresql.conf`

Windows: C:\Program Files\PostgreSQL\13\data\postgresql.conf
GNU/Linux: /etc/postgresql/13/main/postgresql.conf

```
#authentication_timeout = 1min
password_encryption = md5
#db_user_namespace = off
```

2. Change `scram-sha-256` to `md5` in `pg_hba.conf`

Windows: C:\Program Files\PostgreSQL\13\data\pg_hba.conf
GNU/Linux: /etc/postgresql/13/main/pg_hba.conf

```
host all all 0.0.0.0/0 md5

# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
host all all 0.0.0.0/0 md5
```

3. Change Password (this restore password in md5 format).

Example: `ALTER ROLE postgres WITH PASSWORD 'root';`

4. Make sure you set `listen_addresses = '*'` in `postgresql.conf` if you are working non production environment.

Share Improve this answer Follow

edited May 22, 2022 at 15:16 answered Oct 13, 2020 at 13:37

answered Oct 13, 2020 at 13:37

6 This downgrades the security of the driver. You might want to consider upgrading your JDBC driver instead.

1 Password are still created with `scram-sha-256` encryption after the described changes... Any idea why? Thanks - `noamyy` Jan 17, 2021 at 8:11

2 Don't forget the `;` at the end of the ALTER ROLE statement. Took me a while to realize. - `Tobias Grunwald` Mar 1, 2021 at 7:12

1 For me, I had to `ALTER USER` instead as the last step, i.e. `ALTER USER postgres WITH PASSWORD 'root';` - `Sam Amiri` May 2, 2021 at 4:20

4 I strongly recommend people to follow `this answer` instead, which is to upgrade the PostgreSQL JDBC driver to 42.2.0 or higher. - `Mark Rotteveel` Jul 29, 2021 at 11:11

Show 1 more comment

According to the [wiki](#), the supported JDBC driver for `SCRAM-SHA-256` encryption is 42.2.0 or above. In my case, the driver was 41.1.1. Change it to 42.2.0 or above. That fixed it for me.

(Maven, `pom.xml`):

```
<dependency>
<groupId>org.postgresql</groupId>
<artifactId>postgresql</artifactId>
<version>42.2.0</version>
</dependency>
```

Share Improve this answer Follow

edited Apr 7, 2021 at 14:12 answered Jan 17, 2021 at 8:40

answered Jan 17, 2021 at 8:40

Strange, I have the same problem with version 42.2.8... - `Mathieu` Apr 19, 2021 at 10:39

4 Worked! Upgraded 42.1.1 to 42.2.4 - `Gishu` Oct 10, 2021 at 8:23

There's a bug in PostgreSQL jdbc driver 42.2.0, better use higher version. Check out: [youtrack.jetbrains.com/issue/DBE-5833](#) - `Dmitry Kibitsky` May 24, 2022 at 14:40

1 for Java 1.7 works updating to 42.2.25-jre7 - `Abel Matos` Jun 9, 2022 at 22:16

Add a comment

Get your `pg_hba.conf` File in the Directory `C:\Program Files\PostgreSQL\13\data\pg_hba.conf`

And Simply Change `scram-sha-256` under Column Method to trust.

```
# TYPE DATABASE USER ADDRESS METHOD
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 trust
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
host all all 0.0.0.0/0 md5
```

It worked for me!

Share Improve this answer Follow

answered Jun 9, 2021 at 7:23

answered Jun 9, 2021 at 7:23

4 This solution worked for me as well. +1 - `Jayapriya Athrean` Oct 22, 2021 at 5:33

This solution worked for me on Kali 2022.2 trying to connect locally with Armitage - `Steve Weaver` Dec 28, 2022 at 19:57

Add a comment

By setting `password_encryption` to `scram-sha-256` (which is the default value in v13) you also get `scram-sha-256` authentication, even if you have `md5` in `pg_hba.conf`.

Now you are using an old JDBC driver version on the client side that does not support that authentication method, even though PostgreSQL introduced it in v10, three years ago.

You should upgrade your JDBC driver. An alternative would be to set `password_encryption` back to `md5`, but then you'll have to reset all passwords and live with lower security.

Share Improve this answer Follow

edited Oct 5, 2020 at 15:33 answered Oct 5, 2020 at 15:27

answered Oct 5, 2020 at 15:27

Just curious, how am I able to connect using a maven project and not a gradle project. - `Tushar Banne` Oct 5, 2020 at 15:29

No, that is independent of what you are running above the JDBC driver. - `Laurenz Albe` Oct 5, 2020 at 15:36

The driver that is used in my maven project is same as above (org.postgresql.Driver). That is why I used the same in my gradle project. - `Tushar Banne` Oct 6, 2020 at 7:54

Upgrading the driver did the job for me, as well. - `Simon K.` Oct 16, 2020 at 14:23

2 This should be the accepted answer, as changing values in `pg_hba.conf` is just a workaround and the real solution is to upgrade the JDBC driver version to one that supports `scram-sha-256`. Eg. version 8.3.0.084, JDBC, doesn't support it, but version 42.2.139 does. - `Baku` Mar 29, 2021 at 13:43

Show 1 more comment

<xml version="1.0" encoding="UTF-8">

```
<groupId>org.postgresql</groupId>
<artifactId>postgresql</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
<java.version>11</java.version>
<maven.compiler.target>${java.version}</maven.compiler.target>
<maven.compiler.source>${java.version}</maven.compiler.source>
</properties>

<dependencies>
<dependency>
<groupId>org.postgresql</groupId>
<artifactId>postgresql</artifactId>
<version>42.2.18</version>
</dependency>
</dependencies>
```

you have to check your maven dependency if you are using postgresql 9.1+ then your dependency should be like above

to know about maven dependency refer this link [How do you add PostgreSQL Driver as a dependency in Maven?](#)

Share Improve this answer Follow

answered Dec 27, 2020 at 13:20

answered Dec 27, 2020 at 13:20

Add a comment

Change METHOD to "trust" in `pg_hba.conf`

Share Improve this answer Follow

answered Feb 18, 2021 at 8:38

answered Feb 18, 2021 at 8:38

Not secure, and definitely not suitable for ongoing use in a real system, but for my purposes (temporary testing without needing to reconfigure encryption or user passwords) it was a convenient workaround! - `Thomas W` Mar 31, 2021 at 3:28

Add a comment

In case you are struggling to get this working in Docker:

1. Firstly: run the container with `docker run -e POSTGRES_HOST_AUTH_METHOD=md5` (doc)

docker run -e POSTGRES_HOST_AUTH_METHOD=md5 -e POSTGRES_PASSWORD=doesntmatter -p 5432:5432

2. Secondly: allow `md5` encryption as discussed in other answers:

docker exec -ti -u postgres CONTAINERNAME bash -c "echo 'password_encryption=md5' >> /var/lib/

3. Thirdly: restart the container

docker restart CONTAINERNAME

4. Fourthly: you need to recreate the postgres password in `md5` format

docker exec -ti -u postgres CONTAINERNAME psql

alter role postgres with password 'THE-NEW-PASSWORD';

* please be aware `scram-sha-256` is much better than `md5` (doc)

Share Improve this answer Follow

edited Sep 11, 2022 at 14:27 answered Mar 25, 2022 at 18:15

answered Mar 25, 2022 at 18:15

Add a comment

Use these :

1. wget <https://jdbc.postgresql.org/download/postgresql-42.2.24.jar>

2. Copy it to your hive library

3. sudo mv postgresql-42.2.24.jar /opt/hive/lib/postgresql-42.2.24.jar

Share Improve this answer Follow

answered Oct 23, 2021 at 11:07

answered Oct 23, 2021 at 11:07

best solution to keep security - `kyle england` Dec 6, 2022 at 7:36

Add a comment

For me, updating the postgres library helped fixing this.

Share Improve this answer Follow

answered May 7, 2022 at 19:26

answered May 7, 2022 at 19:26

Add a comment

working fine with version 12.6... just downgrade the PostgreSQL

Share Improve this answer Follow

answered May 11, 2021 at 12:41

answered May 11, 2021 at 12:41

Add a comment

You might need to check the version of Postgres you are running. Might need to update spring version if the version is being pointed through spring parent. In my case: since current postgres is at v13. Modified spring parent version: it was on 1.4; made it to match to 2.14. Finally update maven dependency and re-run the application. This fixed the issue.

Share Improve this answer Follow

edited May 11, 2021 at 23:26 answered May 11, 2021 at 23:12

answered May 11, 2021 at 23:12

Add a comment

Suggestions:

1. Current JDBC driver will help (e.g. postgresql-42.3.6.jar)

2. Copy it to the /jars folder under your spark install directory (I'm assuming a single machine here in this example)

3. Python - install "findspark" to make pyspark importable as a regular library

4. Here is an example I hope will help someone:

```
import findspark
findspark.init()

from pyspark.sql import SparkSession

sparkClassPath = "C:/spark/spark-3.0.3-bin-hadoop2.7/jars"

spark = SparkSession \
    .builder() \
    .config("spark.driver.extraClassPath", sparkClassPath) \
    .getOrCreate()

df = spark.read \
    .format("jdbc") \
    .option("url", "jdbc:postgresql://(yourhostname):5432/(yourdbname)") \
    .option("driver", "org.postgresql.Driver") \
    .option("database", "(yourtablename)") \
    .option("user", "(yourusername)") \
    .option("password", "(yourkeytopassword)") \
    .load()

5. Install pgadmin if you have not already done so.
Try it via Docker
```

Share Improve this answer Follow

edited Jun 6, 2022 at 13:42 answered Jun 5, 2022 at 23:56

answered Jun 5, 2022 at 23:56

Add a comment

You need to download postgresql-jar and then move it into `.../jre/lib/ext` folder. It worked for me

Share Improve this answer Follow

answered Oct 11, 2022 at 14:28

answered Oct 11, 2022 at 14:28

Add a comment

1 Your answer could be improved with additional supporting information. Please add to add further details, such as citations or documentation, so that others can confirm that your answer is correct. You can find more information on how to write good answers [in the help center](#). - `Community Bot` Oct 15, 2022 at 13:12

Add a comment

Use latest maven dependency for Postgres in pom.xml

-1 Share Improve this answer Follow

answered Dec 2, 2020 at 5:36

answered Dec 2, 2020 at 5:36

Add a comment

1 Please edit your answer and add some explanation. - `Tichenaz FreePalestine` Dec 2, 2020 at 6:27

Add a comment

Changing trust for ipv4 local connect worked for me.

-1 Share Improve this answer Follow

answered Mar 25, 2022 at 6:50

answered Mar 25, 2022 at 6:50

Add a comment

I guess the solution to this problem is using version 9.6. It works just fine after changing the version.

-2 Share Improve this answer Follow

answered Oct 6, 2020 at 7:55

answered Oct 6, 2020 at 7:55

Add a comment

Open pg_hba.conf Set IPv4 local connections to trust

-3 Share Improve this answer Follow

answered Jul 2, 2022 at 18:55

answered Jul 2, 2022 at 18:55

Add a comment

Your Answer

Sign up or log in Post as a guest

Sign up using Google Sign up using Facebook Sign up using Email and Password

By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy

Not the answer you're looking for? Browse other questions tagged java postgresql spring-boot gradle postgresql-13 or ask your own question.