

Project4Task2

Project4API Reference: <https://www.abibliadigital.com.br/en>. It is a biledigital website which offers content with multiple versions - NIV, RA, ACF, KJV, BBE, RVR, APEE - from the Holy Bible, devotional and reference comments for study.

Function Introduction:

In this task, for android app, I design text views and submit buttons as well as URL connections to codespace to make a http request service to web service.

Also, I use Web service and two important classes such as:

1. AbibliadigitalApi

It is an interface provided for APP use, which is mainly used to call a third-party interface, get a random book of poems (return the poem in a particular book) and then return to the APP

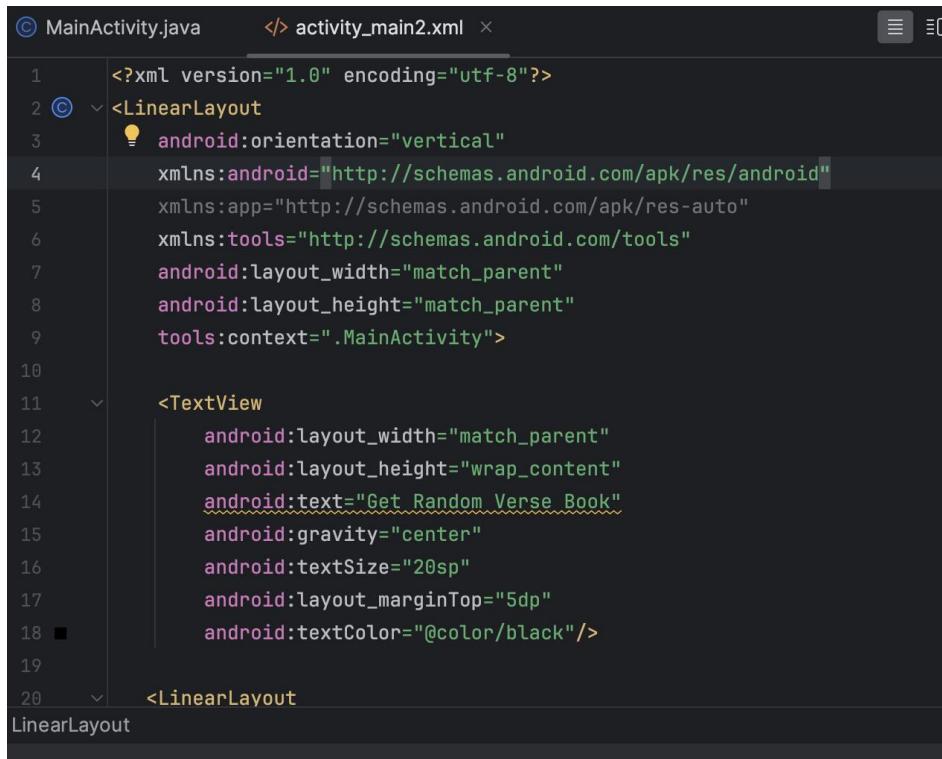
2. DashboardServlet

It is provided to the WEB Dashboard page, returns the interface call log information Returns statistics such as the last call time, the number of times the interface was accessed, the most used version, and so on

Here is the completed instructions for each steps with code and output :

1. Implement a native Android application

a. Has at least three different kinds of Views in your Layout (TextView, EditText, ImageView, or anything that extends android.view.View). In order to figure out if something is a View, find its API. If it extends android.view.View then it is a View.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     android:orientation="vertical"
4     xmlns:android="http://schemas.android.com/apk/res/android"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     xmlns:tools="http://schemas.android.com/tools"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".MainActivity">
10
11     <TextView
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:text="Get Random Verse Book"
15         android:gravity="center"
16         android:textSize="20sp"
17         android:layout_marginTop="5dp"
18         android:textColor="@color/black"/>
19
20 </LinearLayout>
```



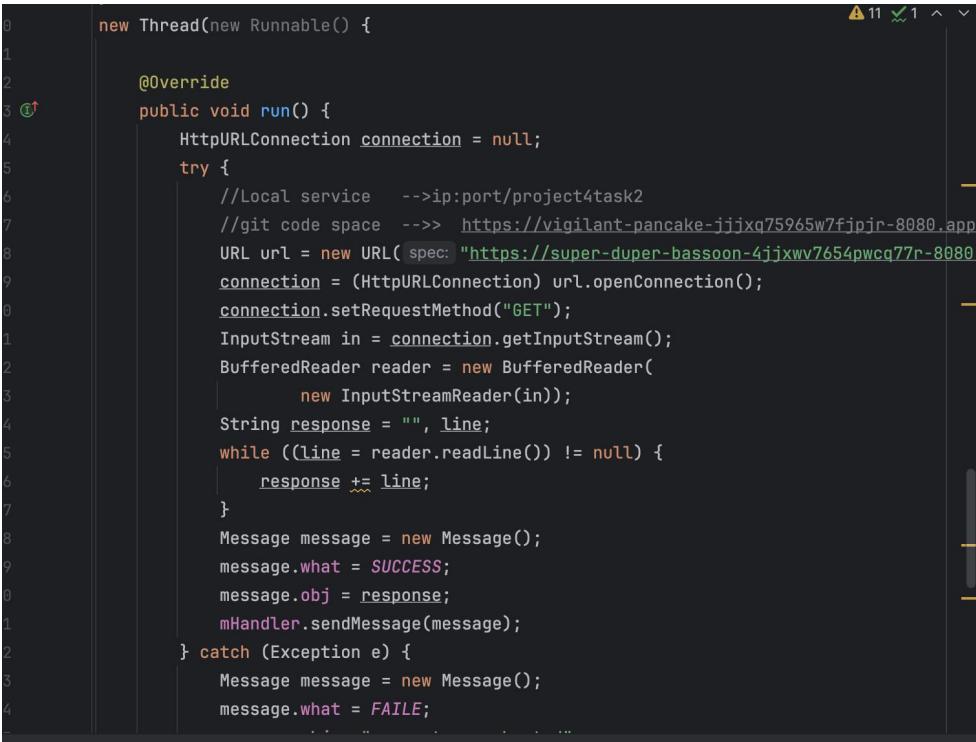
```
40 <EditText
41     android:layout_width="match_parent"
42     android:layout_height="wrap_content"
43     android:id="@+id/abbrev"/>
44 </LinearLayout>
45
46 <com.google.android.material.button.MaterialButton
47     android:layout_marginTop="10dp"
48     android:layout_width="match_parent"
49     android:layout_height="wrap_content"
50     android:text="submit"
51     android:id="@+id/btn"/>
52
53 <ListView
54     android:layout_width="match_parent"
55     android:layout_height="wrap_content"
56     android:id="@+id/listview"/>
57
58 </LinearLayout>
```

b. Requires input from the user



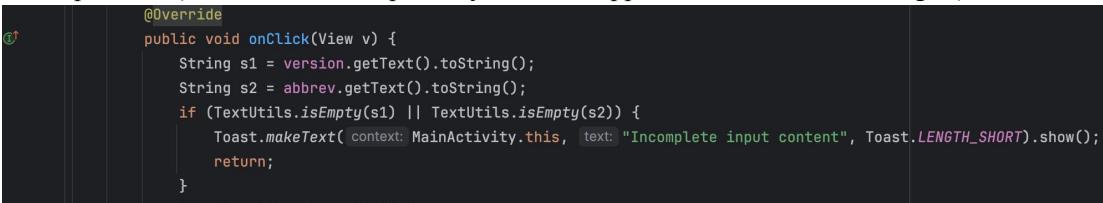
```
1 usage
100     private void initData() {
101         btn.setOnClickListener(new View.OnClickListener() {
102             @Override
103             public void onClick(View v) {
104                 String s1 = version.getText().toString();
105                 String s2 = abbrev.getText().toString();
106                 if (TextUtils.isEmpty(s1) || TextUtils.isEmpty(s2)) {
107                     Toast.makeText(context: MainActivity.this, text: "Incomplete").show();
108                     return;
109                 }
110                 new Thread(new Runnable() {
```

- c. Makes an HTTP request (using an appropriate HTTP method) to your web service
- d. Receives and parses an XML or JSON formatted reply from your web service
- e. Displays new information to the user



```
new Thread(new Runnable() {
    @Override
    public void run() {
        HttpURLConnection connection = null;
        try {
            //Local service -->ip:port/project4task2
            //git code space --> https://vigilant-pancake-ijjxq75965w7fjpjr-8080.app
            URL url = new URL(spec: "https://super-duper-bassoon-4jjxwv7654pwcq77r-8080",
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            InputStream in = connection.getInputStream();
            BufferedReader reader = new BufferedReader(
                new InputStreamReader(in));
            String response = "", line;
            while ((line = reader.readLine()) != null) {
                response += line;
            }
            Message message = new Message();
            message.what = SUCCESS;
            message.obj = response;
            mHandler.sendMessage(message);
        } catch (Exception e) {
            Message message = new Message();
            message.what = FAIL;
```

- f. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

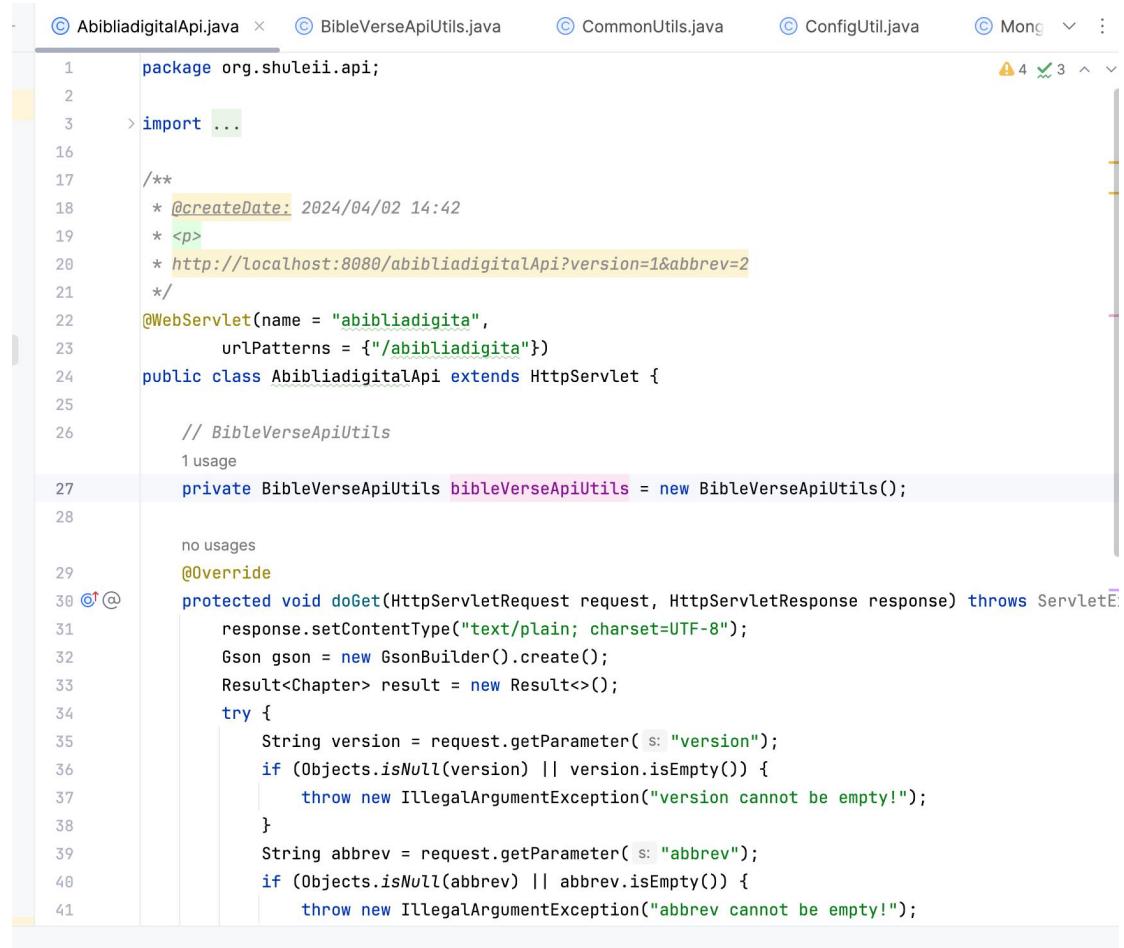


```
@Override
public void onClick(View v) {
    String s1 = version.getText().toString();
    String s2 = abbrev.getText().toString();
    if (TextUtils.isEmpty(s1) || TextUtils.isEmpty(s2)) {
        Toast.makeText(context: MainActivity.this, text: "Incomplete input content", Toast.LENGTH_SHORT).show();
    }
}
```

2. Implement a web service

- a. Implement a simple (can be a single path) API.
- b. Receives an HTTP request from the native Android application
- c. Executes business logic appropriate to your application. This includes fetching XML or JSON

information from some 3rd party API and processing the response.



```
1 package org.shuleii.api;
2
3 > import ...
4
5 /**
6 * @createDate: 2024/04/02 14:42
7 * <p>
8 * http://localhost:8080/abibliadigitalApi?version=1&abbrev=2
9 */
10 @WebServlet(name = "abibliadigital",
11             urlPatterns = {"/abibliadigital"})
12 public class AbibliadigitalApi extends HttpServlet {
13
14     // BibleVerseApiUtils
15     1 usage
16     private BibleVerseApiUtils bibleVerseApiUtils = new BibleVerseApiUtils();
17
18     no usages
19     @Override
20     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
21         response.setContentType("text/plain; charset=UTF-8");
22         Gson gson = new GsonBuilder().create();
23         Result<Chapter> result = new Result<>();
24         try {
25             String version = request.getParameter( s: "version");
26             if (Objects.isNull(version) || version.isEmpty()) {
27                 throw new IllegalArgumentException("version cannot be empty!");
28             }
29             String abbrev = request.getParameter( s: "abbrev");
30             if (Objects.isNull(abbrev) || abbrev.isEmpty()) {
31                 throw new IllegalArgumentException("abbrev cannot be empty!");
32             }
33             result = bibleVerseApiUtils.getChapter(version, abbrev);
34         } catch (Exception e) {
35             log.error("Error occurred while fetching chapter: " + e.getMessage());
36             result.setError(true);
37             result.setMessage("An error occurred while fetching the chapter. Please try again later.");
38         }
39         gson.toJson(result, response.getWriter());
40     }
41 }
```

3. Handle error conditions

Your application should test for and handle gracefully:

- a.Invalid mobile app input
- b.Invalid server-side input (regardless of mobile app input validation)

```

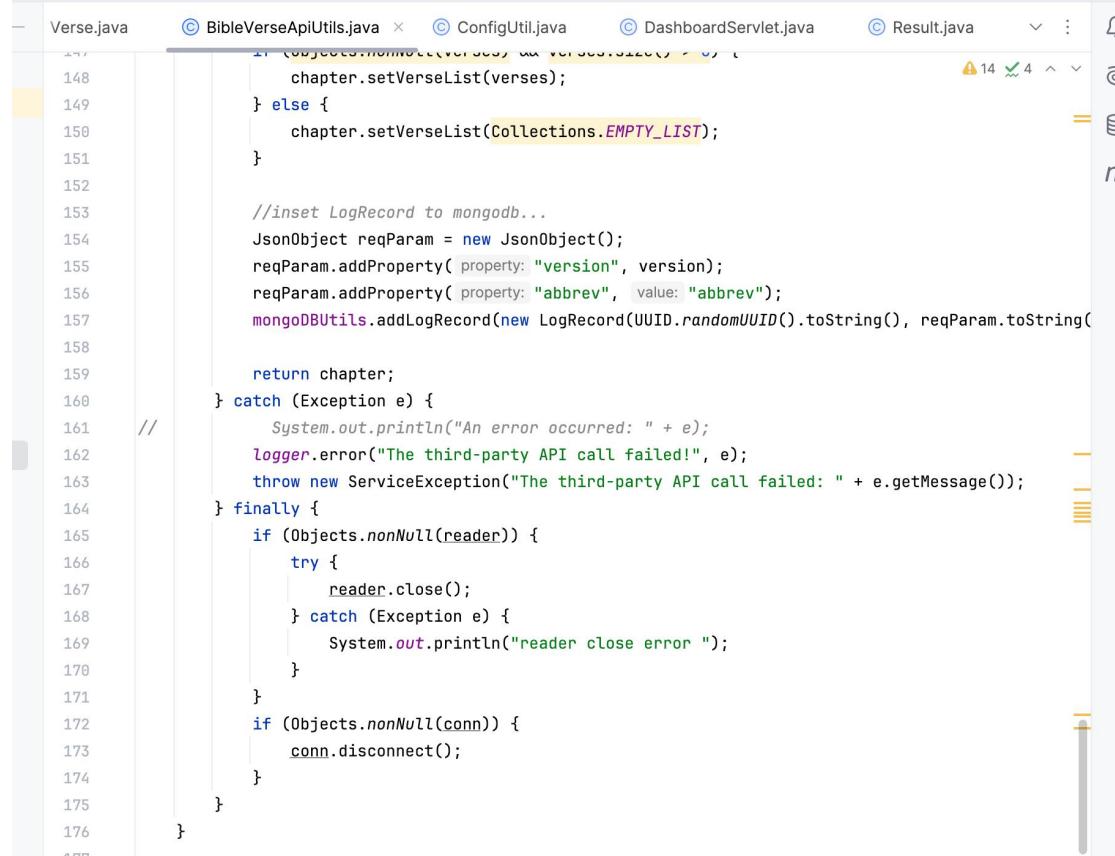
53     Result<Chapter> result = new Result<>();
54     try {
55         String version = request.getParameter( s: "version");
56         if (Objects.isNull(version) || version.isEmpty()) {
57             throw new IllegalArgumentException("version cannot be empty!");
58         }
59         String abbrev = request.getParameter( s: "abbrev");
60         if (Objects.isNull(abbrev) || abbrev.isEmpty()) {
61             throw new IllegalArgumentException("abbrev cannot be empty!");
62         }
63
64         Chapter chapter = bibleVerseApiUtils.sendRequest(version, abbrev);
65         if (Objects.nonNull(chapter)) {
66             result.setCode(Constants.SUCCESS_CODE);
67             result.setData(chapter);
68         }
69         // Set the response content type to JSON
70         response.setContentType("application/json");
71         response.getWriter().write(gson.toJson(result));
72     } catch (Exception e) {
73         result.setCode(Constants.FAIL_CODE);
74         result.setMsg(e.getMessage());
75         response.getWriter().write(gson.toJson(result));
76     }
77 }

```

c. Mobile app network failure, unable to reach server

d. Third-party API unavailable

e. Third-party API invalid data



```

Verse.java          BibleVerseApiUtils.java ×      ConfigUtil.java      DashboardServlet.java      Result.java      ▾  ⋮
148     chapter.setVerseList(verses);
149 } else {
150     chapter.setVerseList(Collections.EMPTY_LIST);
151 }
152
153 //insert LogRecord to mongodb...
154 JsonObject reqParam = new JsonObject();
155 reqParam.addProperty( property: "version", version);
156 reqParam.addProperty( property: "abbrev", value: "abbrev");
157 mongoDBUtils.addLogRecord(new LogRecord(UUID.randomUUID().toString(), reqParam.toString()
158
159     return chapter;
160 } catch (Exception e) {
161     System.out.println("An error occurred: " + e);
162     logger.error("The third-party API call failed!", e);
163     throw new ServiceException("The third-party API call failed: " + e.getMessage());
164 } finally {
165     if (Objects.nonNull(reader)) {
166         try {
167             reader.close();
168         } catch (Exception e) {
169             System.out.println("reader close error ");
170         }
171     }
172     if (Objects.nonNull(conn)) {
173         conn.disconnect();
174     }
175 }
176

```

```

MainActivity.java x activity_main2.xml
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
    InputStream in = connection.getInputStream();
    BufferedReader reader = new BufferedReader(
        new InputStreamReader(in));
    String response = "", line;
    while ((line = reader.readLine()) != null) {
        response += line;
    }
    Message message = new Message();
    message.what = SUCCESS;
    message.obj = response;
    mHandler.sendMessage(message);
} catch (Exception e) {
    Message message = new Message();
    message.what = FAIL;
    message.obj = "request was aborted";
    mHandler.sendMessage(message);
    e.printStackTrace();
} finally {
    if (connection != null) {
        connection.disconnect();
    }
}
}).start();

```

4. Log useful information

At least 6 pieces of information is logged for each request/reply with the mobile phone. It should include information about the request from the mobile phone, information about the request and reply to the 3rd party API, and information about the reply to the mobile phone. (You should NOT log data from interactions from the operations dashboard.)

Operations Dashboard

Operations Analytics

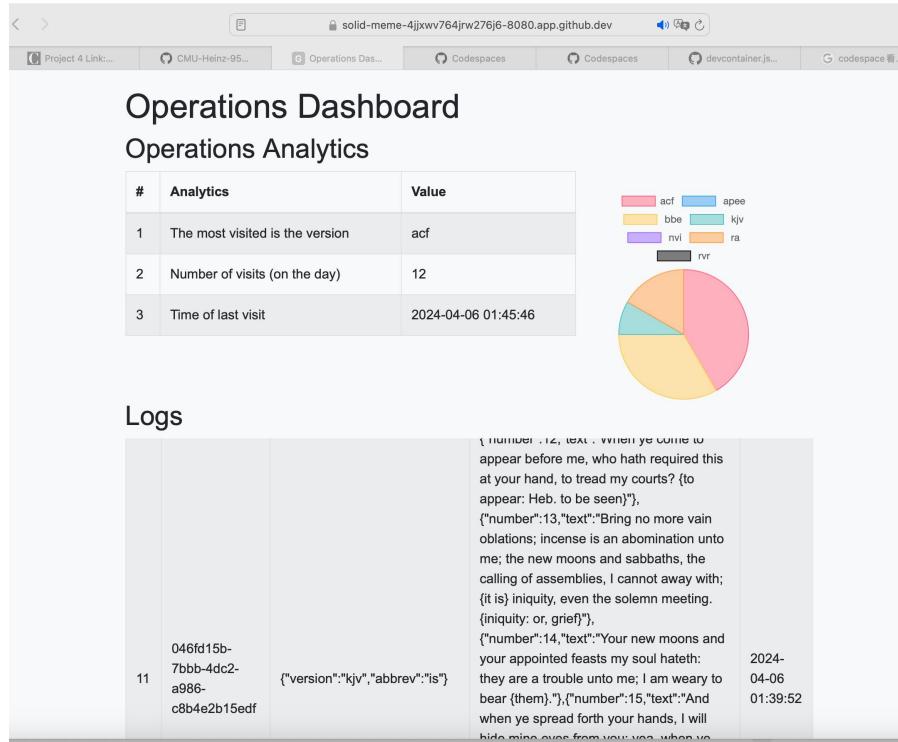
#	Analytics	Value
1	The most visited is the version	acf
2	Number of visits (on the day)	12
3	Time of last visit	2024-04-06 01:45:46



Logs

#	uuid	request param	response	Request time

// In this api, acf,apee,bbe refers to different versions of bible story, like acf refers to Audubon Christian Fellowship.



5. Store the log information in a database

The web service can connect, store, and retrieve information from a MongoDB database in the cloud.

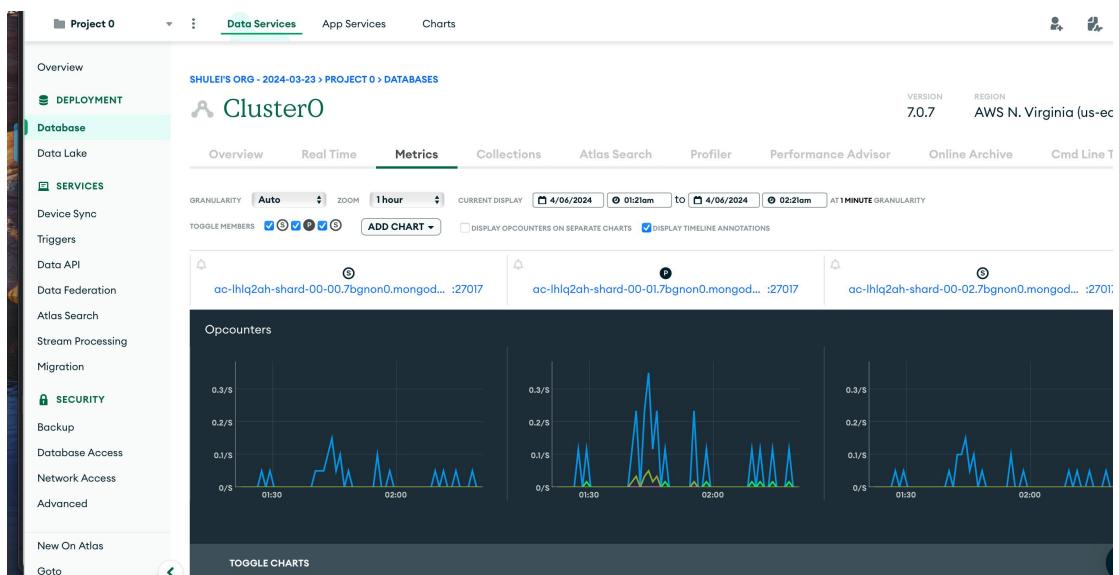
The screenshot shows a Java project structure on the left and a code editor on the right.

Project Structure:

- Project
- .idea
- out
- pom.xml
- project4task2.iml
- src
 - main
 - java
 - org.shuleii
 - api
 - AbibladigitalApi
 - DashboardServlet
 - common
 - BibleVerseApiUtils
 - CommonUtils
 - ConfigUtil
 - Constants
 - MongoDBUtils
 - Result
 - ex
 - model
 - Apistatistics
 - Chapter
 - LogRecord
 - Verse
 - resources
 - application.properties
 - test
 - tarnet

Code Editor (BibleVerseApiUtils.java):

```
29 * @description: addMovie
30 * @author: shuleii
31 * @return:
32 */
33 public Boolean addLogRecord(LogRecord logRecord) {
34     // Connect to the MongoDB service
35     try (MongoClient mongoClient = MongoClients.create(ConfigUtil.getMongoDBUri())) {
36         //Selective database
37         MongoDatabase database = mongoClient.getDatabase("LOGRECORD_DATABASE");
38
39         // Selection set
40         MongoCollection<Document> collection = database.getCollection("LOGRECORD_COLLECTION");
41
42         // Create movie information to insert
43         Document document = new Document()
44             .append("id", logRecord.getId())
45             .append("request", logRecord.getRequest())
46             .append("response", logRecord.getResponse())
47             .append("requestTime", logRecord.getRequestTime());
48
49         // Insert document
50         InsertOneResult insertOneResult = collection.insertOne(document);
51
52         logger.info("add res:" + insertOneResult.wasAcknowledged());
53         return insertOneResult.wasAcknowledged();
54     } catch (Exception e) {
55         logger.error("add LogRecord error", e);
56     }
57
58     return false;
59 }
```



In mongodb, it records the connect and retrieve of web service recently. And default timing is set 3 hours earlier than United States

(GMT-4).

Screenshot of a web-based Operations Dashboard titled "Operations Analytics".

The dashboard displays the following information:

- A table showing three analytics items:

#	Analytics	Value
1	The most visited is the version	bbe
2	Number of visits (on the day)	15
3	Time of last visit	2024-04-06 02:32:22
- A pie chart showing the distribution of visits across different versions:
 - acf (Red)
 - bbf (Yellow)
 - apee (Blue)
 - kjv (Teal)
 - mvl (Purple)
 - ra (Orange)
 - rvr (Grey)
- A section titled "Logs" showing a single log entry:

ID	Timestamp	Message	Time	
11	c09209ea-6ade-4ddf-b1c9-2a62c5ce8ae0	{"version": "acf", "abbrev": "gr"}	{number: 9, text: "Que ele me de a cova de macpeia, que ele tem no fim do seu campo; que ma dê pelo devido preço em herança de sepulcro no meio de vós."}, {"number": 10, "text": "Ora Efrom habitava no meio dos filhos de Hete; e respondeu Efrom, heteu, a Abraão, aos ouvidos dos filhos de Hete, de todos os que entravam pela porta da sua cidade, dizendo:"}, {"number": 11, "text": "Não, meu senhor, ouve-me: O campo te dou, também te dou a cova que nele está, diante dos olhos dos filhos do meu povo ta dou; sepulta a tua morta."}, {"number": 12, "text": "Então Abraão se inclinou diante da face do povo da terra."}, {"number": 13, "text": "E falou a Efrom, aos ouvidos do povo da terra, dizendo: Mas se tu estás por isto, ouve-me, pego-te. O preço do campo o darei; toma-o de mim e sepultarei ali a minha morta."}, {"number": 14, "text": "E respondeu Efrom a Abraão, dizendo-lhe:"}, {"number": 15, "text": "Meu senhor, ouve-me, a terra é de quatrocentos siclos de prata; que é isto entre mim e ti? Sepulta"}]	2024-04-06 01:43:23

project4.apiLogRecord

STORAGE SIZE: 104KB LOGICAL DATA SIZE: 60.53KB TOTAL DOCUMENTS: 13 INDEXES TOTAL SIZE: 36KB

Find **Indexes** **Schema Anti-Patterns** **Aggregation** **Search Indexes** **INSERT DOCUMENT**

Filter Type a query: { field: 'value' } **Reset** **Apply** **Options**

QUERY RESULTS: 1-13 OF 13

```

_id: ObjectId('6610a4ff65c7af25d9a59f58')
id : "75dd236d-a15c-4755-a3b4-edc1b972ec7d"
request : {"version":"bbe","abbrev":"gn"}"
response : [{"number":27,"verses":46,"verseList":[{"number":1,"text":"Now when Isa..."}
requestTime : 2024-04-06T01:27:27.732+00:00

_id: ObjectId('6610a7e965c7af25d9a59f5b')
id : "946fd15b-7bbb-4dc2-a986-c8b4e2b15edf"
request : {"version":"kjv","abbrev":"is"}"
response : [{"number":1,"verses":31,"verseList":[{"number":1,"text":"The vision of..."}
requestTime : 2024-04-06T01:39:52.988+00:00

_id: ObjectId('6610a84b65c7af25d9a59f5d')
id : "6610a84b65c7af25d9a59f5d"
request : {"version":"kjv","abbrev":"is"}"
response : [{"number":1,"verses":1,"verseList":[{"number":1,"text":"Now after tw..."}
requestTime : 2024-04-06T01:45:38.010+00:00

_id: ObjectId('6610a94a65c7af25d9a59f6f')
id : "7c34bbef-0f44-4830-a7ab-37e94df6abad"
request : {"version":"bbe","abbrev":"gn"}"
response : [{"number":32,"verses":32,"verseList":[{"number":1,"text":"And on his w..."}
requestTime : 2024-04-06T01:45:46.469+00:00

_id: ObjectId('6610ab7365c7af25d9a59f72')
id : "cb8e0e3c-f96c-4c08-ab60-df924e04eef"
request : {"version":"bbe","abbrev":"gn"}"
response : [{"number":50,"verses":26,"verseList":[{"number":1,"text":"And Joseph p..."}
requestTime : 2024-04-06T01:54:59.839+00:00

```

Each id, request and response as well as request time are both recorded in the project4.apiLogRecord in Cluster0 in database of MongoDB.

6. Display operations analytics and full logs on a web-based dashboard

- A unique URL addresses a web interface dashboard for the web service.

Solid Meme Application Dashboard

The dashboard displays operations analytics:

#	Analytics	Value
1	The most visited is the version	bbe
2	Number of visits (on the day)	13

- b. The dashboard displays at least 3 interesting operations analytics logs.

Solid Meme Application Dashboard

The dashboard displays operations analytics and logs:

#	Analytics	Value
1	The most visited is the version	acf
2	Number of visits (on the day)	12
3	Time of last visit	2024-04-06 01:45:46

Logs:

12	75dd236d-a15c-4755-a3b4-edc1b972ec7d	{"version":"bbe","abbrev":"gn"}	sonuuuzorfxzr;s meat, so that I may give you a blessing. And he put it before him and he took it; and he gave him wine, and he had a drink.", {"number":26,"text":"And his father Isaac said to him, Come near now, my son, and give me a kiss."}, {"number":27,"text":"And he came near and gave him a kiss; and smelling the smell of his clothing, he gave him a blessing, and said, See, the smell of my son is like the smell of a field on which the blessing of the Lord has come:"}, {"number":28,"text":"May God give you the dew of heaven, and the good things of the earth, and grain and wine in full	2024-04-06 01:27:27
----	--------------------------------------	---------------------------------	--	---------------------

Pie chart showing distribution of versions:

//pie chart, analytics and click volume statistics.

- c. The dashboard displays formatted full

Logs				Request time
#	uuid	request param	response	
3	858f6751-5db6-4295-b8a9-0f57a9abd939	{"version":"bbe","abbrev":"gn"}	I saw seven heads of grain, thin and good, coming up on one stem: 1,(number:1, text: "And when I saw seven other heads, dry, thin, and wasted by the east wind, coming up after them.");,(number:24, text:"And the seven thin heads made a meal of the seven good heads; and I put this dream before the wise men, but no one of them was able to give me the sense of it.");,(number:25, text:"Then Joseph said, These two dreams have the same sense: God has made clear to Pharaoh what he is about to do.");,(number:26, text:"The seven fat cows are seven years, and the seven good heads of grain are seven years: the two have the same sense.");,(number:27, text:"The seven thin and poor-looking cows who came up after them are seven years; and the seven heads of grain, dry and wasted by the east wind, are seven years when there will be no food.");,(number:28, text:"As I said to Pharaoh before, God has made clear to him what he is about to do.");,(number:29, text:"Seven years are coming in which there will be great wealth of grain in Egypt.");,(number:30, text:"And after that will come seven years when there will not be enough food; and the memory of the good years will go from men\u0026#x27;s minds; and the land will be made waste by the bad years.");,(number:31, text:"And men will have no memory of the good time because of the need which will come after, for it will be very bitter.");,(number:32, text:"And this dream came to Pharaoh twice, because this thing is certain, and God will quickly make it come about.");,(number:33, text:"And now let Pharaoh make search for a man of wisdom and good sense, and put him in authority over the land of Egypt.");,(number:34, text:"Let Pharaoh do this, and let him put overseers over the land of Egypt to put in store a fifth part of the produce of the land in the good years.");,(number:35, text:"And let them get together all the food in those good years and make a store of grain under Pharaoh\u0026#x27;s control for the use of the towns, and let them keep it.");,(number:36, text:"And let that food be kept in store for	2024-04-06 01:45:38

7. Deploy the web service to GitHub Codespaces

A .devcontainer.json file is set up in the GitHub Codespaces, specifying the Dockerfile to build the container and configuring port 8080 to be forwarded.

The ROOT.war web application is deployed and running in a development container on GitHub Codespaces, with Catalina as the servlet container.

Port 8080 is exposed and set to public visibility, allowing external access to the web application.

Accessing the public URL of the deployed "Hello World" web application leads to a GitHub warning page, which is standard for development ports in Codespaces.

The screenshot shows a GitHub Codespace interface. On the left, there's a sidebar with icons for file management and project settings. The main area has tabs for 'README.md' and 'devcontainer.json'. The 'devcontainer.json' tab is active, displaying the following JSON code:

```

1  {
2     "build": {
3         "dockerfile": "Dockerfile",
4         "forwardPorts": [8080]
5     }
6 }

```

Below the code editor is a terminal window showing Tomcat startup logs:

```

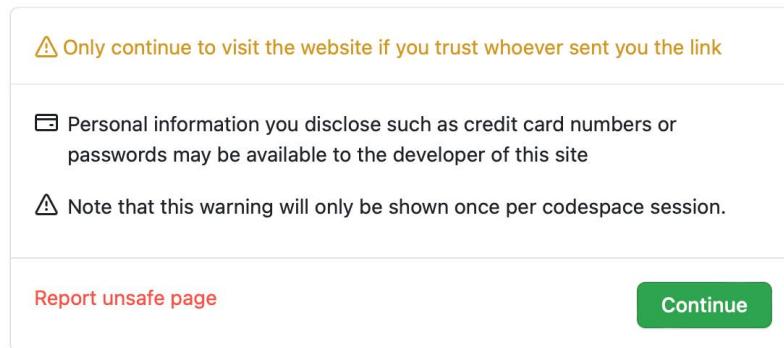
06-Apr-2024 01:08:57.955 INFO [main] org.apache.catalina.core.AprLifecycleListener.initializeSSL OpenSSL successfully initialized [OpenSSL 1.1.1k 25 Mar 2021]
06-Apr-2024 01:08:58.400 INFO [main] org.apache.AbstractProtocol.init Initializing ProtocolHandler ["http-nio-8080"]
06-Apr-2024 01:08:58.436 INFO [main] org.apache.catalina.startup.Catalina.load Server initialization in [780] milliseconds
06-Apr-2024 01:08:58.441 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet engine: [Apache Tomcat/10.1.0-M5]
06-Apr-2024 01:08:58.533 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet engine: [Apache Tomcat/10.1.0-M5]
06-Apr-2024 01:08:58.533 INFO [main] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive [/usr/local/tomcat/webapps/ROOT.war]
06-Apr-2024 01:08:59.132 INFO [main] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Enabling debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
06-Apr-2024 01:08:59.195 INFO [main] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive [/usr/local/tomcat/webapps/ROOT.war] has finished in [711] milliseconds
06-Apr-2024 01:08:59.198 INFO [main] org.apache.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
06-Apr-2024 01:08:59.208 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [771] milliseconds

```

At the bottom, a browser window shows a GitHub page at port 8080.



You are about to access a development port
served by someone's codespace



Next: You'll be redirected to: solid-meme-4jxwv764jrw276j6-8080.app.github.dev

Web link

<https://solid-meme-4jxwv764jrw276j6-8080.app.github.dev>

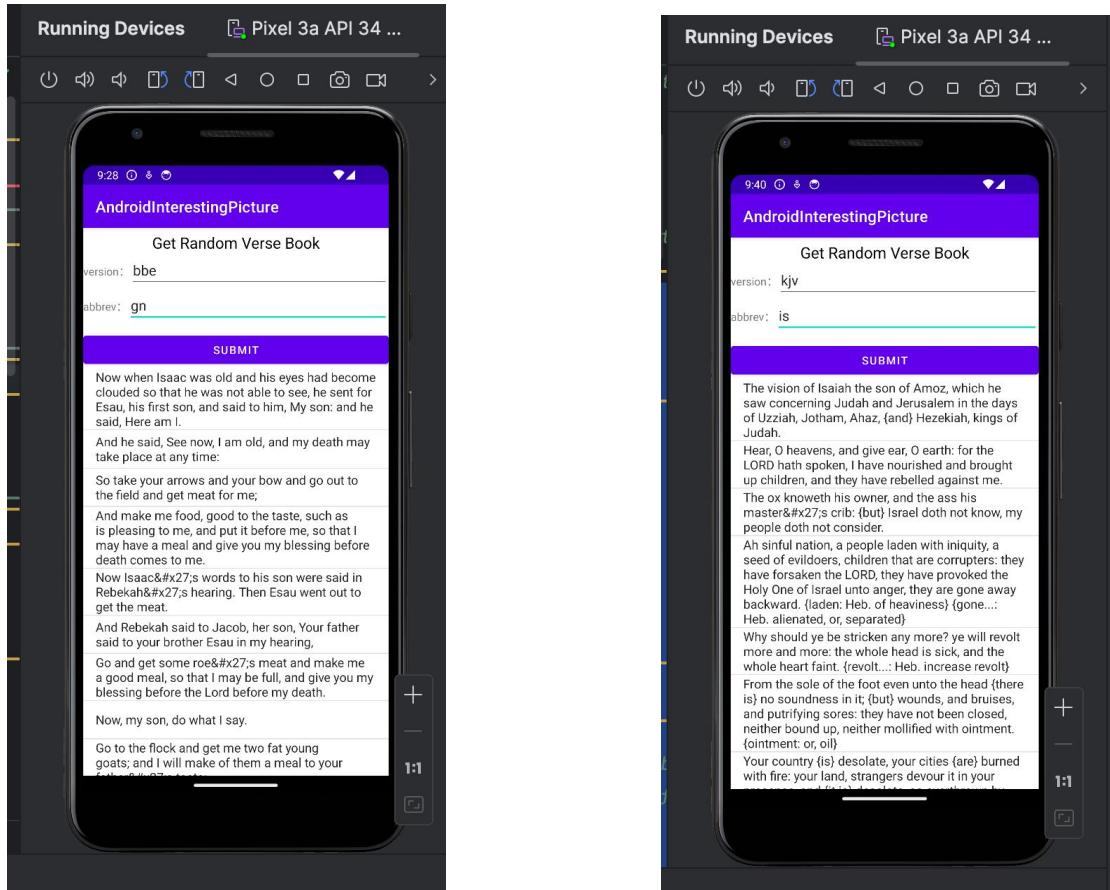
Change the url link at andriodstudio to connect them.

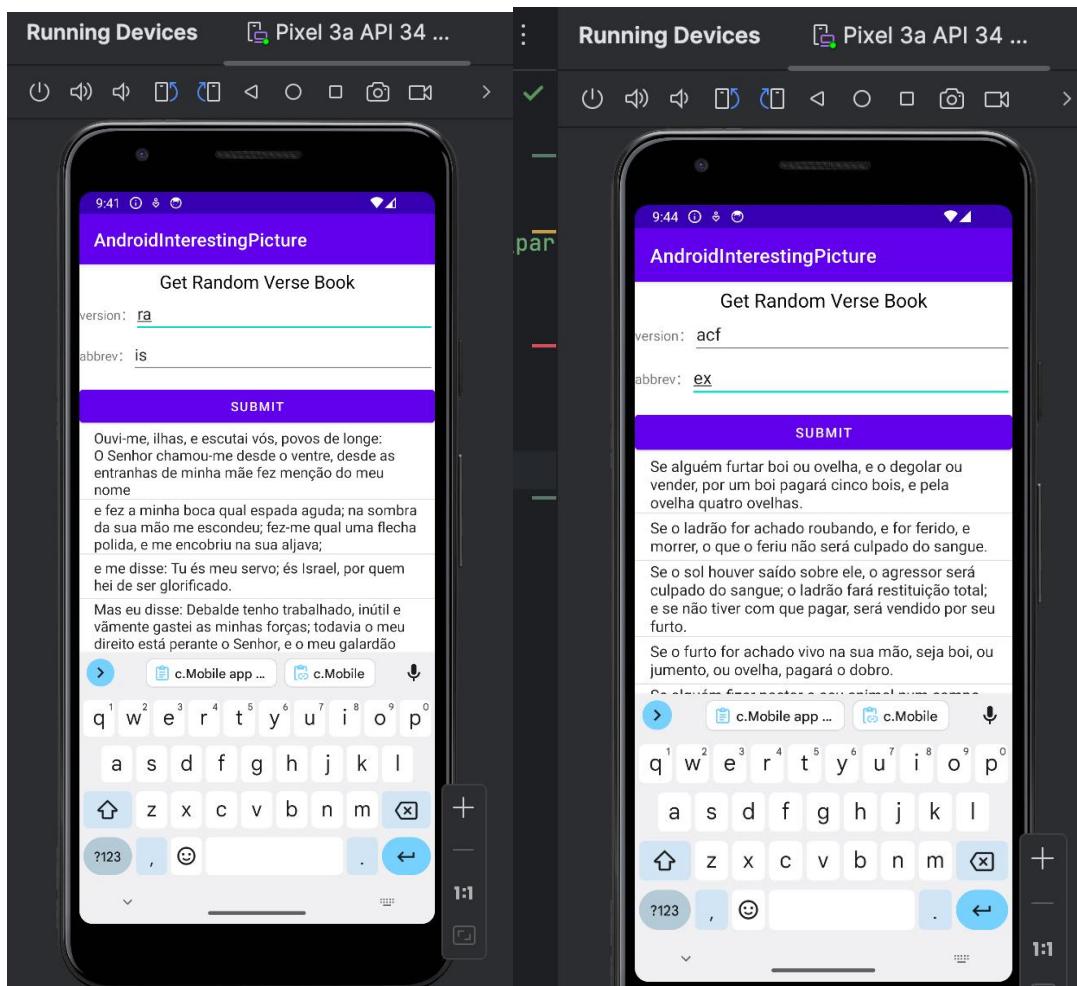
```

113    lic void run() {
114        HttpURLConnection connection = null;
115        try {
116            //Local service -->ip:port/project4task2
117            //git code space --> https://vigilant-pancake-jjjxq75965w7fjpjr-8080.app.github.dev
118            URL url = new URL(spec: "https://solid-meme-4jjxwv764jrw276j6-8080.app.github.dev/abibliadigitalApi?version");
119            connection = (HttpURLConnection) url.openConnection();
120            connection.setRequestMethod("GET");
121            InputStream in = connection.getInputStream();
122            BufferedReader reader = new BufferedReader(
123                new InputStreamReader(in));
124            String response = "" + line:

```

Start:





By entering the version name and book name in Version and abbrev, click submit to get the random Bible verse resulting from the api call.