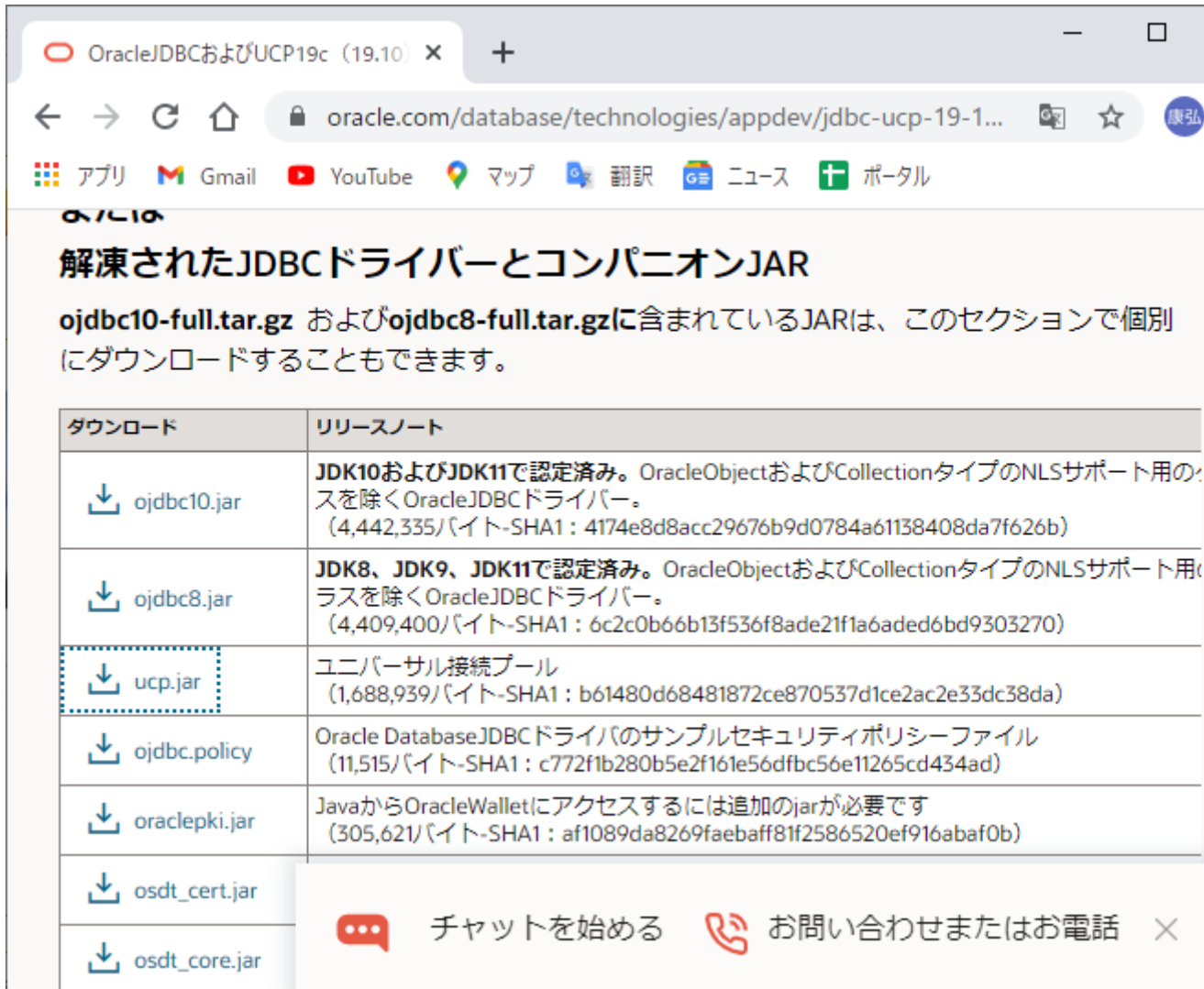






VSCodeでJava(thinドライバ)-Oracleへ接続

1. JDBCをインストール

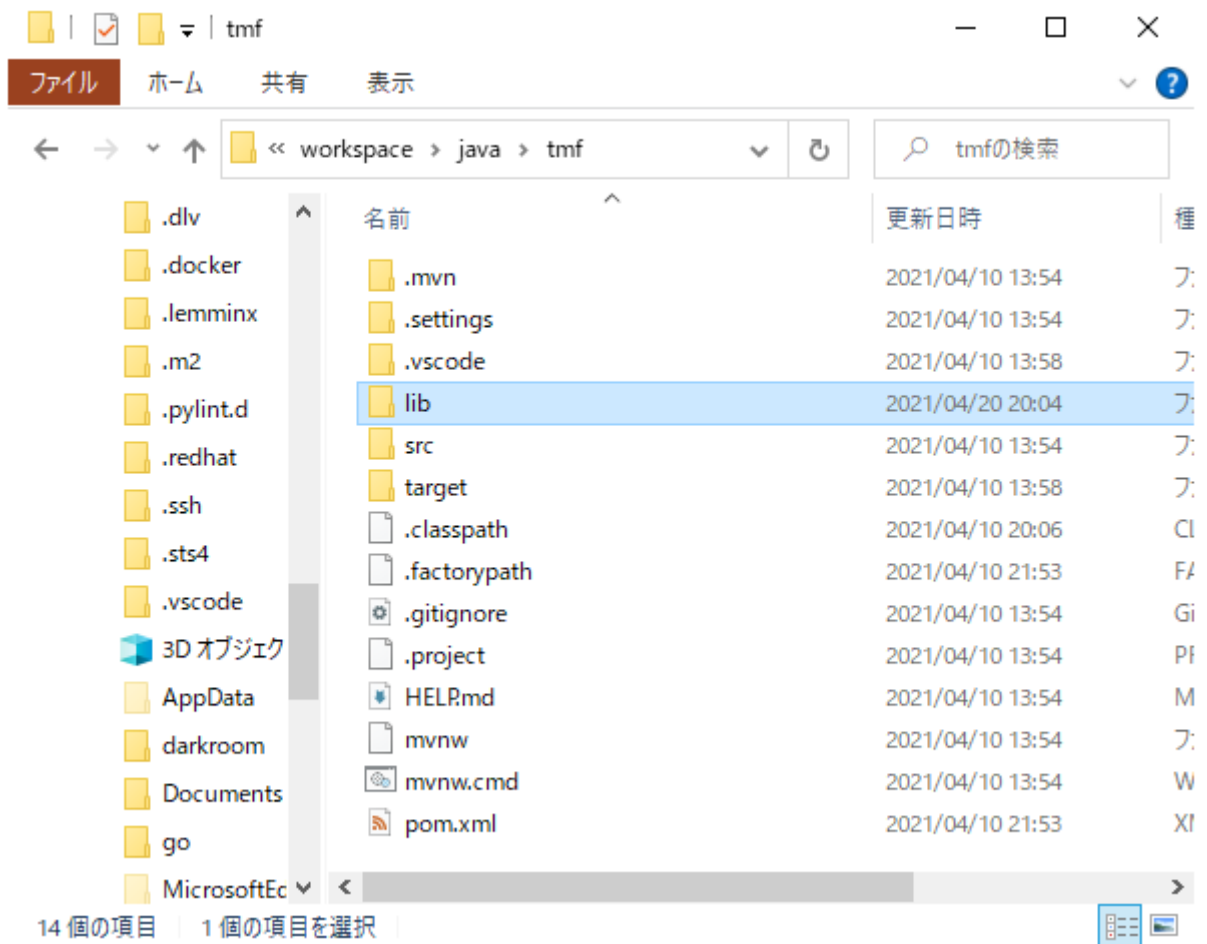
- 「<https://www.oracle.com/database/technologies/appdev/jdbc-ucp-19-10-c-downloads.html>」にアクセス。「ojdbc10.jar」「ucp.jar」をクリック。



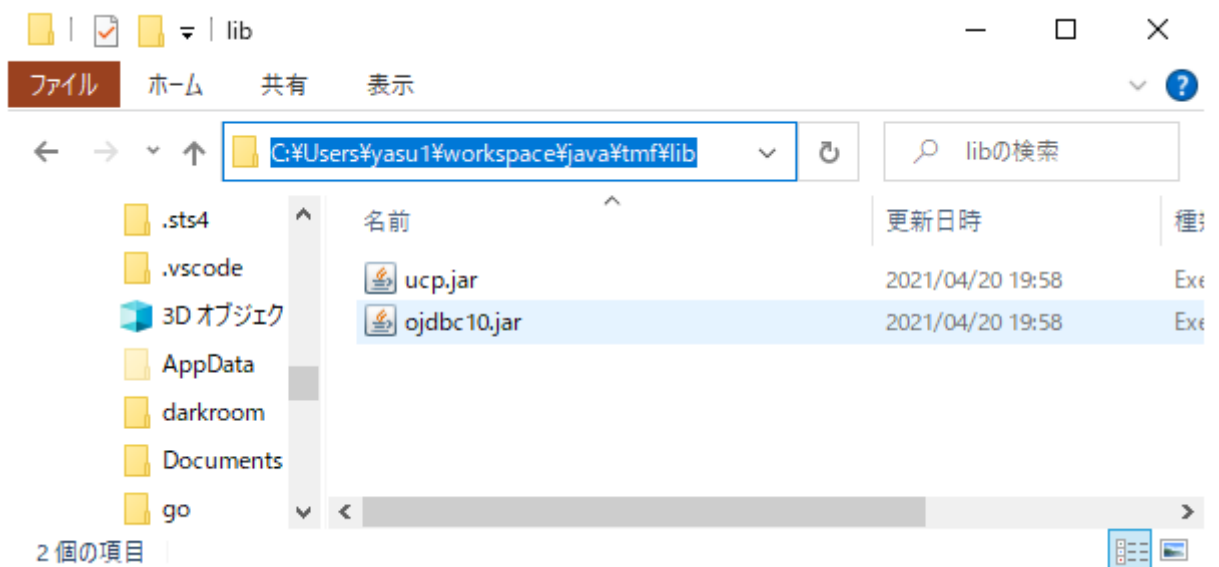
ダウンロード	リリースノート
 ojdbc10.jar	JDK10およびJDK11で認定済み。 OracleObjectおよびCollectionタイプのNLSサポート用のスを除くOracleJDBCドライバー。 (4,442,335/バイト-SHA1: 4174e8d8acc29676b9d0784a61138408da7f626b)
 ojdbc8.jar	JDK8、JDK9、JDK11で認定済み。 OracleObjectおよびCollectionタイプのNLSサポート用のラスを除くOracleJDBCドライバー。 (4,409,400/バイト-SHA1: 6c2c0b66b13f536f8ade21f1a6aded6bd9303270)
 ucp.jar	ユニバーサル接続プール (1,688,939/バイト-SHA1: b61480d68481872ce870537d1ce2ac2e33dc38da)
 ojdbc.policy	Oracle DatabaseJDBCドライバのサンプルセキュリティポリシーファイル (11,515/バイト-SHA1: c772f1b280b5e2f161e56dfbc56e11265cd434ad)
 oraclepki.jar	JavaからOracleWalletにアクセスするには追加のjarが必要です (305,621/バイト-SHA1: af1089da8269faebaff81f2586520ef916abaf0b)
 osdt_cert.jar	
 osdt_core.jar	

チャットを始める お問い合わせまたはお電話

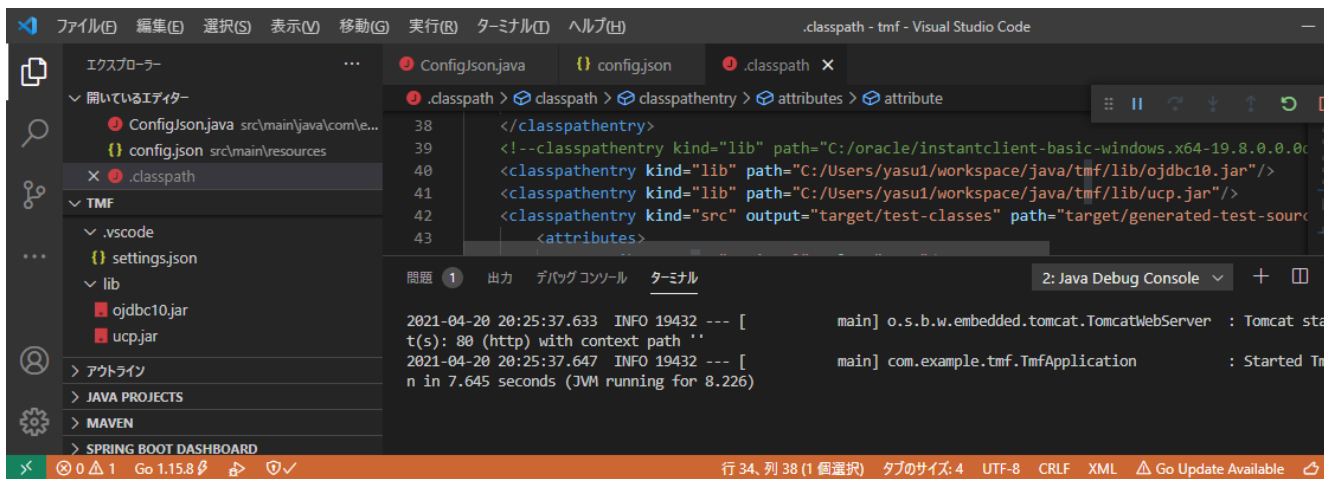
- 「C:\Users\yasu1\workspace\java\tmf」フォルダを作成。



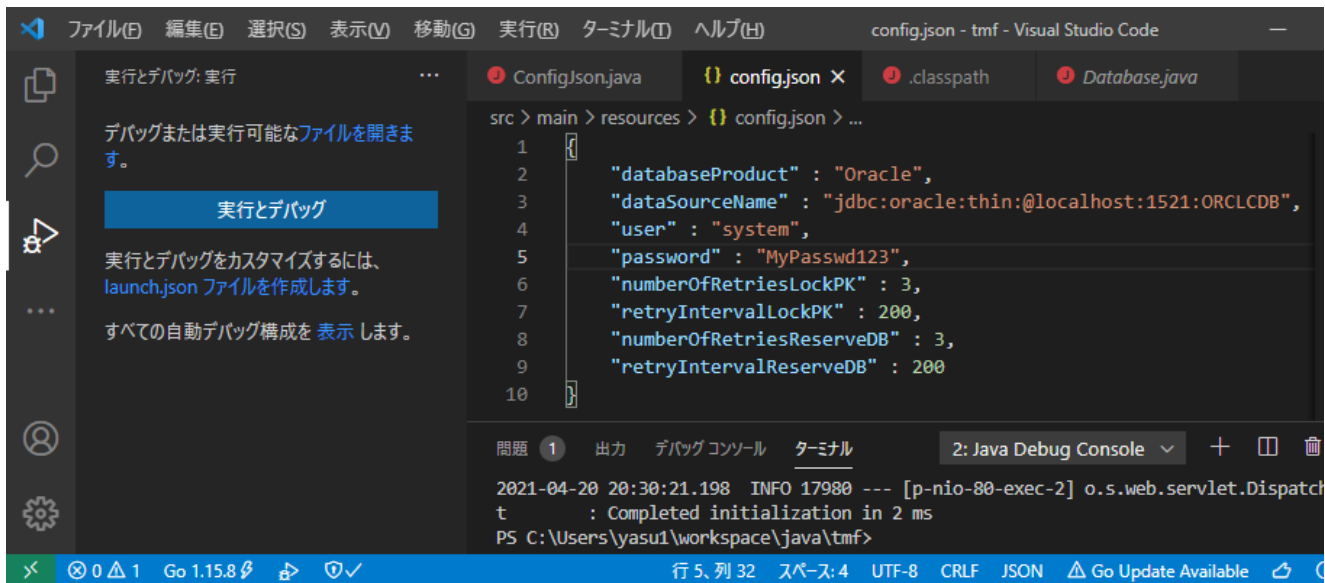
c. ダウンロードしたjarファイルを移動



d. 「C:\Users\yasu1\workspace\java\demo\classpath」の<classpath>の中に、「<classpathentry kind="lib" path="C:/Users/yasu1/workspace/java/tmf/lib/ojdbc10.jar"/><classpathentry kind="lib" path="C:/Users/yasu1/workspace/java/tmf/lib/ucp.jar"/>」を追記。

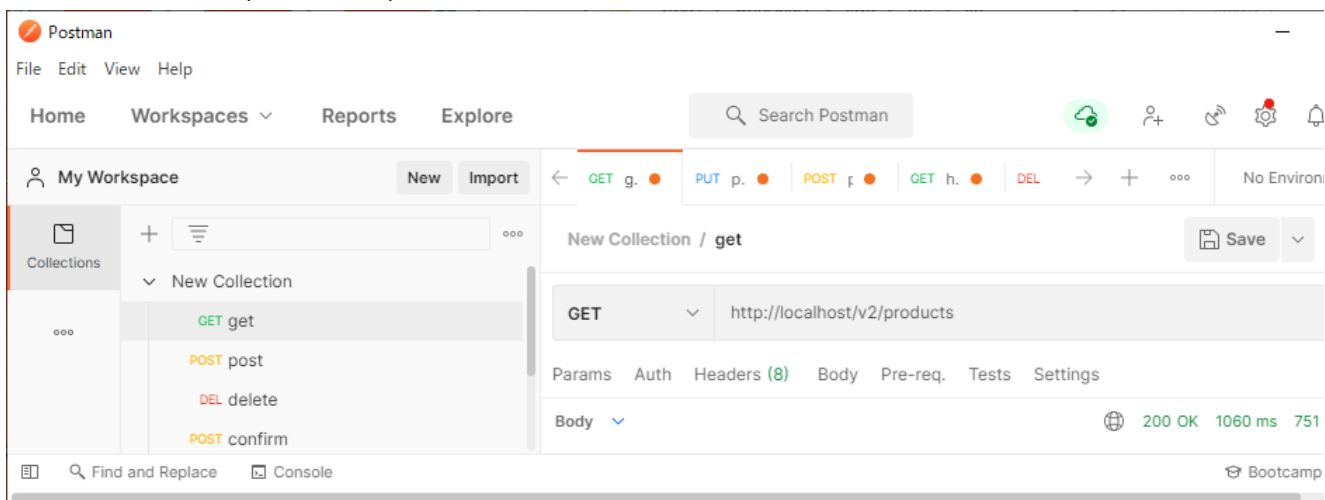


e. config.jsonのdataSourceNameに「jdbc:oracle:thin:@localhost:1521:ORCLCDB」

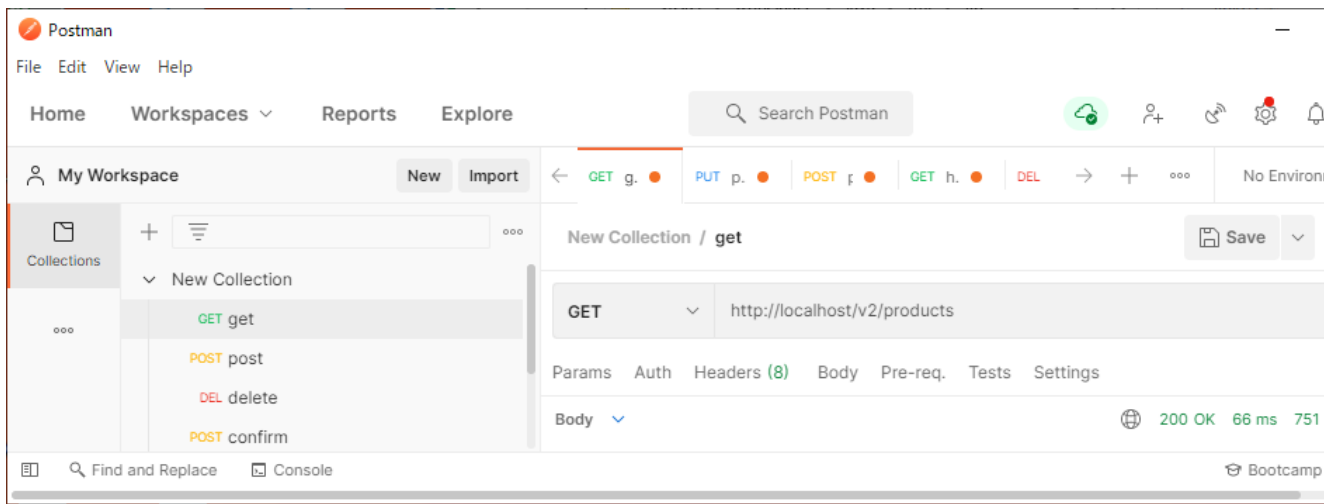


2. 速度比較

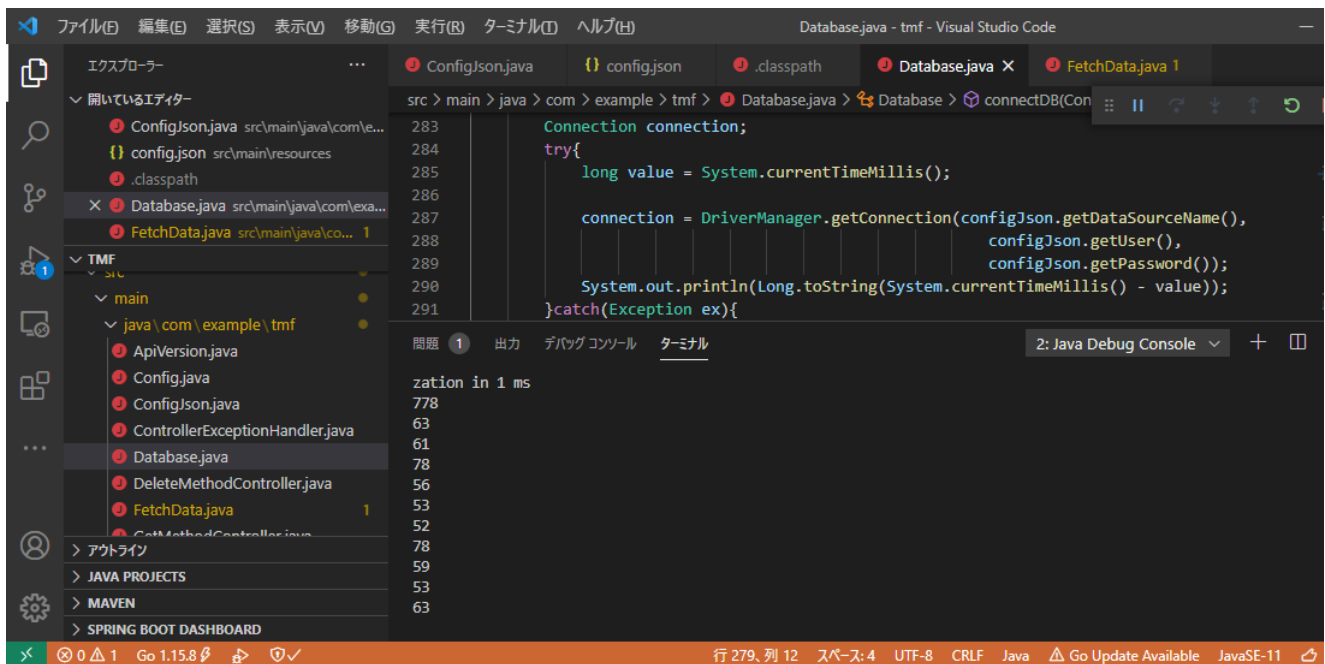
a. oci8だと約1000ms(2回目以降)



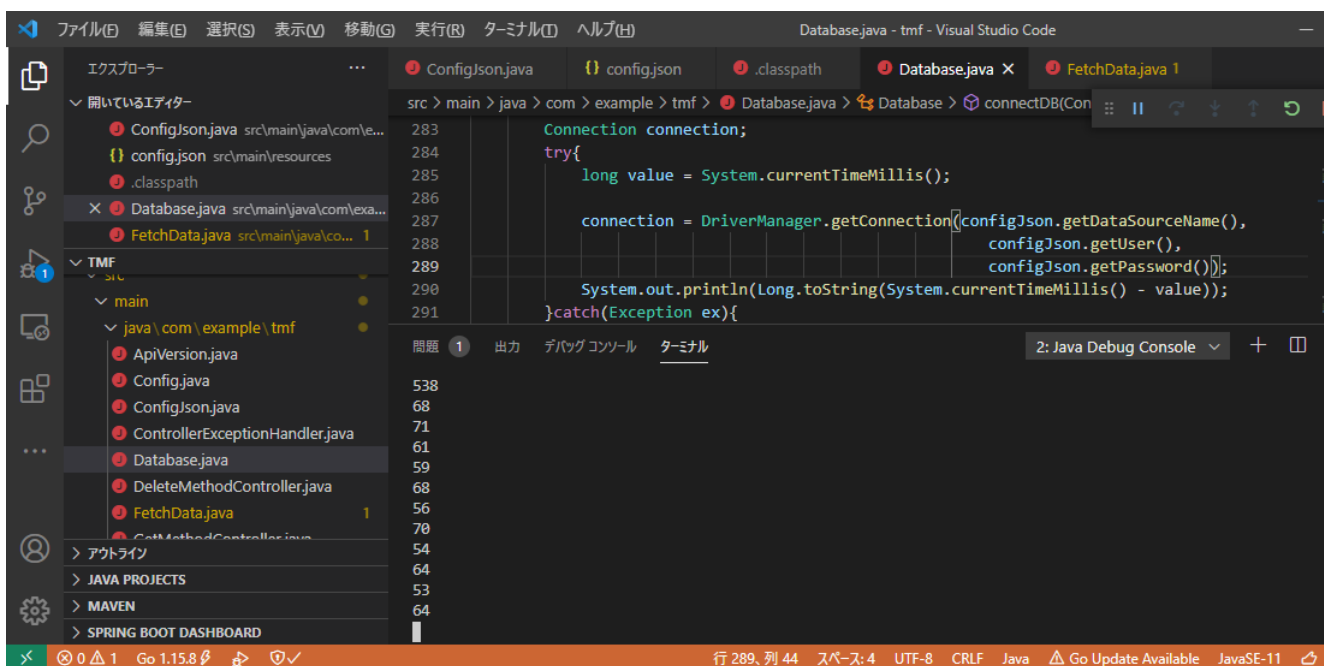
b. thinだと約70ms. 93%改善。(2回目以降)



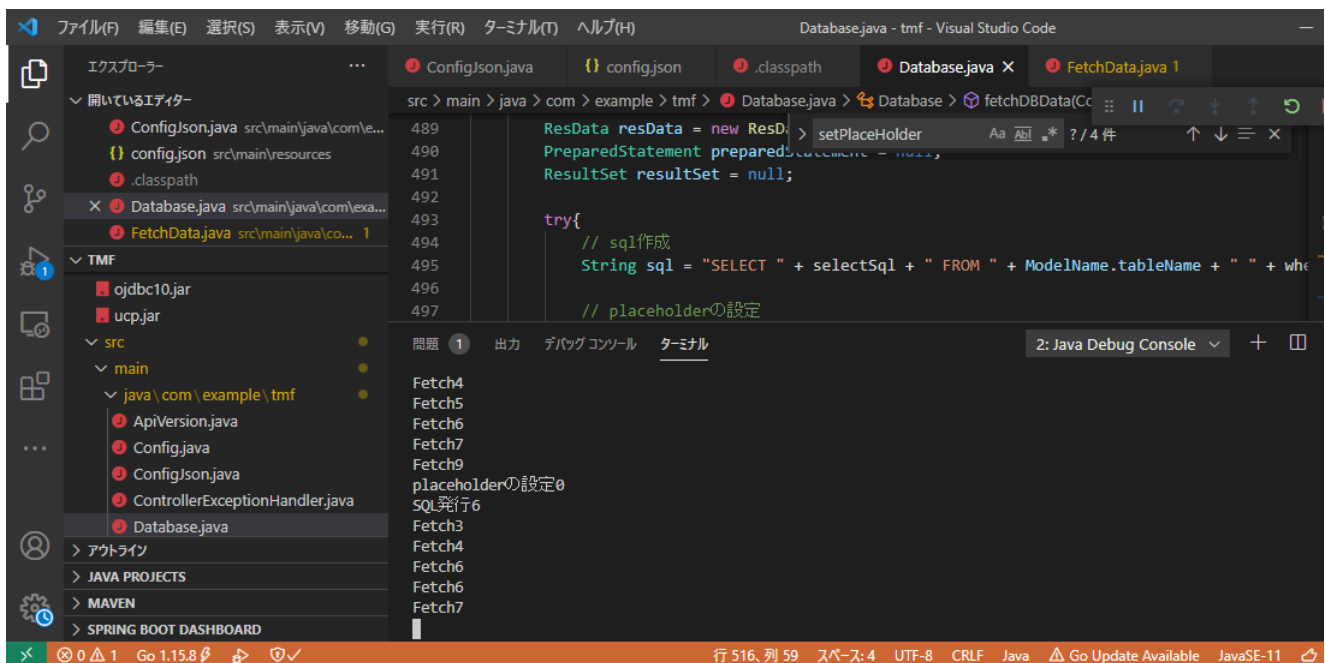
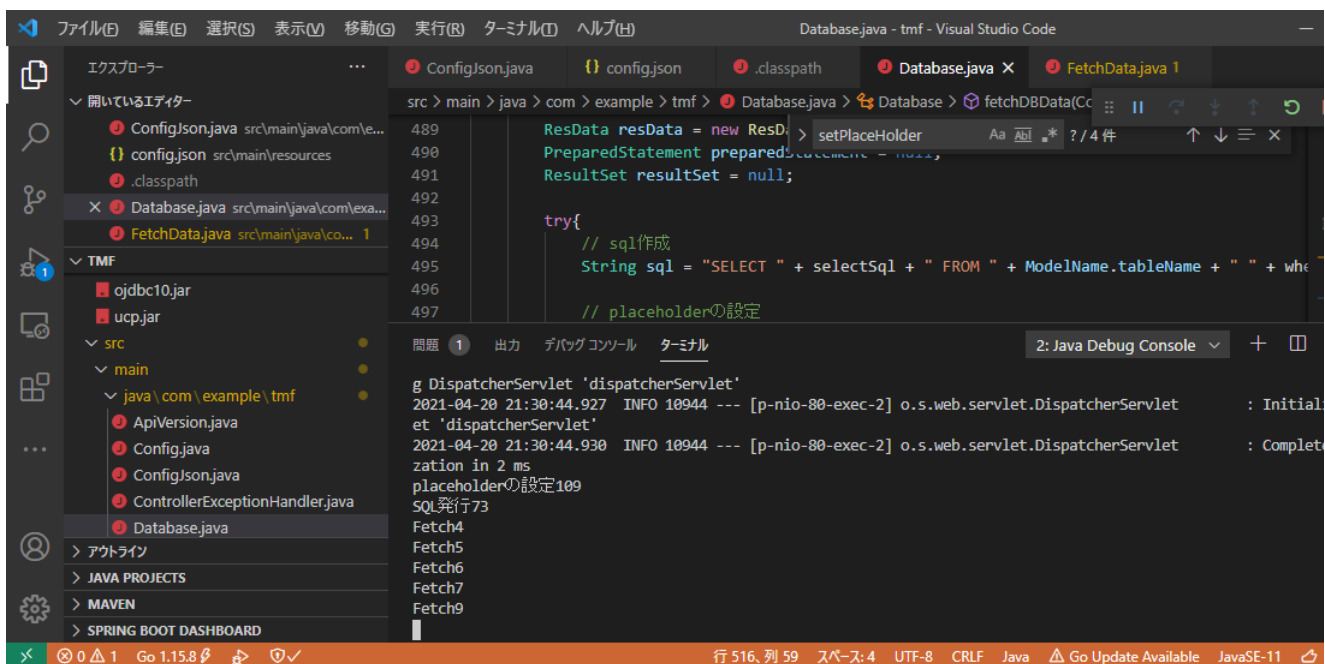
c. connectionのみ。oci8だと1回目が778ms、2回目以降は平均56ms(n=11)

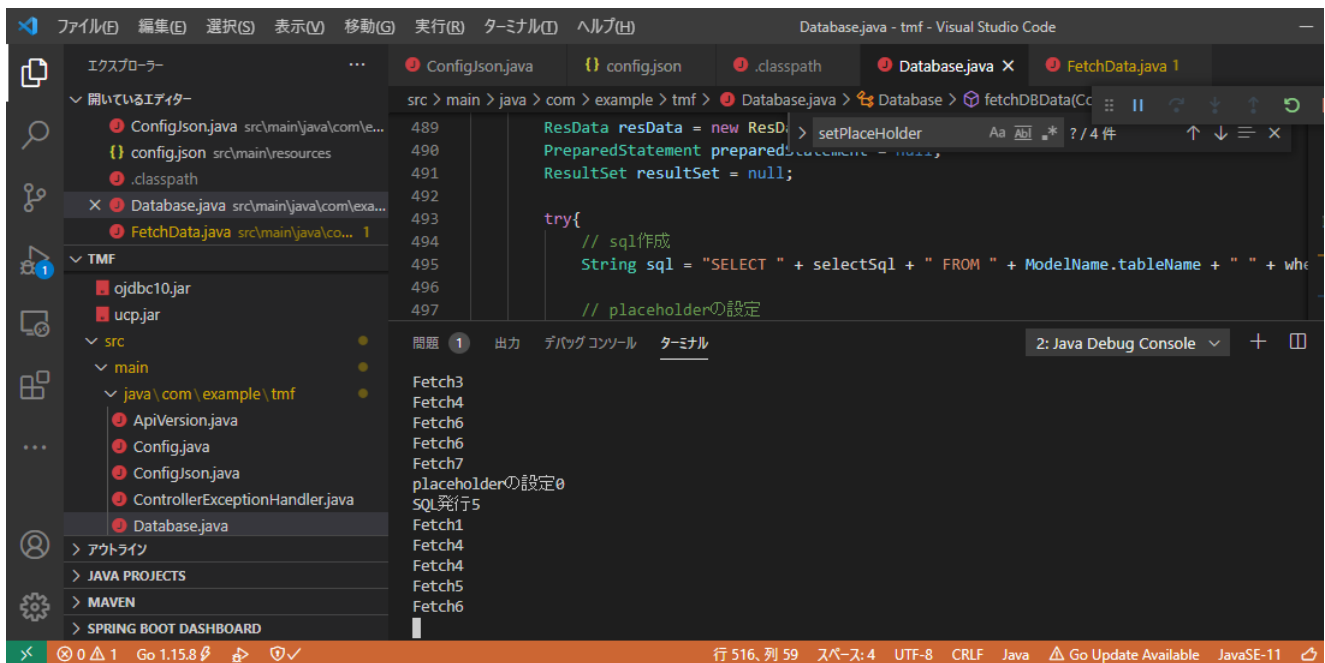


d. connectionのみ。thinだと1回目が538ms、2回目以降は平均62.5ms(n=11)。connectionの確立には有意な差は見られなかった。

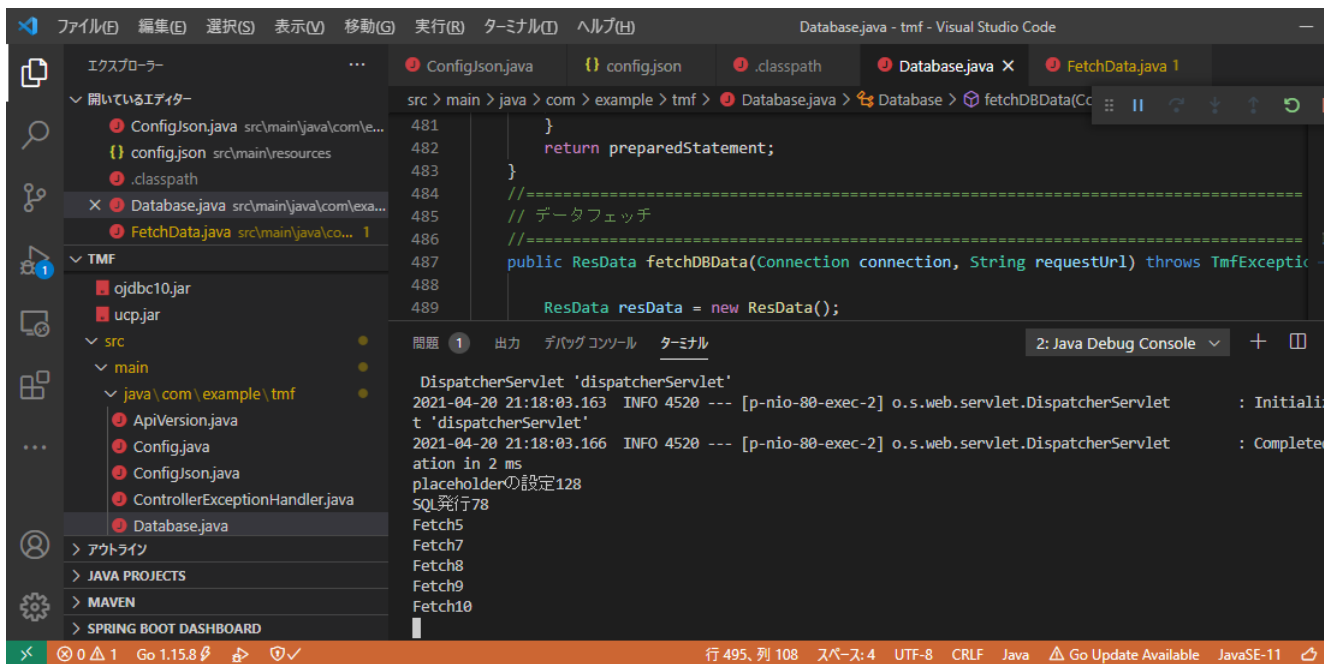


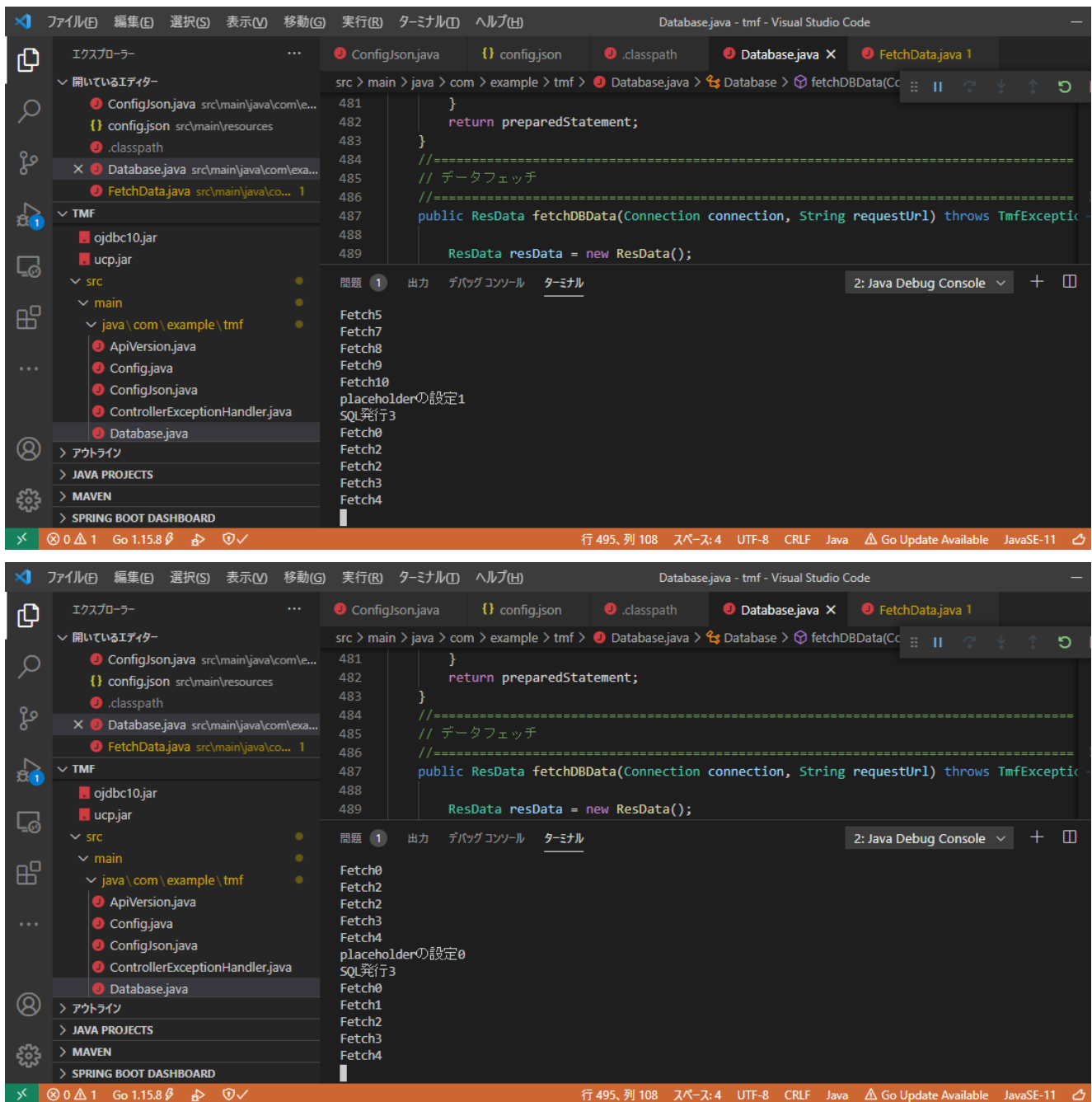
- e. ociの場合。placeholderの設定。SQL発行。Fetchを計測。preparestatementの使用によるparse処理の省略により、1回目と比較し88%-85%の高速化が図られている。



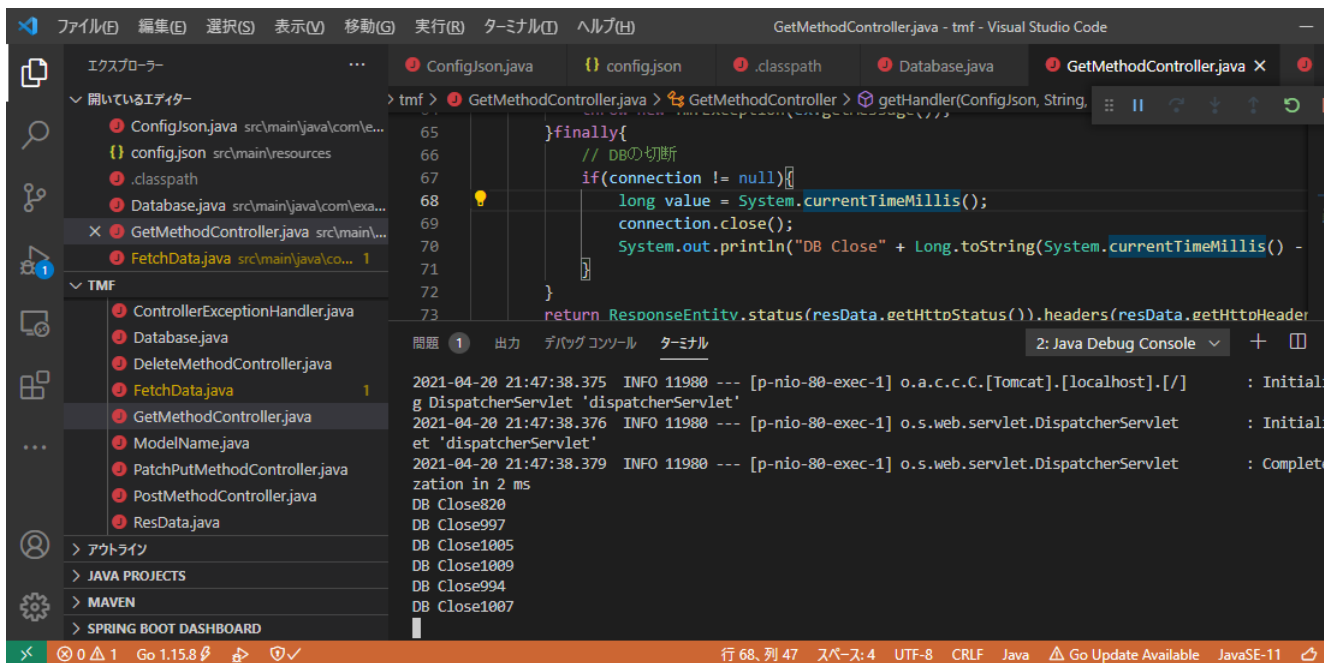


- f. thinの場合。placeholderの設定。SQL発行。Fetchを計測。preparestatementの使用によるparse処理の省略により、1回目と比較し94%-95%の高速化が図られている。oci8との有意な速度差は見られない。

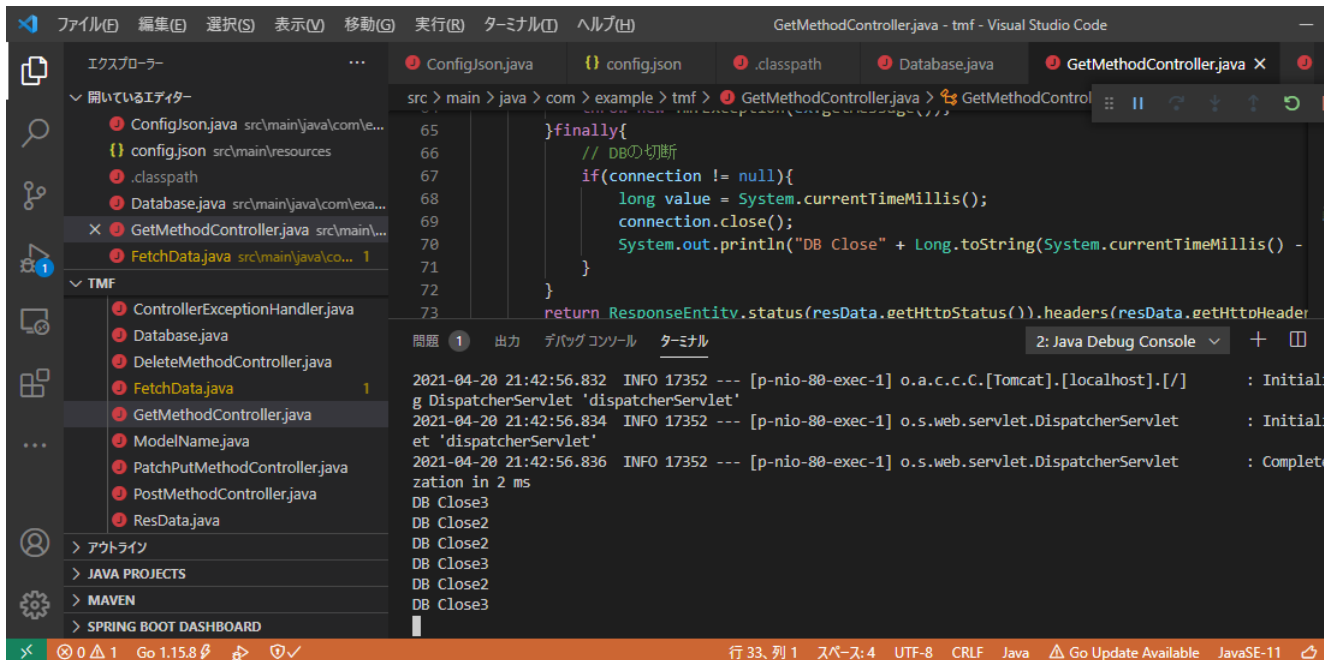




g. ociの場合。close処理に平均972S(n=6)。2回目以降の処理時間の90%以上を占める。

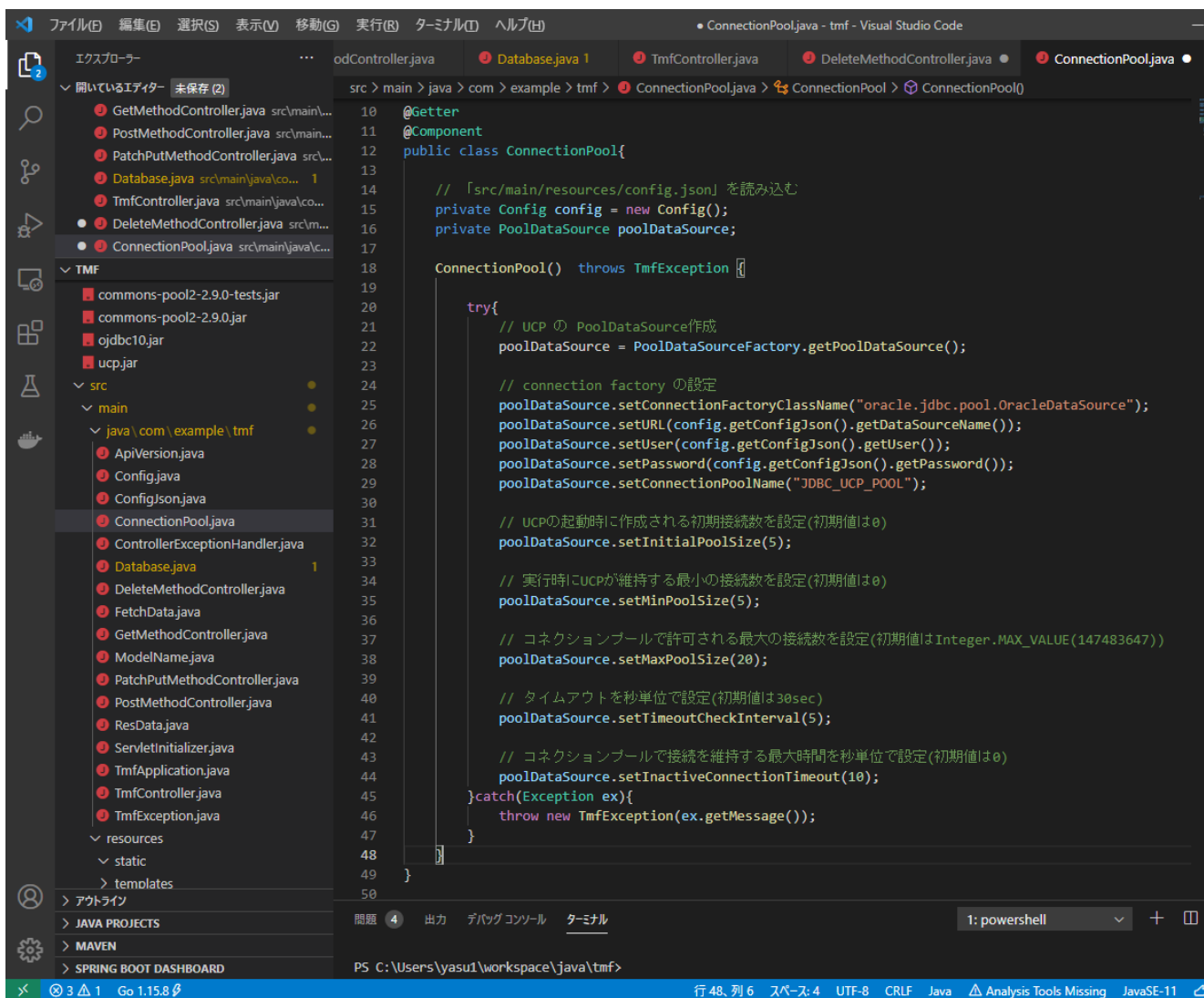


- h. thinの場合。close処理に平均2.5s(n=6)。oci8と比較し有意な速度差がでた。SBはthin標準とのことなの、原因追及はせずにthinを利用する。

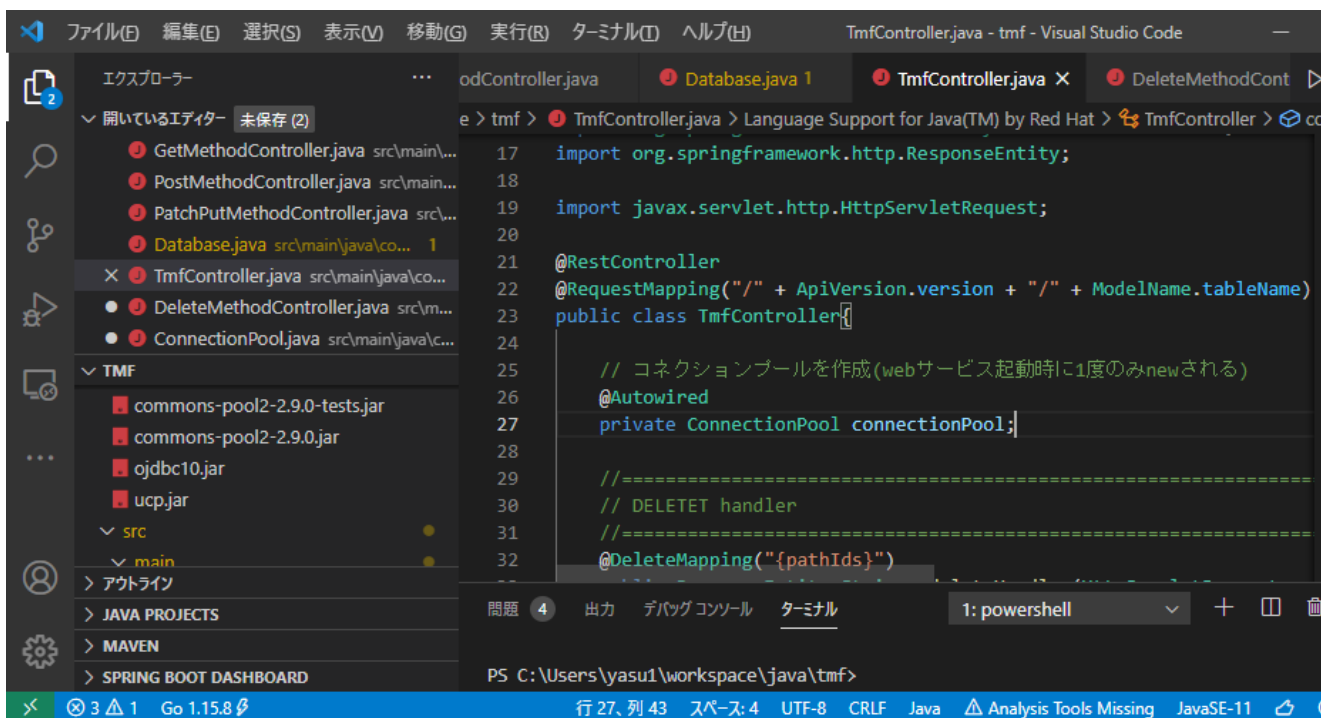


3. コネクションプールの利用

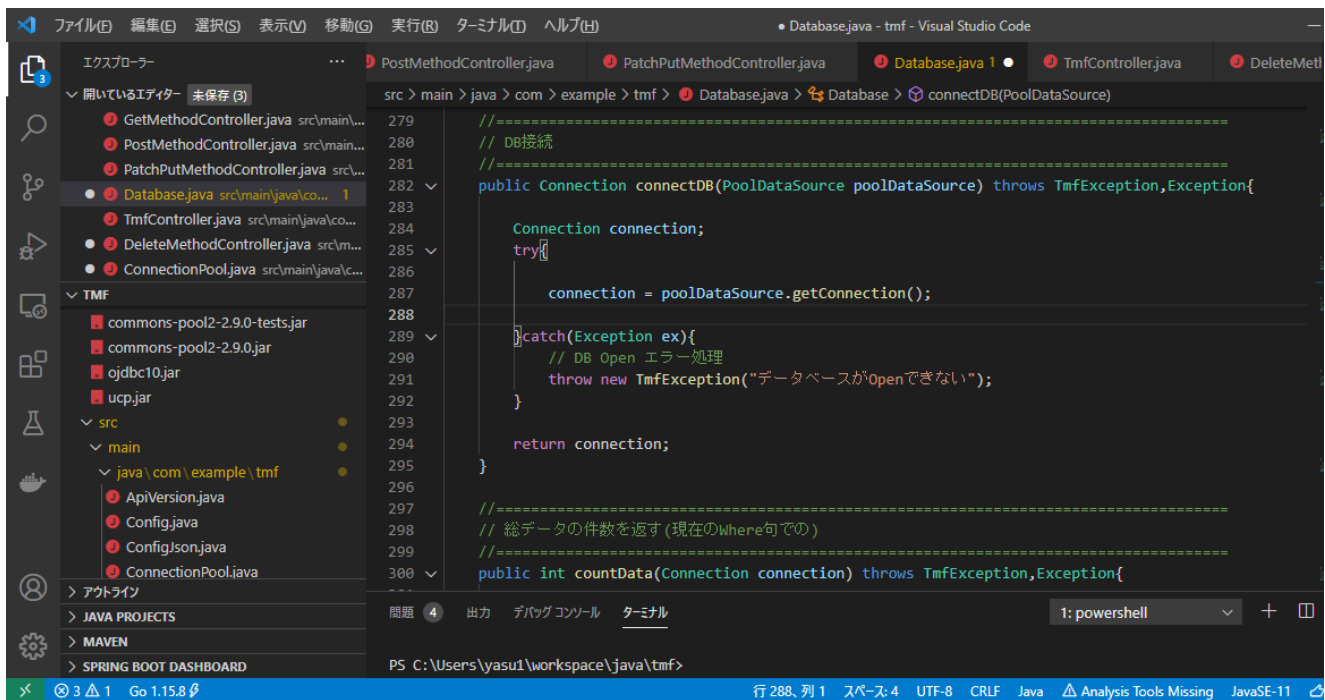
- a. ConnectionPoolクラスを作成



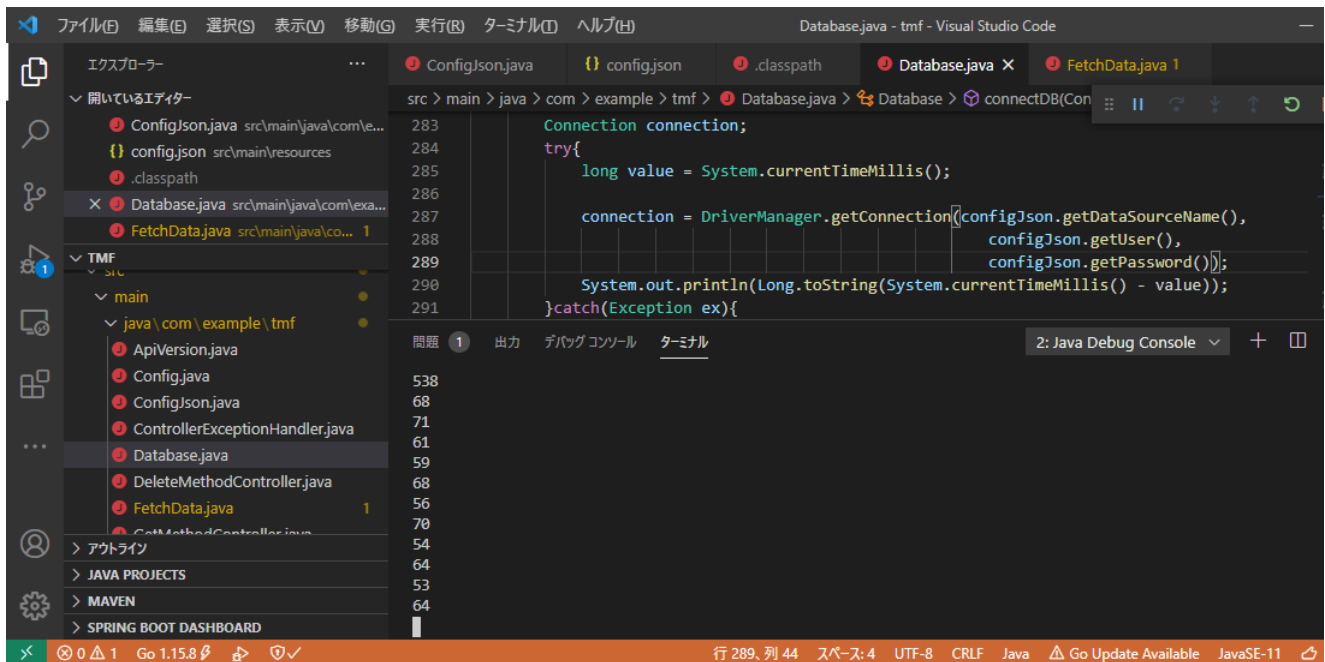
b. ConnectionPoolクラスをDI

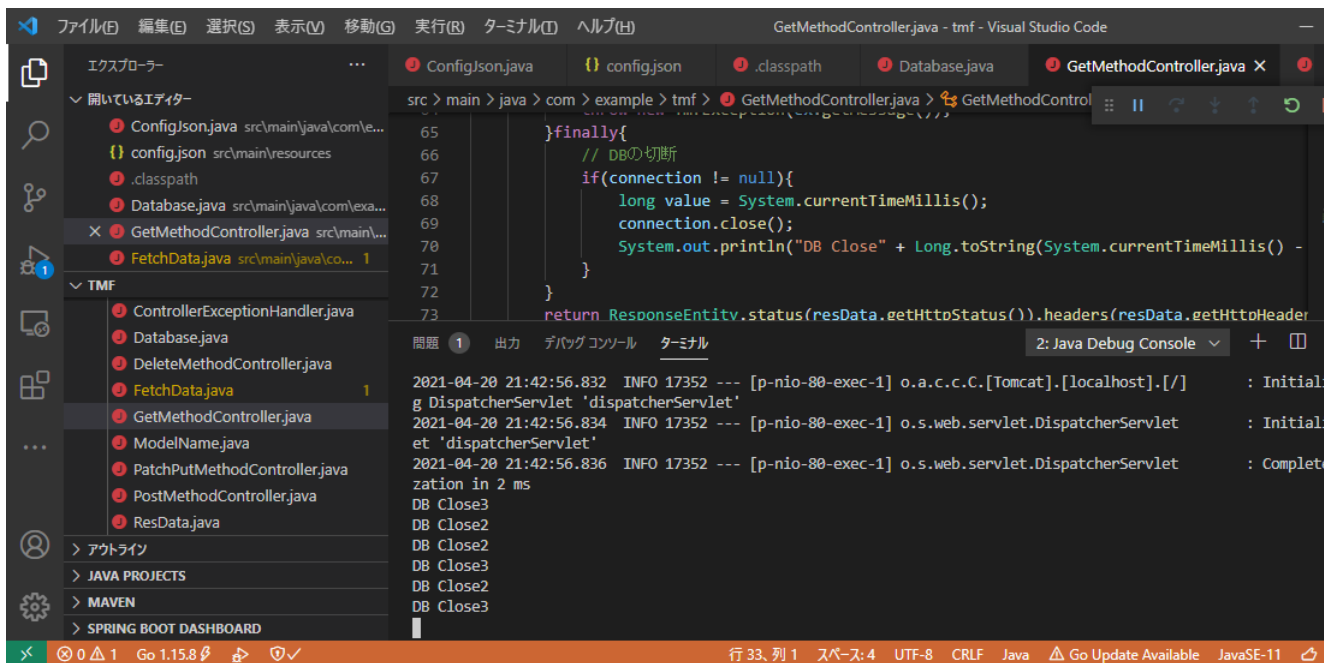


c. DB接続をコネクションプールからOpen



- d. connectionPool利用前。thinだとopen1回目が538ms、2回目以降は平均62.5ms(n=11)。close処理に平均2.5s(n=6)。





- e. connectionPool利用。thinだとopen1回目が2540ms(+472%)、2回目以降は平均0.64ms(n=11。▲99.9%)。close処理に平均0.36s(n=6。▲98.6%)。

Visual Studio Code interface showing a Java application running. The main editor displays the `Database.java` file with the following imports:

```

import java.lang.reflect.Field;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import java.text.ParseException;
import java.text.SimpleDateFormat;

```

The 'Run and Debug' sidebar on the left shows the 'Call Stack' with various threads, including:

- Thread [Reference Handler]
- Thread [Finalizer]
- Thread [Signal Dispatcher]
- Thread [Attach Listener]
- Thread [Common-Cleaner]
- Thread [Catalina-utility-1]
- Thread [Catalina-utility-2]
- Thread [container-0]
- Thread [http-nio-80-BlockPoll...]
- Thread [http-nio-80-exec-1]
- Thread [http-nio-80-exec-2]
- Thread [http-nio-80-exec-3]
- Thread [http-nio-80-exec-4]
- Thread [http-nio-80-exec-5]
- Thread [http-nio-80-exec-6]
- Thread [http-nio-80-exec-7]
- Thread [http-nio-80-exec-8]
- Thread [http-nio-80-exec-9]

The 'Breakpoints' section shows 'Uncaught Exceptions' and 'Caught Exceptions'.

The 'Terminal' pane at the bottom shows the output of the application, including log messages and database operations:

```

2021-04-21 13:11:35.110 INFO 10792 --- [p-nio-80-exec-2] o.s.w...
herServlet'
2021-04-21 13:11:35.112 INFO 10792 --- [p-nio-80-exec-2] o.s.w...
ms
DB Open2540
DB Close2
DB Open2
DB Close0
DB Open1
DB Close0
DB Open0
DB Close0
DB Open0
DB Close0
DB Open0
DB Close1
DB Open1
DB Close1
DB Open1
DB Close0
DB Open0
DB Close0
DB Open1
DB Close1
DB Open0
DB Close0
DB Open1
DB Close1
DB Open0
DB Close0
DB Open1
DB Close1

```

f. この時点のレスポンスタイム

Postman

File Edit View Help

HomeWorkspacesReportsExplore

Search Postman

My Workspace

NewImport

GET g. PUT p. POST f. GET h. DEL → + ... No Environment

Collections

New Collection

GET get

POST post

DEL delete

POST confirm

POST cancel

PUT put

PATCH patch

APIs

Environments

Mock Servers

Monitors

History

New Collection / get

Save

GET

http://localhost/v2/products

ParamsAuthHeaders (8)BodyPre-req. TestsSettings

Query Params

	KEY	VALUE	DESCRIPTION	
	Key	Value	Description	

Body

200 OK23 ms751 BSave

PrettyRawPreviewVisualizeJSON

```
1 {
2   "product_no": 1,
3   "name": "Cheese",
4   "id": "1",
5   "href": "http://localhost/v2/products/1"
```

Find and ReplaceConsoleBootcampRun