Hybrid Translation Application Document Version 1.0

1. Modules

I. ML-KIT API

Requirements

- **Language Identification**:

  Entered text must be identified.

Generally, those texts will be few words sentence, or just words. They must be identified accurately.

- **Language Translation**:

  Translated result text must be semantically accurate.

  Target text must consist of alphabets of the target language.

- **User Interface:**

  It must be responsive, understandable.

  Source, and target languages can be decided by user, or source language might be auto-identified. Decision must be decidable by user.

  Identification must be done after each key stroke.

  Translation must only be done after button click.

  Reverse translation must be done with reverse button.

  Result and source texts must be readable.

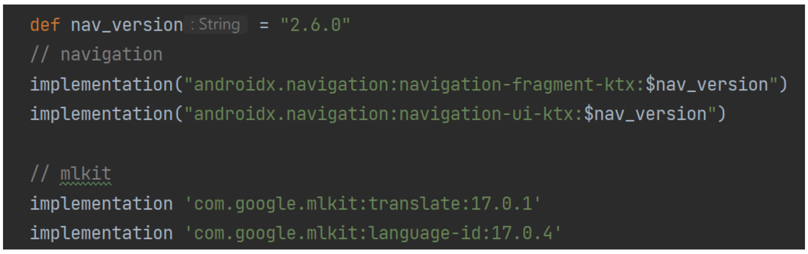
- **Functional:**

  Application must work without internet connection except downloading the models.

  User must be informed if a download is started, and ends (fail or success).

Implementation Details

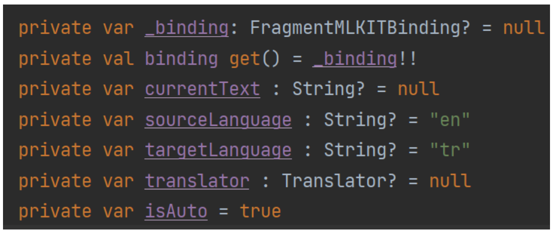
- **Dependencies:**



```
def nav_version : String = "2.6.0"
// navigation
implementation("androidx.navigation:navigation-fragment-ktx:$nav_version")
implementation("androidx.navigation:navigation-ui-ktx:$nav_version")

// mlkit
implementation 'com.google.mlkit:translate:17.0.1'
implementation 'com.google.mlkit:language-id:17.0.4'
```

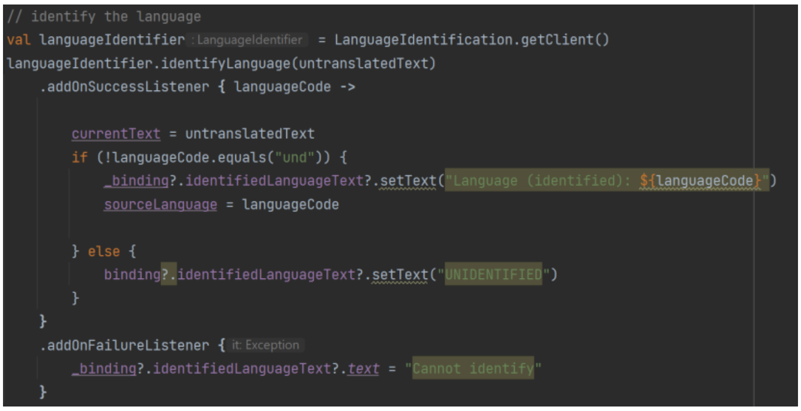
Navigation, and ML-KIT dependencies.

- **ML-KIT methods:**



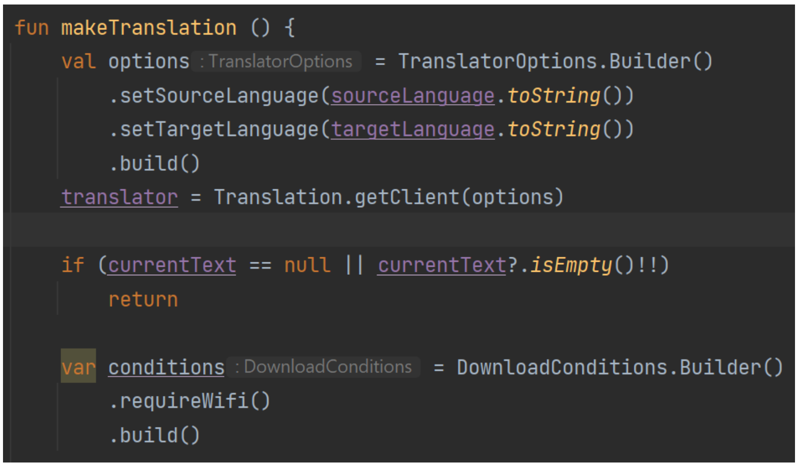
```
private var _binding: FragmentMLKITBinding? = null
private val binding get() = _binding!!
private var currentText : String? = null
private var sourceLanguage : String? = "en"
private var targetLanguage : String? = "tr"
private var translator : Translator? = null
private var isAuto = true
```

Fields of the fragment class. Current text, source, and target languages are held in here. isAuto means, is auto mode on? Translator is the object accessed for translation.



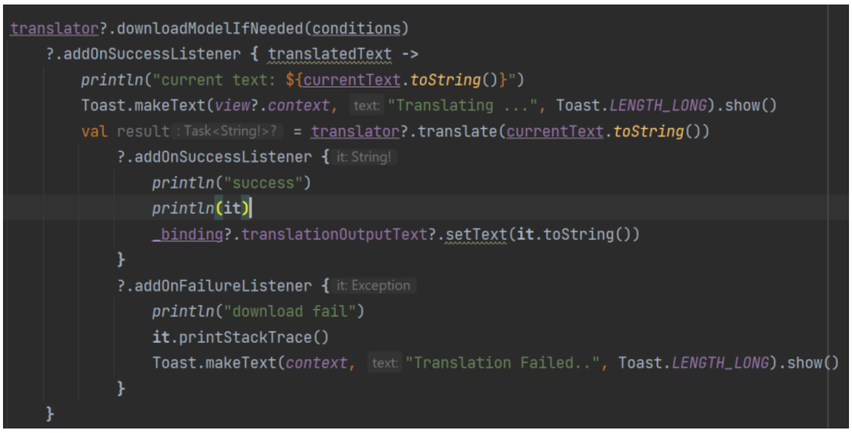
```
// identify the language
val languageIdentifier : LanguageIdentifier  = LanguageIdentification.getClient()
languageIdentifier.identifyLanguage(untranslatedText)
    .addOnSuccessListener { languageCode ->

        currentText = untranslatedText
        if (!languageCode.equals("und")) {
            _binding?.identifiedLanguageText?.setText("Language (identified): ${languageCode}")
            sourceLanguage = languageCode

        } else {
            binding?.identifiedLanguageText?.setText("UNIDENTIFIED")
        }
    }
    .addOnFailureListener { it: Exception
        _binding?.identifiedLanguageText?.text = "Cannot identify"
    }
```

Identifying part inside identifyLanguage() function. Views are accessed with binding object. Calling identifyLanguage is enough.



```
fun makeTranslation () {
    val options : TranslatorOptions  = TranslatorOptions.Builder()
        .setSourceLanguage(sourceLanguage.toString())
        .setTargetLanguage(targetLanguage.toString())
        .build()
    translator = Translation.getClient(options)

    if (currentText == null || currentText?.isEmpty()!!)
        return

    var conditions : DownloadConditions  = DownloadConditions.Builder()
        .requireWifi()
        .build()
```

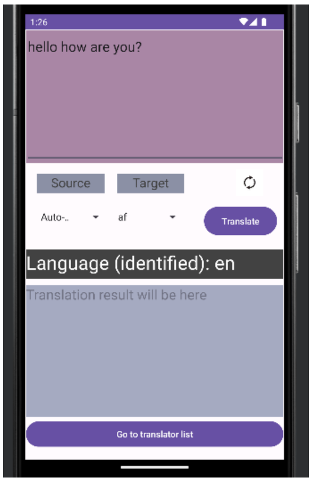
Inside make translation, options are set, condition object is set.



```
translator?.downloadModelIfNeeded(conditions)
    ?.addOnSuccessListener { translatedText ->
        println("current text: ${currentText.toString()}")
        Toast.makeText(view?.context,  text: "Translating ...", Toast.LENGTH_LONG).show()
        val result : Task<String!>?  = translator?.translate(currentText.toString())
            ?.addOnSuccessListener { it: String!
                println("success")
                println(it)
                _binding?.translationOutputText?.setText(it.toString())
            }
            ?.addOnFailureListener { it: Exception
                println("download fail")
                it.printStackTrace()
                Toast.makeText(context,  text: "Translation Failed..", Toast.LENGTH_LONG).show()
            }
    }
```

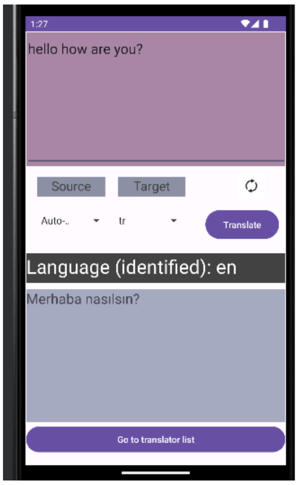
Translator object tries to downloads before, if model is not downloaded background task will try to download it. After that translate method is called from translator object.

Usage


hello how are you?

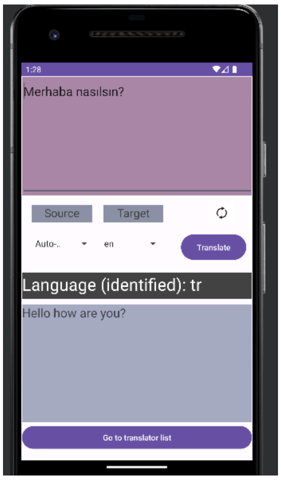
Language (identified): en

Translation result will be here


hello how are you?

Language (identified): en

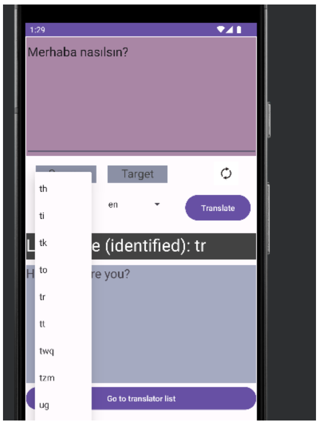
Merhaba nasılsın?


Merhaba nasılsın?

Language (identified): tr

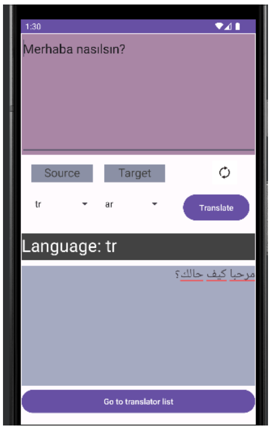
Hello how are you?

Language identified Language translated Reversed


Merhaba nasılsın?

e (identified): tr

re you?


Merhaba nasılsın?

Language: tr

مرحبا كيف حالك؟

Snipper Translation to different alphabet

TODO, and Unchecked Parts

- TODO: Add downloading screen, and ask user if model wants to be uploaded.
- TODO: Fix spinner linear search, search languages in it using hashing. (It is implemented like this in google cloud part.

Requirement Results

Requirement Type	Requirement	MET	AVG MET	NOT MET
Language Identification	Entered text must be identified.		X	
	Generally, texts will be few words sentence, or just words. They must be identified accurately.		X	

Language Translation	Translated result text must be semantically accurate.		X	
	Target text must consist of alphabets of the target language.	X		
User Interface	It must be responsive, understandable.	X		
	Source, and target languages can be decided by user, or source language might be auto-identified. Decision must be decidable by user.	X		
	Identification must be done after each key stroke.	X		
	Translation must only be done after button click.	X		
	Reverse translation must be done with reverse button.	X		
	Result and source texts must be readable.	X		
Functional	Application must work without internet connection except downloading the models.	X		
	User must be informed if a download is started, and ends (fail or success).			X

Future Suggestions

- Text recognition can be added.

II. Google Cloud Translation API

Requirements

- **Language Translation with Glossary:**

  Translation must be done according to the given glossary.

  If source text does not match with glossary, Google’s Neural Machine Translation must be used.

  Translated result text must be semantically accurate.

  Target text must consist of alphabets of the target language.

- **User Interface:**

  It must be responsive, understandable.

  Source, and target languages can be decided by user.

  Translation must only be done after button click.

  There must be loading screen while uploading glossary to project path.

Reverse translation must be done with reverse button.

Result and source texts must be readable.

- **Functional:**

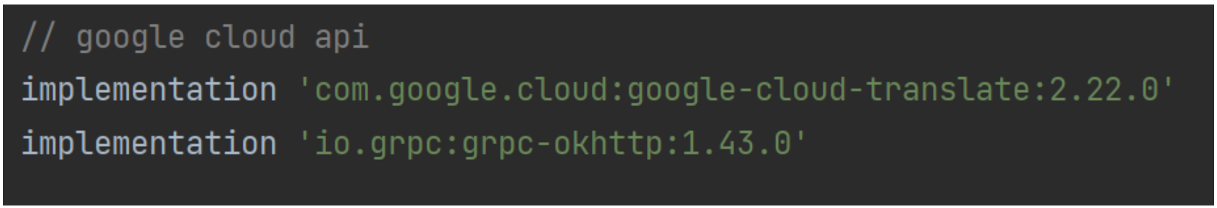
  User must be informed if there is no internet connection.

  While uploading glossary from bucket to project, user must be informed.

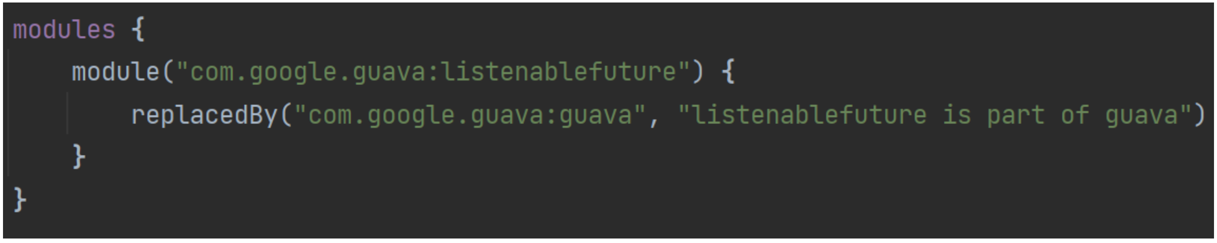
  If glossary upload fails, program must not crash.

Implementation Details

- **Dependencies:**

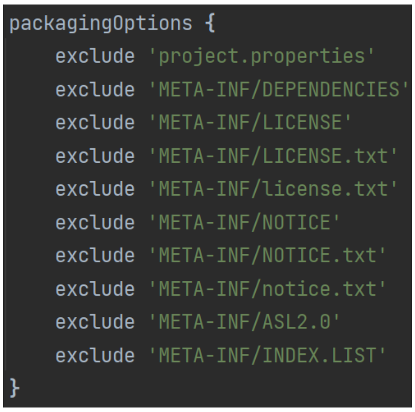


```
// google cloud api
implementation 'com.google.cloud:google-cloud-translate:2.22.0'
implementation 'io.grpc:grpc-okhttp:1.43.0'
```



```
modules {
    module("com.google.guava:listenablefuture") {
        replacedBy("com.google.guava:guava", "listenablefuture is part of guava")
    }
}
```

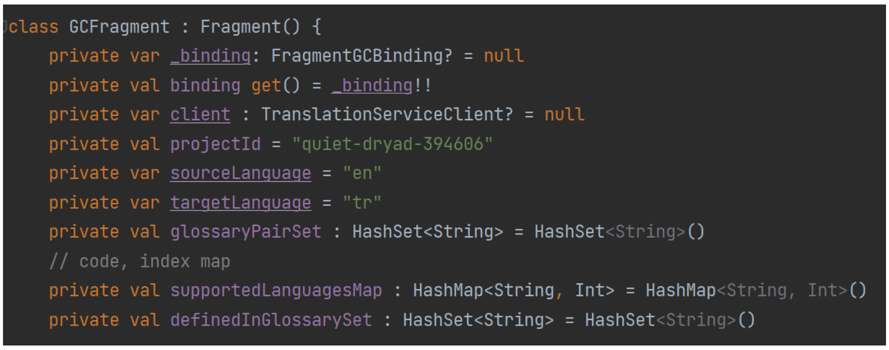
Inside dependencies (build.gradle) those are needed.



```
packagingOptions {
    exclude 'project.properties'
    exclude 'META-INF/DEPENDENCIES'
    exclude 'META-INF/LICENSE'
    exclude 'META-INF/LICENSE.txt'
    exclude 'META-INF/license.txt'
    exclude 'META-INF/NOTICE'
    exclude 'META-INF/NOTICE.txt'
    exclude 'META-INF/notice.txt'
    exclude 'META-INF/ASL2.0'
    exclude 'META-INF/INDEX.LIST'
}
```

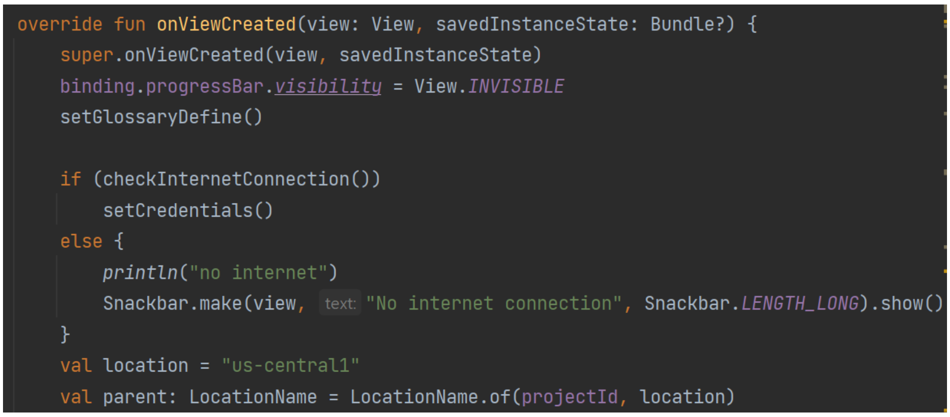
This is needed as packaging options, inside build.gradle (app)

- **Google Cloud Methods Usage:**



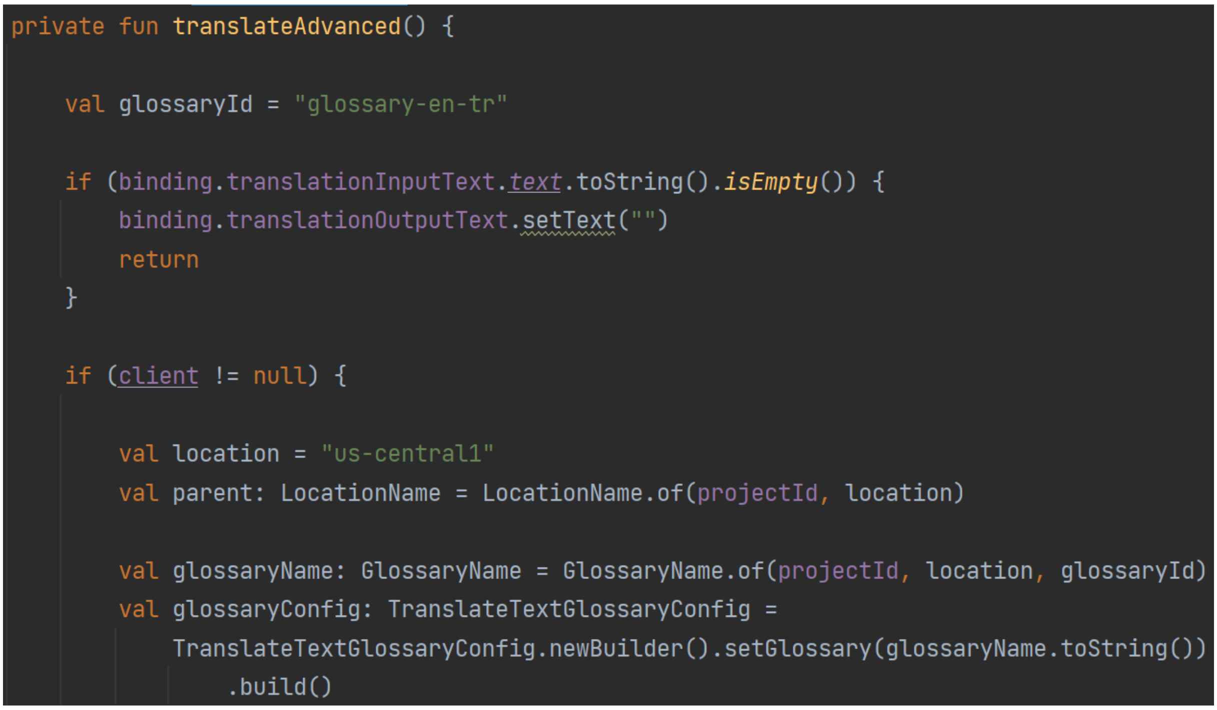
```
class GCFragment : Fragment() {
    private var _binding: FragmentGCBinding? = null
    private val binding get() = _binding!!
    private var client : TranslationServiceClient? = null
    private val projectId = "quiet-dryad-394606"
    private var sourceLanguage = "en"
    private var targetLanguage = "tr"
    private val glossaryPairSet : HashSet<String> = HashSet<String>()
    // code, index map
    private val supportedLanguagesMap : HashMap<String, Int> = HashMap<String, Int>()
    private val definedInGlossarySet : HashSet<String> = HashSet<String>()
```

Class fields. Sets, and map used for spinners in order to make faster search.



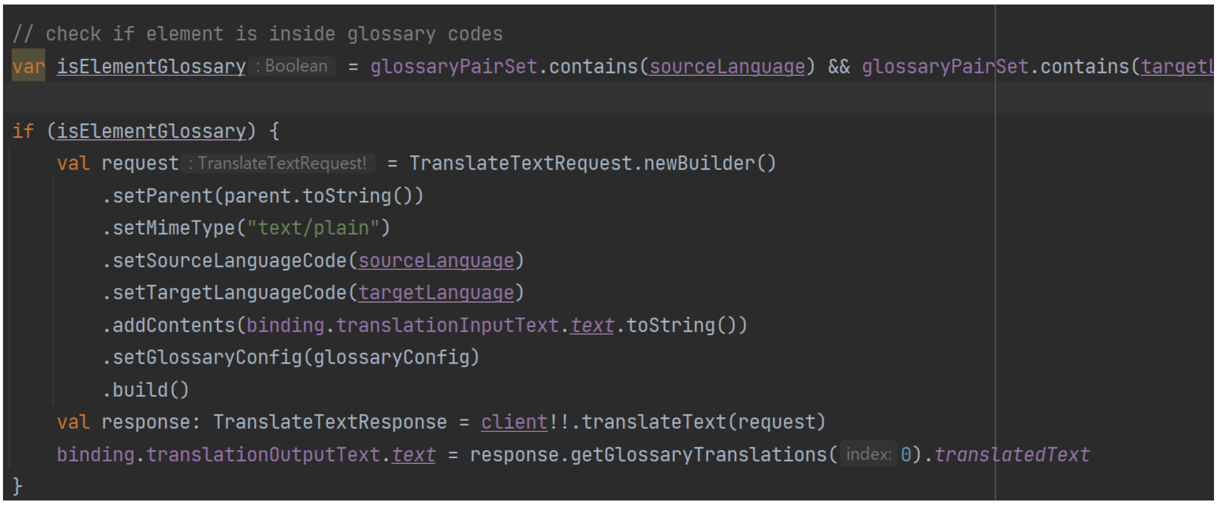
```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    binding.progressBar.visibility = View.INVISIBLE
    setGlossaryDefine()

    if (checkInternetConnection())
        setCredentials()
    else {
        println("no internet")
        Snackbar.make(view, text: "No internet connection", Snackbar.LENGTH_LONG).show()
    }
    val location = "us-central1"
    val parent: LocationName = LocationName.of(projectId, location)
```

Firstly, check internet connection and set location with parent. It must be “us-central1”.



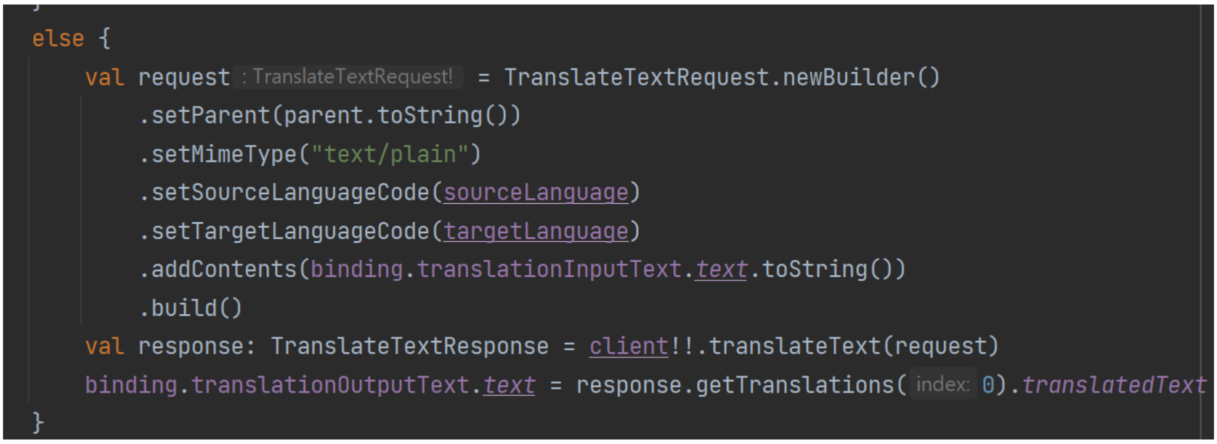
```
private fun translateAdvanced() {

    val glossaryId = "glossary-en-tr"

    if (binding.translationInputText.text.toString().isEmpty()) {
        binding.translationOutputText.setText("")
        return
    }

    if (client != null) {

        val location = "us-central1"
        val parent: LocationName = LocationName.of(projectId, location)

        val glossaryName: GlossaryName = GlossaryName.of(projectId, location, glossaryId)
        val glossaryConfig: TranslateTextGlossaryConfig =
            TranslateTextGlossaryConfig.newBuilder().setGlossary(glossaryName.toString())
                .build()
```

Translation function, set location, parent, glossary ID. Then create a glossary name, and configuration.



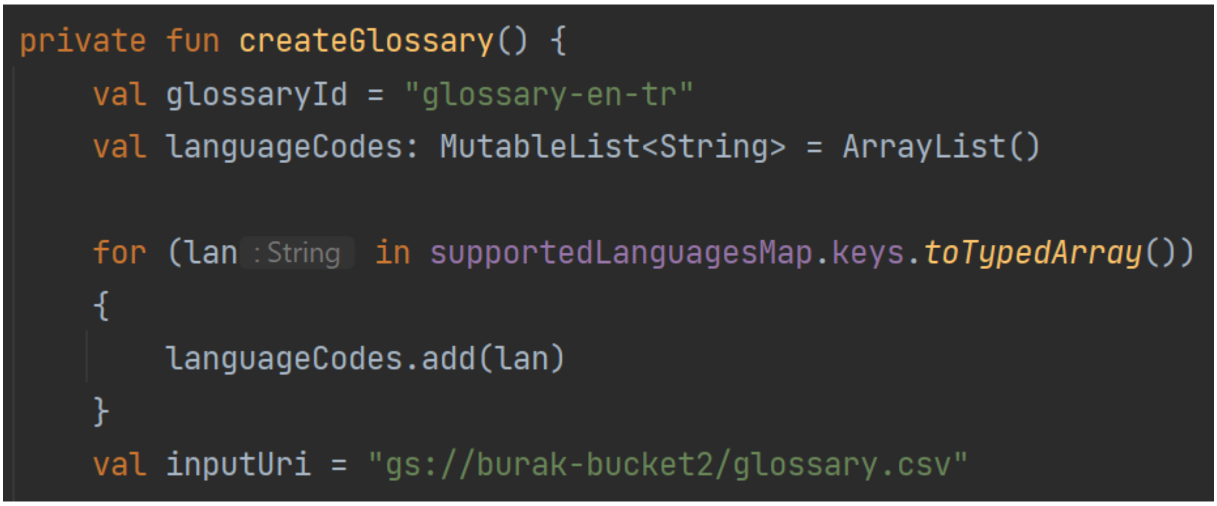
```kotlin
// check if element is inside glossary codes
var isElementGlossary : Boolean = glossaryPairSet.contains(sourceLanguage) && glossaryPairSet.contains(targetL

if (isElementGlossary) {
    val request : TranslateTextRequest! = TranslateTextRequest.newBuilder()
        .setParent(parent.toString())
        .setMimeType("text/plain")
        .setSourceLanguageCode(sourceLanguage)
        .setTargetLanguageCode(targetLanguage)
        .addContents(binding.translationInputText.text.toString())
        .setGlossaryConfig(glossaryConfig)
        .build()
    val response: TranslateTextResponse = client!!.translateText(request)
    binding.translationOutputText.text = response.getGlossaryTranslations( index: 0).translatedText
}
```

If element is in glossary codes, make glossary translation.



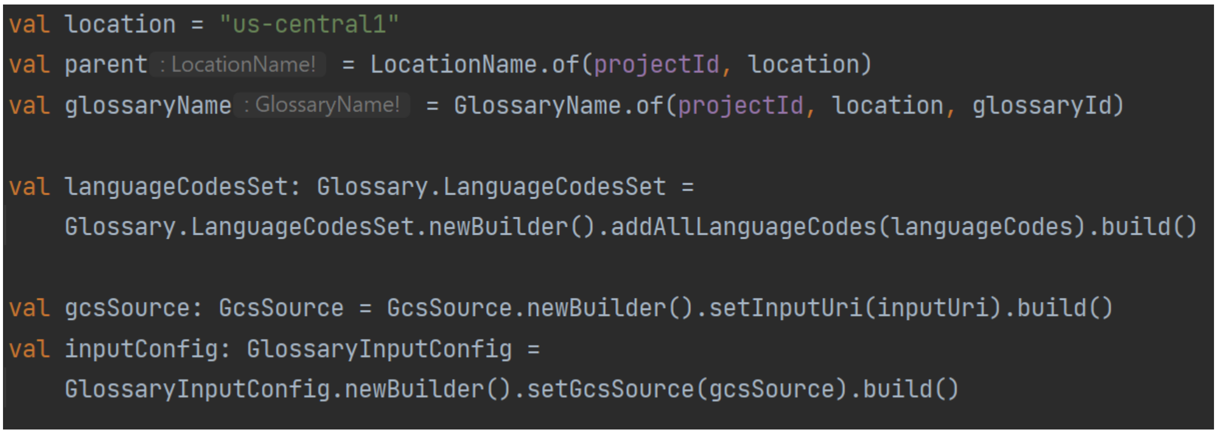
```kotlin
else {
    val request : TranslateTextRequest! = TranslateTextRequest.newBuilder()
        .setParent(parent.toString())
        .setMimeType("text/plain")
        .setSourceLanguageCode(sourceLanguage)
        .setTargetLanguageCode(targetLanguage)
        .addContents(binding.translationInputText.text.toString())
        .build()
    val response: TranslateTextResponse = client!!.translateText(request)
    binding.translationOutputText.text = response.getTranslations( index: 0).translatedText
}
```

If not, make normal translation.

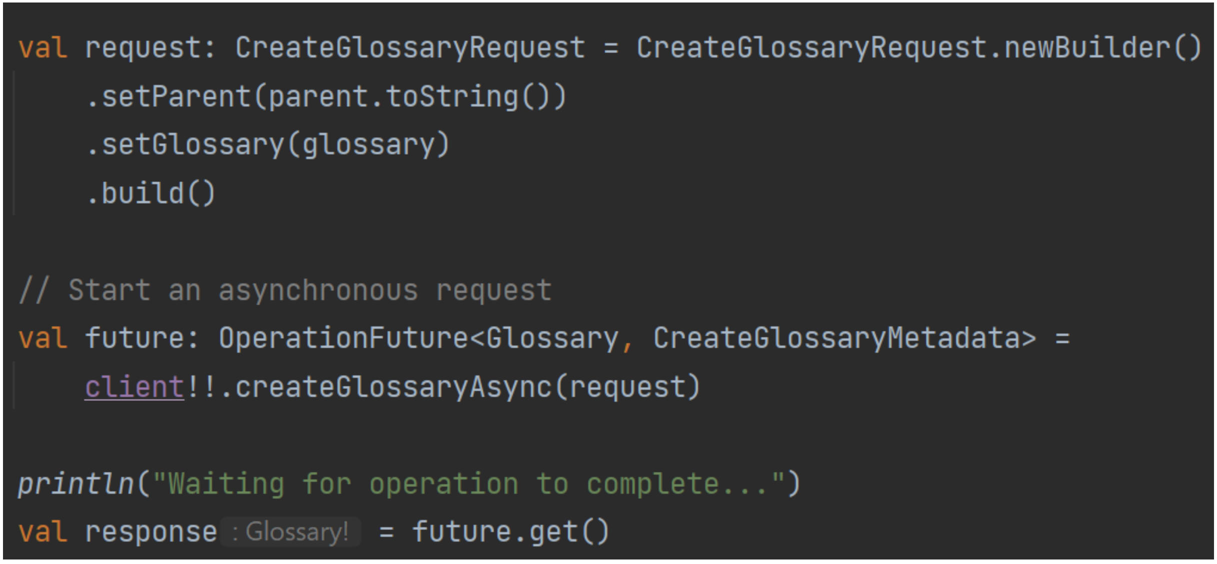


```kotlin
private fun createGlossary() {
    val glossaryId = "glossary-en-tr"
    val languageCodes: MutableList<String> = ArrayList()

    for (lan : String  in supportedLanguagesMap.keys.toTypedArray())
    {
        languageCodes.add(lan)
    }
    val inputUri = "gs://burak-bucket2/glossary.csv"
```

input uri must be glossary in bucket.

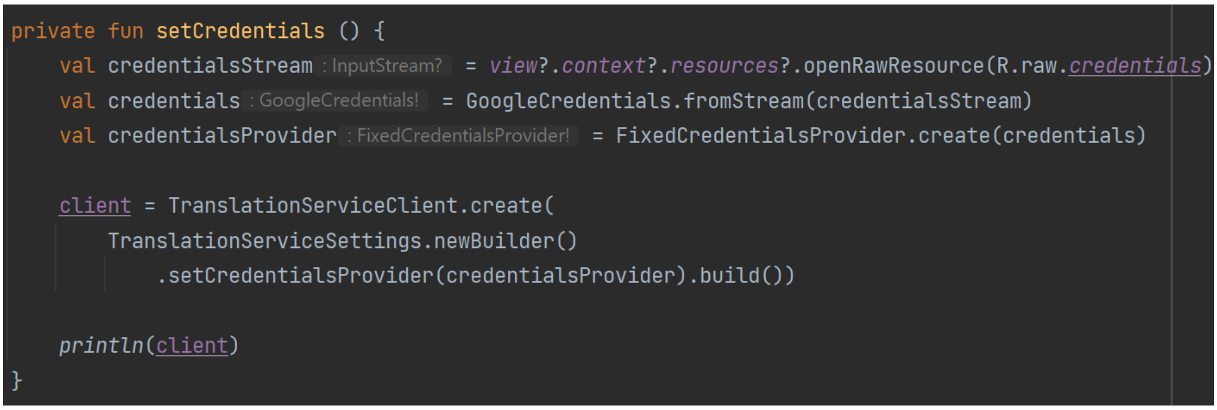


```
val location = "us-central1"
val parent : LocationName!  = LocationName.of(projectId, location)
val glossaryName : GlossaryName!  = GlossaryName.of(projectId, location, glossaryId)

val languageCodesSet: Glossary.LanguageCodesSet =
    Glossary.LanguageCodesSet.newBuilder().addAllLanguageCodes(languageCodes).build()

val gcsSource: GcsSource = GcsSource.newBuilder().setInputUri(inputUri).build()
val inputConfig: GlossaryInputConfig =
    GlossaryInputConfig.newBuilder().setGcsSource(gcsSource).build()
```



```
val request: CreateGlossaryRequest = CreateGlossaryRequest.newBuilder()
    .setParent(parent.toString())
    .setGlossary(glossary)
    .build()

// Start an asynchronous request
val future: OperationFuture<Glossary, CreateGlossaryMetadata> =
    client!!.createGlossaryAsync(request)

println("Waiting for operation to complete...")
val response : Glossary!  = future.get()
```

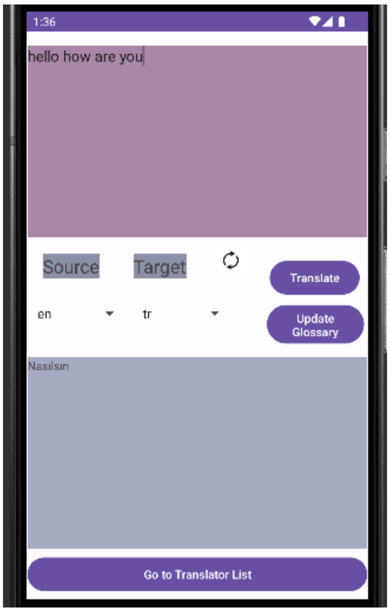
Set location, glossary, and configuration. Then send the request. It must be background process, because of the freezing problem.



```
private fun setCredentials () {
    val credentialsStream : InputStream?  = view?.context?.resources?.openRawResource(R.raw.credentials)
    val credentials : GoogleCredentials!  = GoogleCredentials.fromStream(credentialsStream)
    val credentialsProvider : FixedCredentialsProvider!  = FixedCredentialsProvider.create(credentials)

    client = TranslationServiceClient.create(
        TranslationServiceSettings.newBuilder()
            .setCredentialsProvider(credentialsProvider).build())

    println(client)
}
```

Credentials must be set. It reads it from raw directory.

Usage

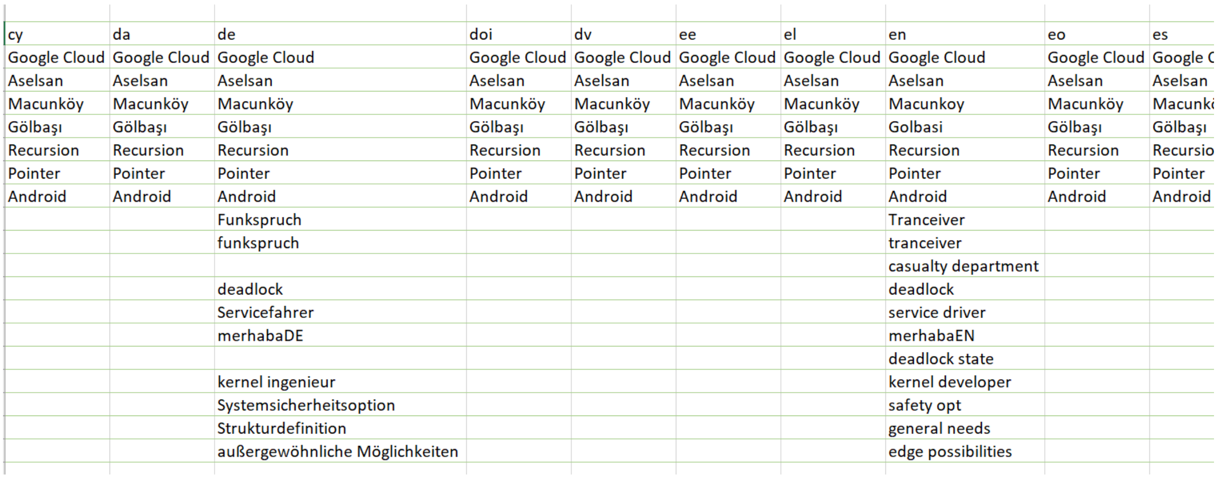


hello how are you

en → tr

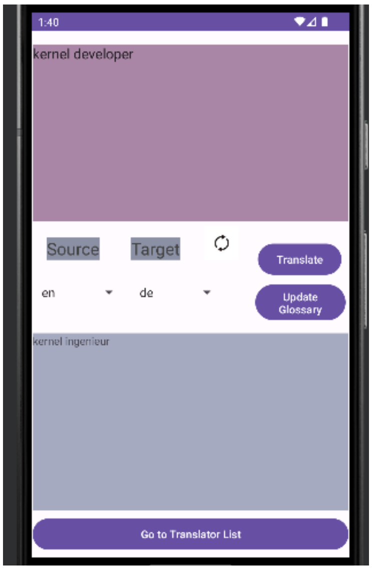
Nasılsın

Regular translation



cy	da	de	doi	dv	ee	el	en	eo	es
Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google C
Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan
Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunkoy	Macunköy	Macunk
Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Golbasi	Gölbaşı	Gölbaşı
Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursio
Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer
Android	Android	Android	Android	Android	Android	Android	Android	Android	Android
		Funkspruch					Tranceiver		
		funkspruch					tranceiver		
							casualty department		
		deadlock					deadlock		
		Servicefahrer					service driver		
		merhabaDE					merhabaEN		
							deadlock state		
		kernel ingenieur					kernel developer		
		Systemsicherheitsoption					safety opt		
		Strukturdefinition					general needs		
		außergewöhnliche Möglichkeiten					edge possibilities		

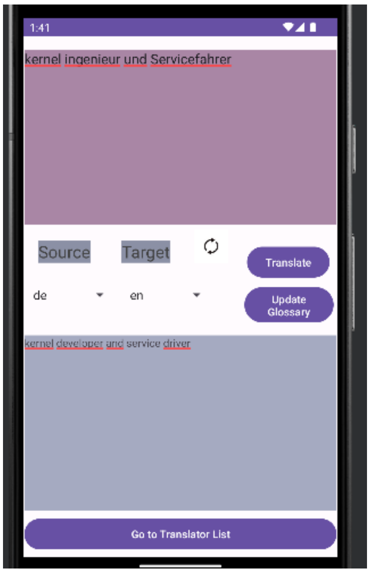
Some part of the current glossary



kernel developer

en → de

kernel ingenieur

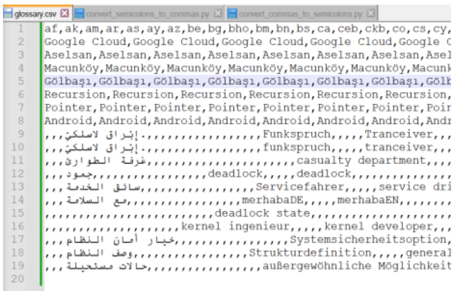


kernel ingenieur und Servicefahrer

de → en

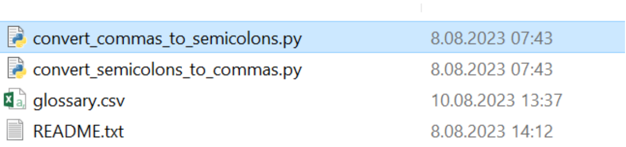
kernel developer und service driver

It is translates according to glossary

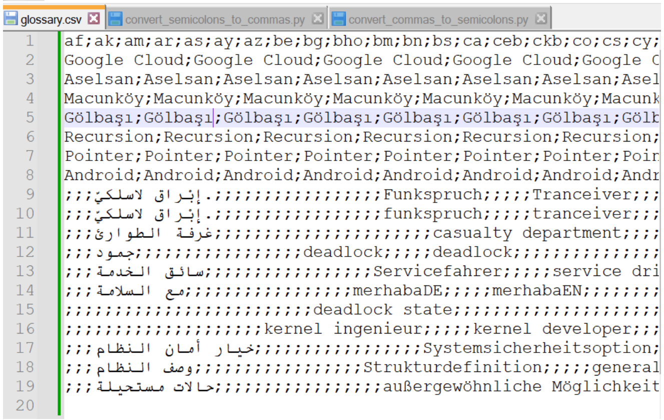


glossary.csv
convert_semicolons_to_commas.py
convert_commas_to_semicolons.py

```
af,ak,am,ar,as,ay,az,be,bg,bho,bm,bn,bs,ca,ceb,ckb,co,cs,cy,
Google Cloud,Google Cloud,Google Cloud,Google Cloud,Google C
Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Asel
Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunk
Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölb
Recursion,Recursion,Recursion,Recursion,Recursion,Recursion,
Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Poir
Android,Android,Android,Android,Android,Android,Android,Andr
,,,إبْراق لاسلكيّ.,,,,,,,,,,,,,,,,,Funkspruch,,,,Tranceiver,,
,,,إبْراق لاسلكيّ.,,,,,,,,,,,,,,,,,funkspruch,,,,tranceiver,,
,,,غرفة الطوارئ,,,,,,,,,,,,,,,,,,,,,,casualty department,,,
,,,جمود,,,,,,,,,,,,,,,,,deadlock,,,,deadlock,,,,,,,,,,,,,,,
,,,سائق الخدمة,,,,,,,,,,,,,,,,,Servicefahrer,,,,service dri
,,,مع السلامة,,,,,,,,,,,,,,,,,merhabaDE,,,,merhabaEN,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,deadlock state,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,kernel ingenieur,,,,kernel developer,,
,,,خيار أمان النظام,,,,,,,,,,,,,,,,,Systemsicherheitsoption,
,,,وصف النظام,,,,,,,,,,,,,,,,,Strukturdefinition,,,,general
,,,حالات مستحيلة,,,,,,,,,,,,,,,,,außergewöhnliche Möglichkeit
```



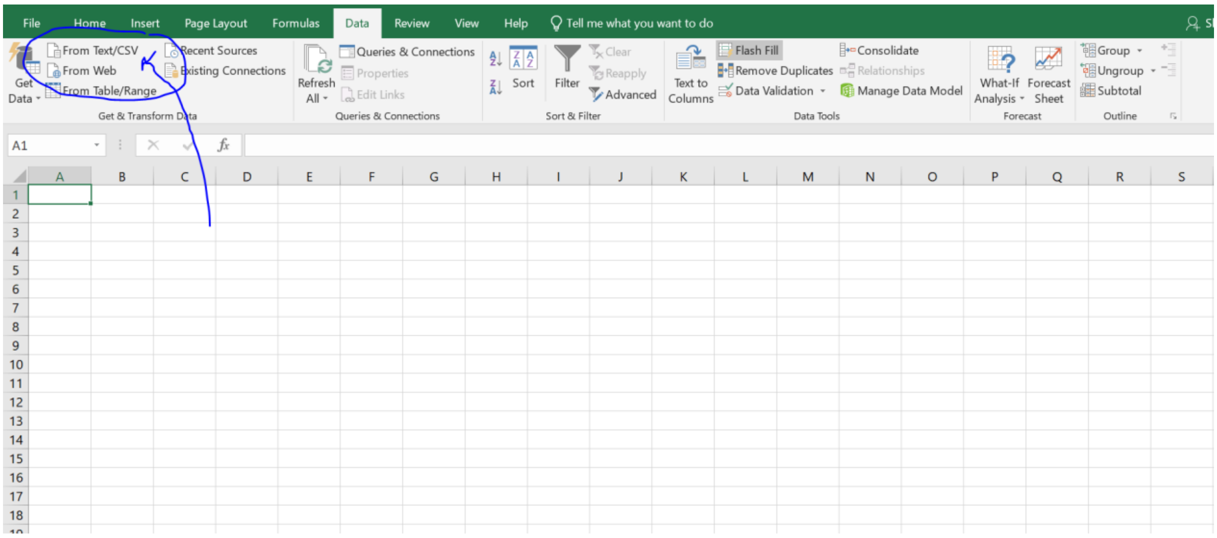
convert_commas_to_semicolons.py	8.08.2023 07:43
convert_semicolons_to_commas.py	8.08.2023 07:43
glossary.csv	10.08.2023 13:37
README.txt	8.08.2023 14:12



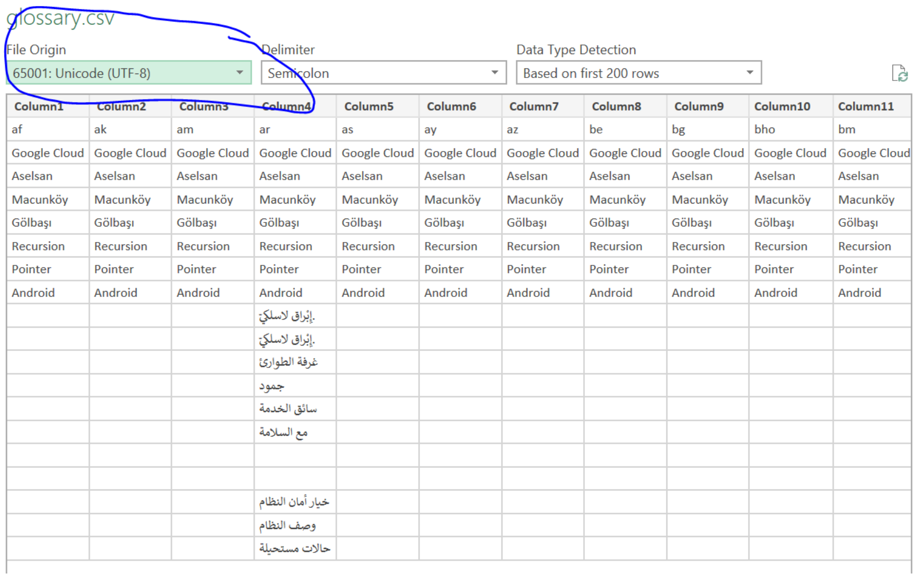
glossary.csv
convert_semicolons_to_commas.py
convert_commas_to_semicolons.py

```
af;ak;am;ar;as;ay;az;be;bg;bho;bm;bn;bs;ca;ceb;ckb;co;cs;cy;
Google Cloud;Google Cloud;Google Cloud;Google Cloud;Google C
Aselsan;Aselsan;Aselsan;Aselsan;Aselsan;Aselsan;Aselsan;Asel
Macunköy;Macunköy;Macunköy;Macunköy;Macunköy;Macunköy;Macunk
Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölb
Recursion;Recursion;Recursion;Recursion;Recursion;Recursion;
Pointer;Pointer;Pointer;Pointer;Pointer;Pointer;Pointer;Poir
Android;Android;Android;Android;Android;Android;Android;Andr
;;;إبْراق لاسلكيّ.;;;;;;;;;;;;;;;;;Funkspruch;;;;Tranceiver;;;
;;;إبْراق لاسلكيّ.;;;;;;;;;;;;;;;;;funkspruch;;;;tranceiver;;;
;;;غرفة الطوارئ;;;;;;;;;;;;;;;;;;;;;;casualty department;;;;
;;;جمود;;;;;;;;;;;;;;;;;deadlock;;;;deadlock;;;;;;;;;;;;;;;
;;;سائق الخدمة;;;;;;;;;;;;;;;;;Servicefahrer;;;;service dri
;;;مع السلامة;;;;;;;;;;;;;;;;;merhabaDE;;;;merhabaEN;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;deadlock state;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;kernel ingenieur;;;;kernel developer;;;
;;;خيار أمان النظام;;;;;;;;;;;;;;;;;Systemsicherheitsoption;
;;;وصف النظام;;;;;;;;;;;;;;;;;Strukturdefinition;;;;general
;;;حالات مستحيلة;;;;;;;;;;;;;;;;;außergewöhnliche Möglichkeit
```

To open it on excel commas must be converted to semicolon.



Open blank excel, and click this button.



glossary.csv

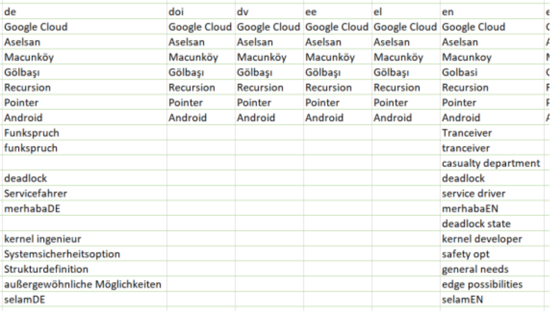
File Origin: 65001: Unicode (UTF-8)

Delimiter: Semicolon

Data Type Detection: Based on first 200 rows

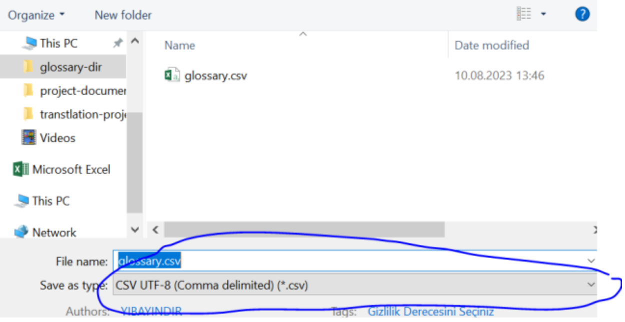
Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11
af	ak	am	ar	as	ay	az	be	bg	bho	bm
Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud
Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan
Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy
Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı
Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion
Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer
Android	Android	Android	Android	Android	Android	Android	Android	Android	Android	Android
			إبْراق لاسلكيّ.							
			إبْراق لاسلكيّ.							
			غرفة الطوارئ							
			جمود							
			سائق الخدمة							
			مع السلامة							
			خيار أمان النظام							
			وصف النظام							
			حالات مستحيلة							

Choose UTF-8 encoding.



de	doi	dv	ee	el	en
Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud
Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan
Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunkoy
Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Golbasi
Recursion	Recursion	Recursion	Recursion	Recursion	Recursion
Pointer	Pointer	Pointer	Pointer	Pointer	Pointer
Android	Android	Android	Android	Android	Android
Funkspruch					Tranceiver
funkspruch					tranceiver
					casualty department
deadlock					deadlock
Servicefahrer					service driver
merhabaDE					merhabaEN
					deadlock state
kernel ingenieur					kernel developer
Systemsicherheitsoption					safety opt
Strukturdefinition					general needs
außergewöhnliche Möglichkeiten					edge possibilities
selamDE					selamEN

selamDE, selamEN, selamTR, selamAR, are added.



Name	Date modified
glossary.csv	10.08.2023 13:46

File name: glossary.csv

Save as type: CSV UTF-8 (Comma delimited) (*.csv)

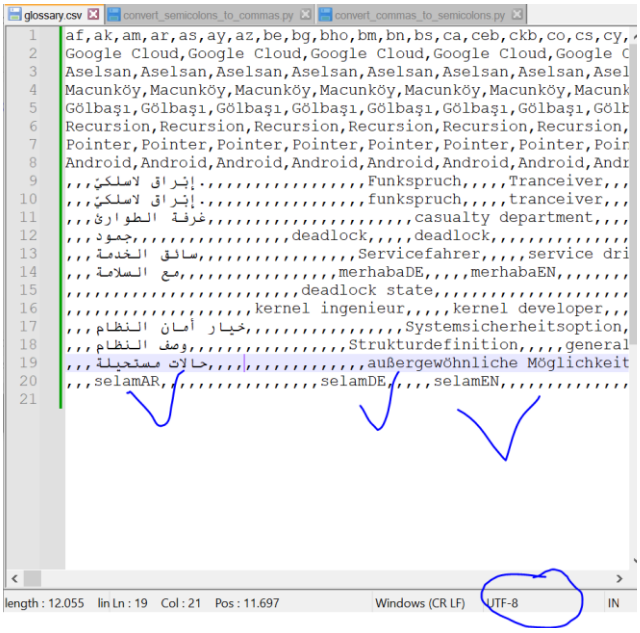
Authors: YIBAYINDIR

Tags: Gizlilik Derecesini Seçiniz

Save the file in this format.

Close the excel.

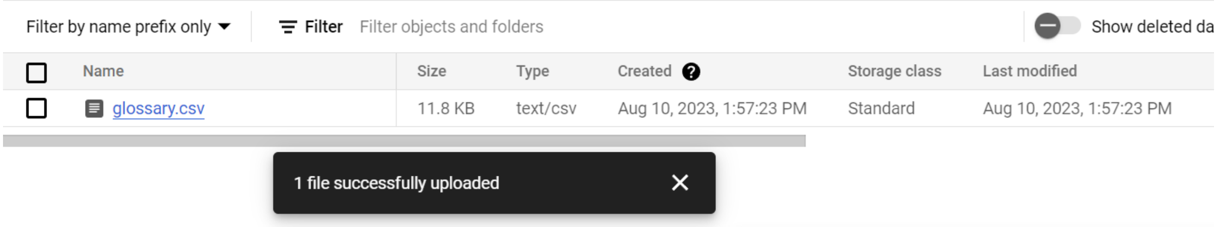
Run semicolon to comma script.


glossary.csv
convert_semicolons_to_commas.py
convert_commas_to_semicolons.py

```
af,ak,am,ar,as,ay,az,be,bg,bho,bm,bn,bs,ca,ceb,ckb,co,cs,cy,
Google Cloud,Google Cloud,Google Cloud,Google Cloud,Google C
Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Asel
Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunk
Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölb
Recursion,Recursion,Recursion,Recursion,Recursion,Recursion,
Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Poin
Android,Android,Android,Android,Android,Android,Android,Andr
,,,إبراق لاسلكيّ.,,,,,,,,,,,,,,,,,,Funkspruch,,,,,Tranceiver,,,
,,,إبراق لاسلكيّ.,,,,,,,,,,,,,,,,,,funkspruch,,,,,tranceiver,,,
,,,غرفة الطوارئ,,,,,,,,,,,,,,,,,,,,,,casualty department,,,,
,,,جمود,,,,,,,,,,,,,,,,,,deadlock,,,,,deadlock,,,,,,,,,,,,,,
,,,سائق الخدمة,,,,,,,,,,,,,,,,,,Servicefahrer,,,,,service dri
,,,مع السلامة,,,,,,,,,,,,,,,,,,merhabaDE,,,,,merhabaEN,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,deadlock state,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,kernel ingenieur,,,,,kernel developer,,,
,,,خيار أمان النظام,,,,,,,,,,,,,,,,,,Systemsicherheitsoption,
,,,وصف النظام,,,,,,,,,,,,,,,,,,Strukturdefinition,,,,,general
,,,حالات مستحيلة,,,,,,,,,,,,,,,,,,außergewöhnliche Möglichkeit
,,,selamAR,,,,,,,,,,,,,,,,,,selamDE,,,,,selamEN,,,,,,,,,,,,,,
```

length : 12.055 lin Ln : 19 Col : 21 Pos : 11.697 Windows (CR LF) UTF-8 IN

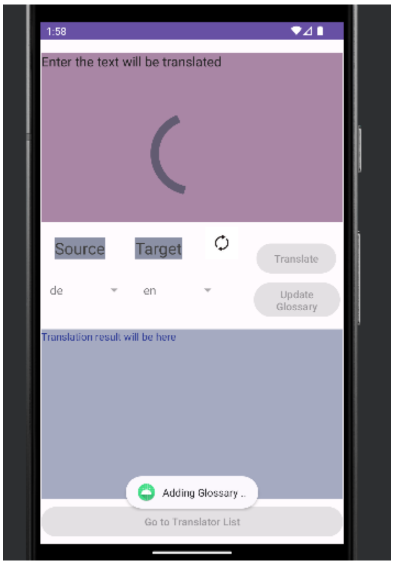
After running, it must be UTF-8. Without BOM.



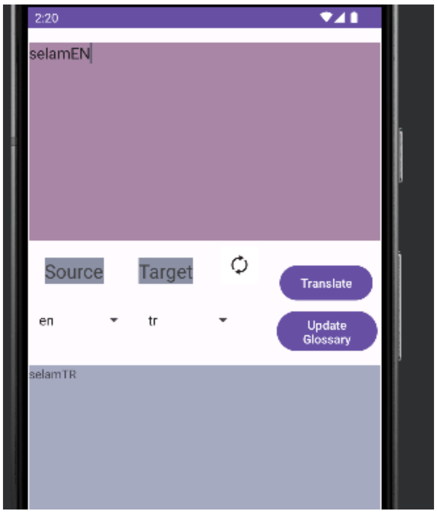
□	Name	Size	Type	Created	Storage class	Last modified
□	glossary.csv	11.8 KB	text/csv	Aug 10, 2023, 1:57:23 PM	Standard	Aug 10, 2023, 1:57:23 PM

1 file successfully uploaded ✕

Upload glossary to bucket.



Click update glossary button.


selamEN

selamTR

Glossary is working

TODO, and Unchecked Parts

- TODO: In hybrid project, if there is no internet connection program freezes when cloud is clicked. Fix it.
- TODO: When csv fails, or credential fails, program crashes. Handle them appropriately.
- TODO: If target and source is same program crashes fix it.

Requirement Results

Requirement Type	Requirement	MET	AVG MET	NOT MET
Language Translation with Glossary	Translation must be done according to the given glossary.	X		
	If source text does not match with glossary, Google's Neural Machine Translation must be used.	X		
	Translated result text must be semantically accurate.		X	
	Target text must consist of alphabets of the target language.	X		
User Interface	It must be responsive, understandable.	X		
	Source, and target languages can be decided by user.	X		
	Translation must only be done after button click.	X		
	Reverse translation must be done with reverse button.	X		
	Result and source texts must be readable.	X		

	There must be loading screen while uploading glossary to project path.	X		
Functional	User must be informed if there is no internet connection.		X	
	While uploading glossary from bucket to project, user must be informed.	X		
	If glossary upload fails, program must not crash.			X

Future Suggestions

- AutoML can be added in order to use custom model.

III. Advanced ML-KIT API

Requirements

- **Language Translation with Glossary:**

  Translation must be done according to the given glossary, but it is not required to detect glossary words inside input text like Google Glossary. Direct translation is enough.

  If source text does not match with glossary, ML-KIT model must be used.

  Translated result text must be semantically accurate.

  Target text must consist of alphabets of the target language.

- **Language Identification:**

  Entered text must be identified.

  Generally, those texts will be few words sentence or just word. They must be identified accurately.

- **User Interface:**

  It must be responsive, understandable.

  Source, and target languages can be decided by user, or source language might be auto-identified. Decision must be decidable by user.

  Identification must be done after each key stroke.

  Translation must only be done after button click.

  Reverse translation must be done with reverse button.

  Result and source texts must be readable.

  There must be update button to load glossary to database.

  User must see loading screen while database is updating.

- **Functional:**

  Application must work without internet connection except downloading the models.

  User must be informed if a download is started, and ends (fail or success).

  User must be informed if update fails.

  Program must not crash if csv file is in wrong format, and user must be informed.

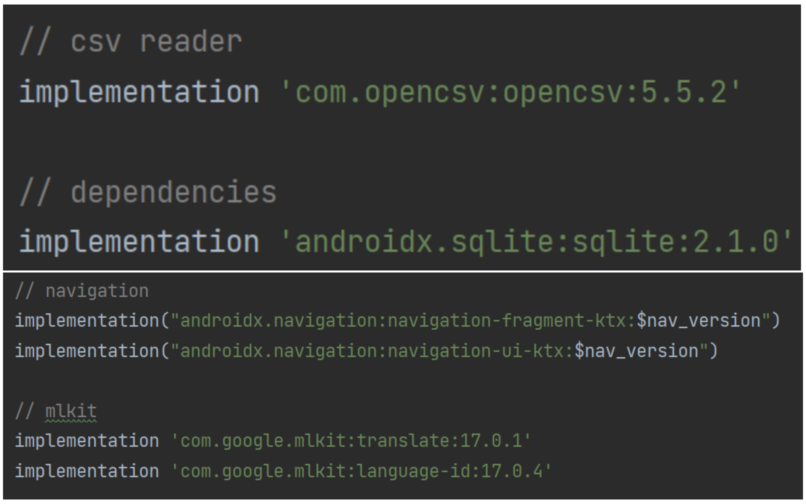
  Program must be able to parse comma separated, utf-8 format csv files.

  Database update, and queries must not take too much time.

  Glossaries must not be held in process memory, if data structure implementation won’t be used.

Implementation Details

- **Dependencies:**

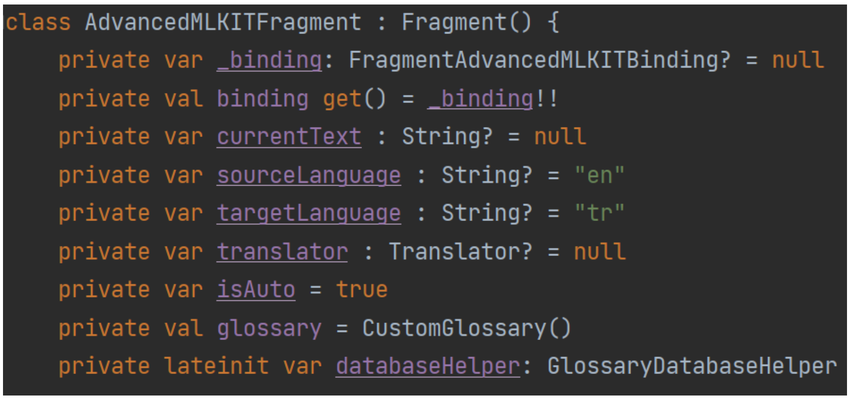


```
// csv reader
implementation 'com.opencsv:opencsv:5.5.2'

// dependencies
implementation 'androidx.sqlite:sqlite:2.1.0'
```

```
// navigation
implementation("androidx.navigation:navigation-fragment-ktx:$nav_version")
implementation("androidx.navigation:navigation-ui-ktx:$nav_version")

// mlkit
implementation 'com.google.mlkit:translate:17.0.1'
implementation 'com.google.mlkit:language-id:17.0.4'
```

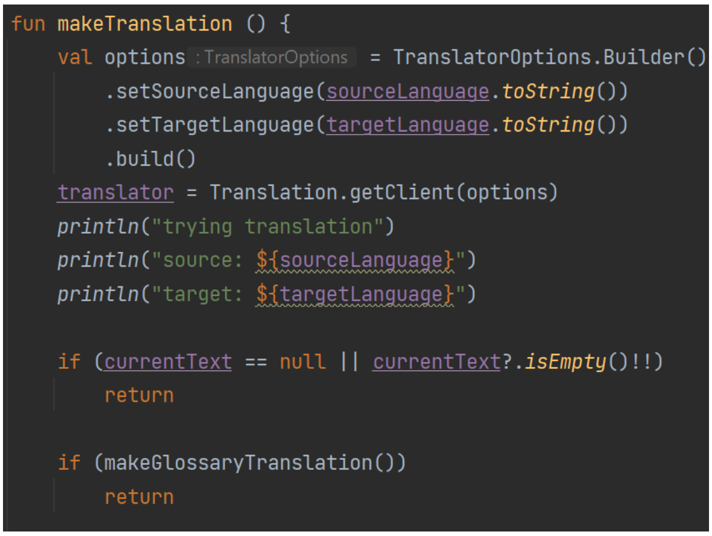
Dependencies for, csv reader, sqlite, ml-kit, and navigation.

- **Advanced ML-KIT methods:**



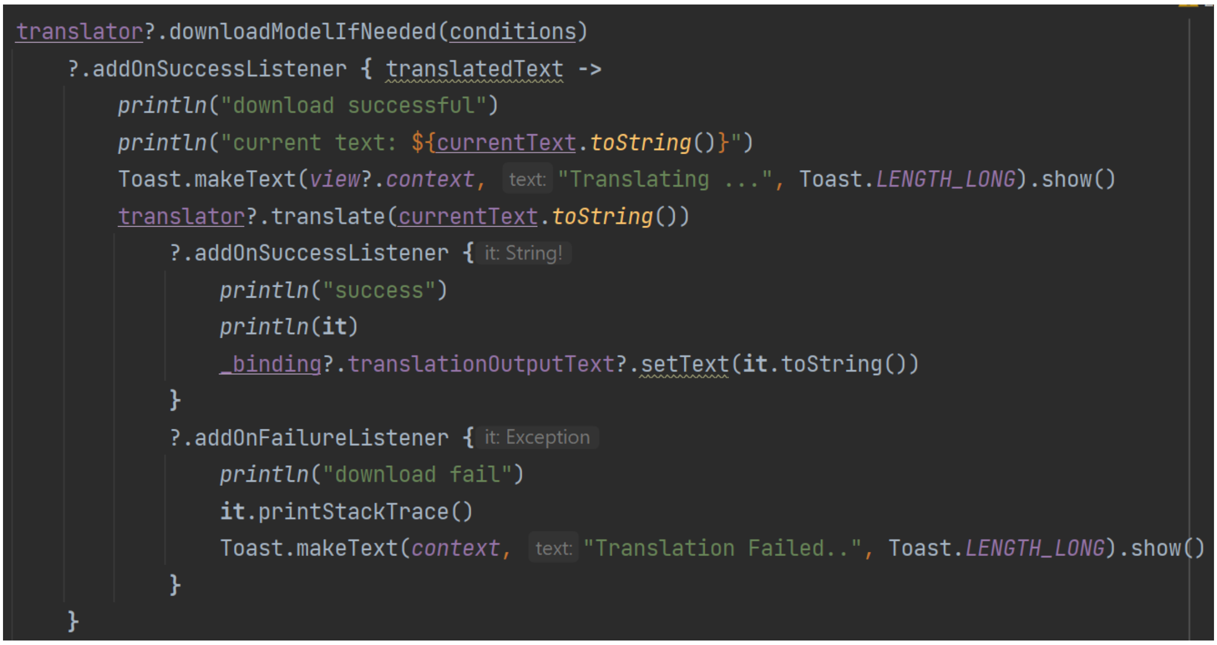
```kotlin
class AdvancedMLKITFragment : Fragment() {
    private var _binding: FragmentAdvancedMLKITBinding? = null
    private val binding get() = _binding!!
    private var currentText : String? = null
    private var sourceLanguage : String? = "en"
    private var targetLanguage : String? = "tr"
    private var translator : Translator? = null
    private var isAuto = true
    private val glossary = CustomGlossary()
    private lateinit var databaseHelper: GlossaryDatabaseHelper
```

Glossary object is for data structure, database helper is for database. Comment one of them if it is not used.



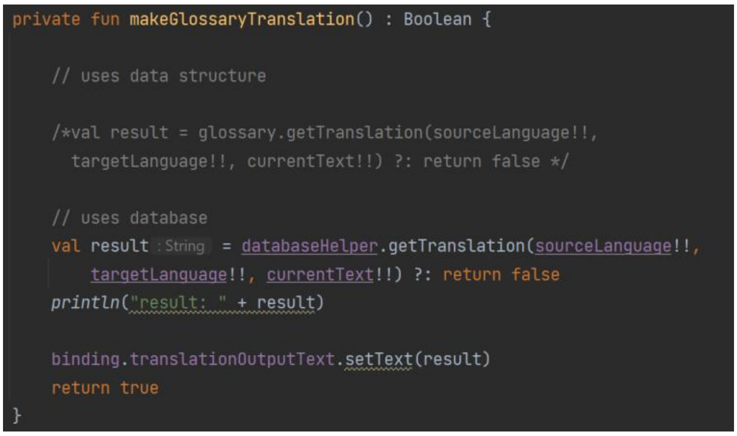
```kotlin
fun makeTranslation () {
    val options : TranslatorOptions  = TranslatorOptions.Builder()
        .setSourceLanguage(sourceLanguage.toString())
        .setTargetLanguage(targetLanguage.toString())
        .build()
    translator = Translation.getClient(options)
    println("trying translation")
    println("source: ${sourceLanguage}")
    println("target: ${targetLanguage}")

    if (currentText == null || currentText?.isEmpty()!!)
        return

    if (makeGlossaryTranslation())
        return
```

If glossary translation returns true, no need to enter ML-KIT translation.



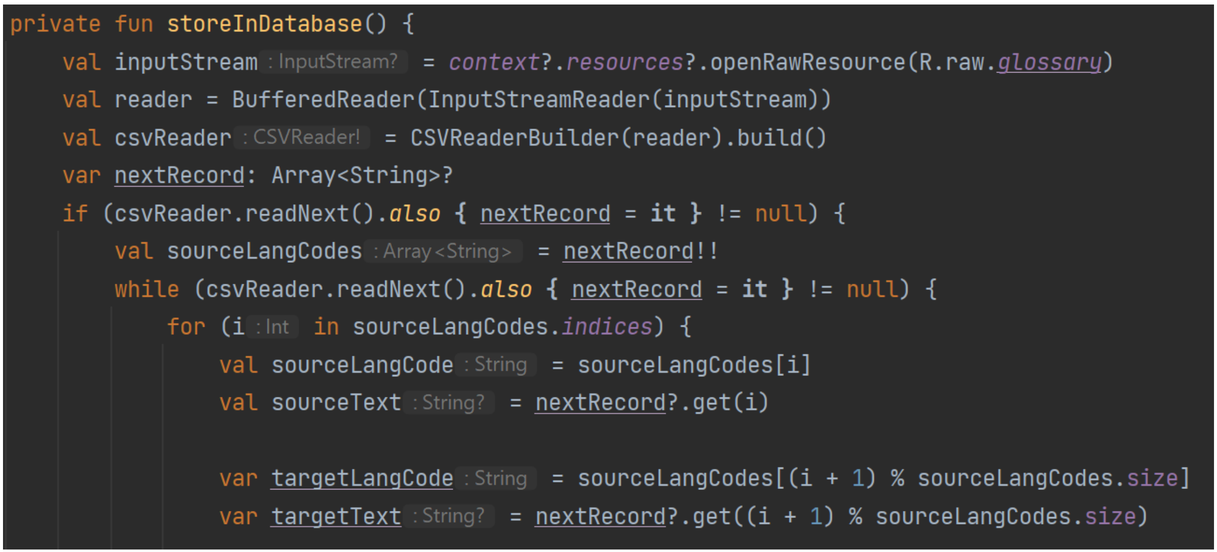
```kotlin
translator?.downloadModelIfNeeded(conditions)
    ?.addOnSuccessListener { translatedText ->
        println("download successful")
        println("current text: ${currentText.toString()}")
        Toast.makeText(view?.context,  text: "Translating ...", Toast.LENGTH_LONG).show()
        translator?.translate(currentText.toString())
            ?.addOnSuccessListener { it: String!
                println("success")
                println(it)
                _binding?.translationOutputText?.setText(it.toString())
            }
            ?.addOnFailureListener { it: Exception
                println("download fail")
                it.printStackTrace()
                Toast.makeText(context,  text: "Translation Failed..", Toast.LENGTH_LONG).show()
            }
    }
```

If glossary translation fails, translate with ML-KIT.

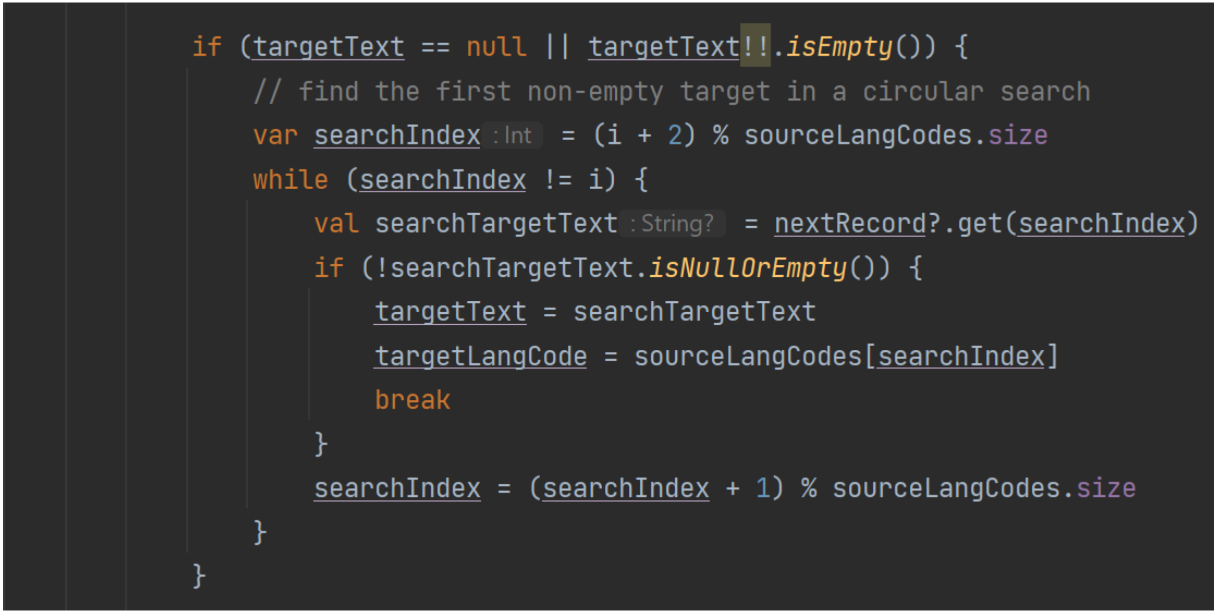


```
private fun makeGlossaryTranslation() : Boolean {

    // uses data structure

    /*val result = glossary.getTranslation(sourceLanguage!!,
      targetLanguage!!, currentText!!) ?: return false */

    // uses database
    val result : String  = databaseHelper.getTranslation(sourceLanguage!!,
        targetLanguage!!, currentText!!) ?: return false
    println("result: " + result)

    binding.translationOutputText.setText(result)
    return true
}
```

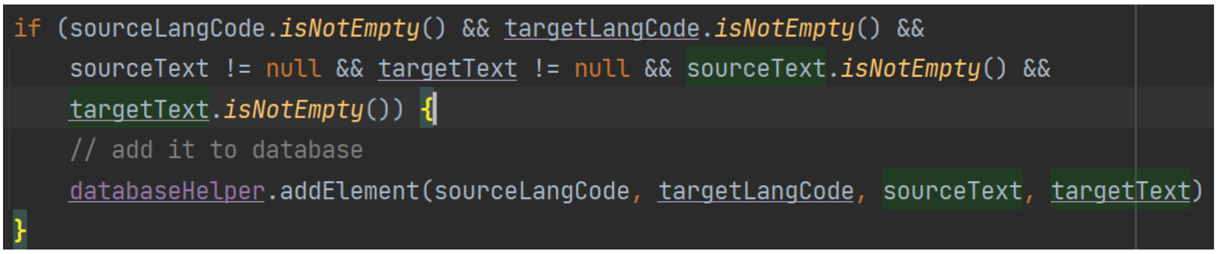
Translation using glossary.



```
private fun storeInDatabase() {
    val inputStream : InputStream?  = context?.resources?.openRawResource(R.raw.glossary)
    val reader = BufferedReader(InputStreamReader(inputStream))
    val csvReader : CSVReader!  = CSVReaderBuilder(reader).build()
    var nextRecord: Array<String>?
    if (csvReader.readNext().also { nextRecord = it } != null) {
        val sourceLangCodes : Array<String>  = nextRecord!!
        while (csvReader.readNext().also { nextRecord = it } != null) {
            for (i : Int  in sourceLangCodes.indices) {
                val sourceLangCode : String  = sourceLangCodes[i]
                val sourceText : String?  = nextRecord?.get(i)

                var targetLangCode : String  = sourceLangCodes[(i + 1) % sourceLangCodes.size]
                var targetText : String?  = nextRecord?.get((i + 1) % sourceLangCodes.size)
```

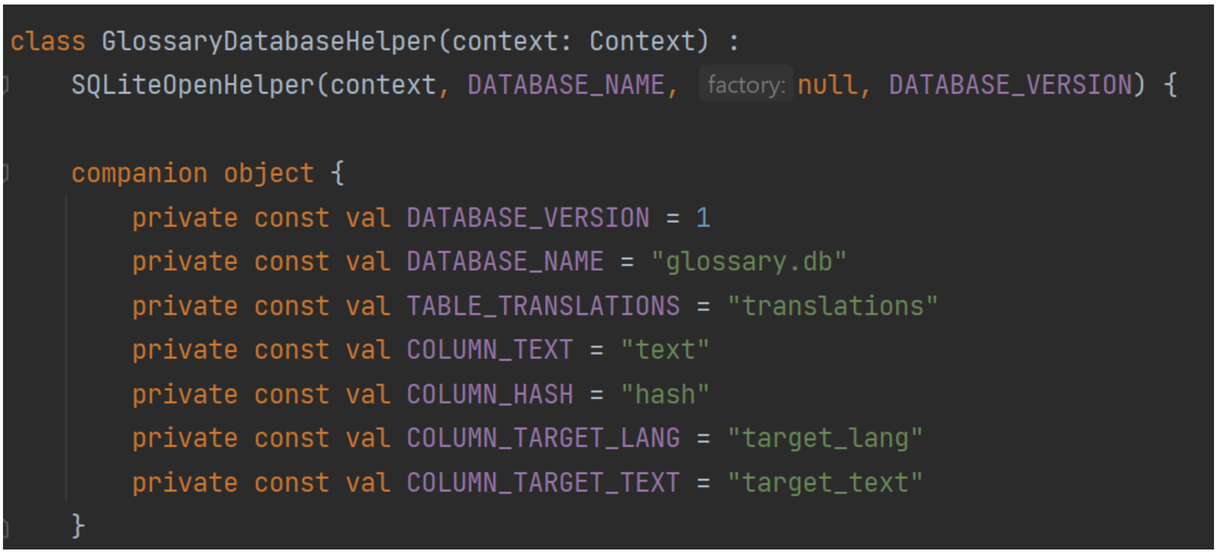


```
                if (targetText == null || targetText!!.isEmpty()) {
                    // find the first non-empty target in a circular search
                    var searchIndex : Int  = (i + 2) % sourceLangCodes.size
                    while (searchIndex != i) {
                        val searchTargetText : String?  = nextRecord?.get(searchIndex)
                        if (!searchTargetText.isNullOrEmpty()) {
                            targetText = searchTargetText
                            targetLangCode = sourceLangCodes[searchIndex]
                            break
                        }
                        searchIndex = (searchIndex + 1) % sourceLangCodes.size
                    }
                }
```



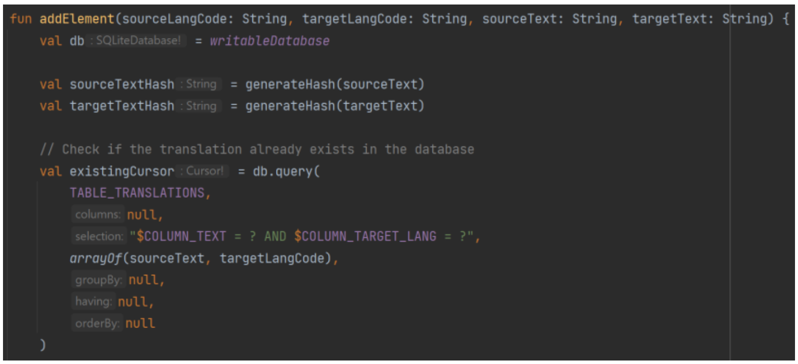
```
if (sourceLangCode.isNotEmpty() && targetLangCode.isNotEmpty() &&
    sourceText != null && targetText != null && sourceText.isNotEmpty() &&
    targetText.isNotEmpty()) {
    // add it to database
    databaseHelper.addElement(sourceLangCode, targetLangCode, sourceText, targetText)
}
```

Parsing algorithm, purpose of the algorithm is creating at least one relation with each language pair.

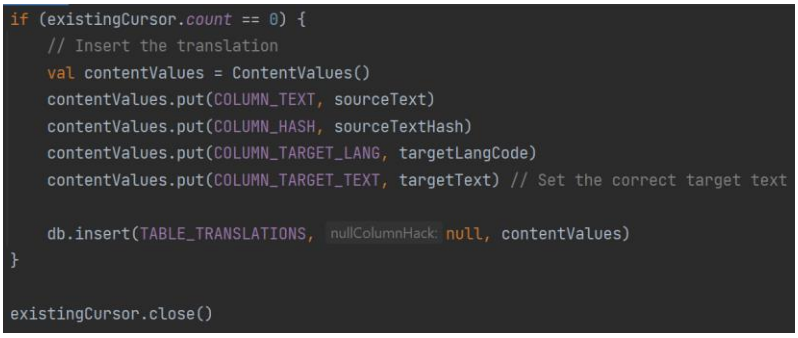


```
class GlossaryDatabaseHelper(context: Context) :
    SQLiteOpenHelper(context, DATABASE_NAME,  factory: null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_VERSION = 1
        private const val DATABASE_NAME = "glossary.db"
        private const val TABLE_TRANSLATIONS = "translations"
        private const val COLUMN_TEXT = "text"
        private const val COLUMN_HASH = "hash"
        private const val COLUMN_TARGET_LANG = "target_lang"
        private const val COLUMN_TARGET_TEXT = "target_text"
    }
```

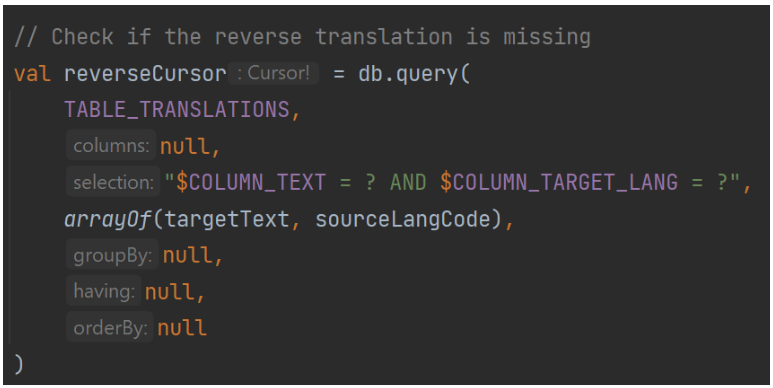
Database schema



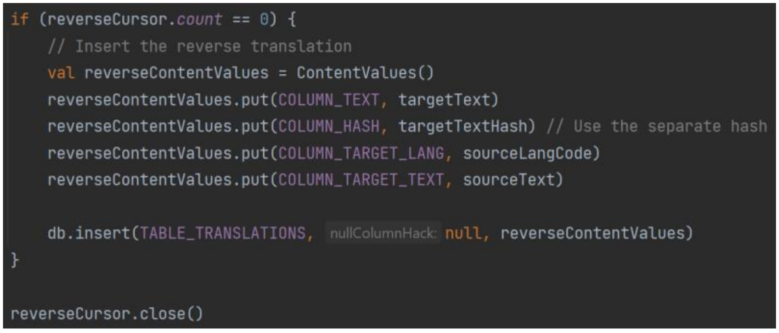
```
fun addElement(sourceLangCode: String, targetLangCode: String, sourceText: String, targetText: String) {
    val db : SQLiteDatabase!  = writableDatabase

    val sourceTextHash : String  = generateHash(sourceText)
    val targetTextHash : String  = generateHash(targetText)

    // Check if the translation already exists in the database
    val existingCursor : Cursor!  = db.query(
        TABLE_TRANSLATIONS,
         columns: null,
         selection: "$COLUMN_TEXT = ? AND $COLUMN_TARGET_LANG = ?",
        arrayOf(sourceText, targetLangCode),
         groupBy: null,
         having: null,
         orderBy: null
    )
```



```
if (existingCursor.count == 0) {
    // Insert the translation
    val contentValues = ContentValues()
    contentValues.put(COLUMN_TEXT, sourceText)
    contentValues.put(COLUMN_HASH, sourceTextHash)
    contentValues.put(COLUMN_TARGET_LANG, targetLangCode)
    contentValues.put(COLUMN_TARGET_TEXT, targetText) // Set the correct target text

    db.insert(TABLE_TRANSLATIONS,  nullColumnHack: null, contentValues)
}

existingCursor.close()
```

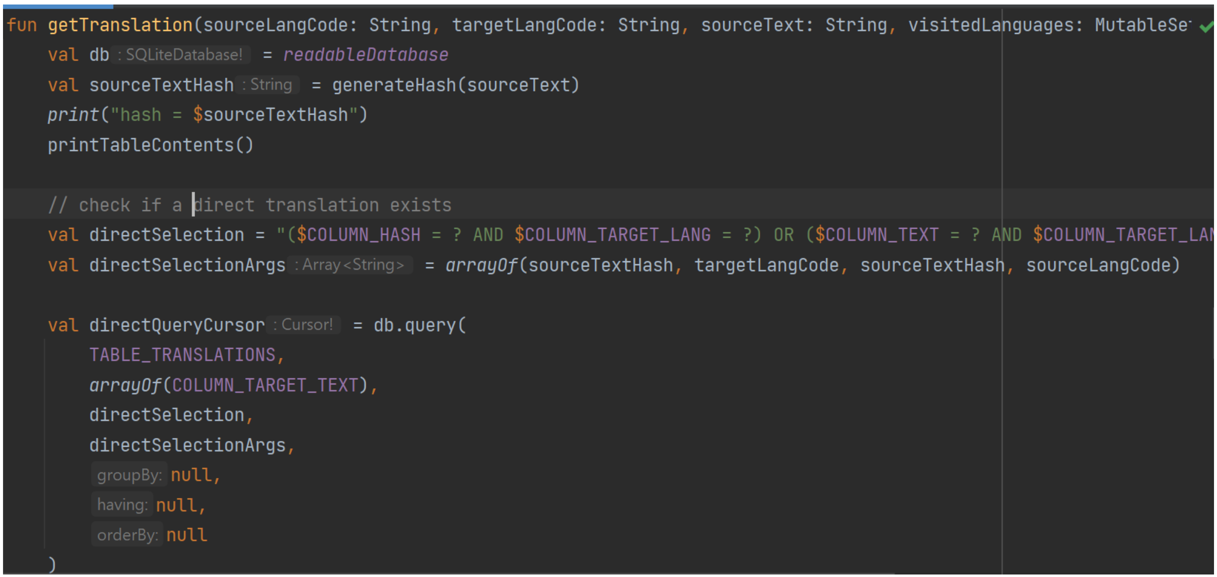


```
// Check if the reverse translation is missing
val reverseCursor : Cursor! = db.query(
    TABLE_TRANSLATIONS,
    columns: null,
    selection: "$COLUMN_TEXT = ? AND $COLUMN_TARGET_LANG = ?",
    arrayOf(targetText, sourceLangCode),
    groupBy: null,
    having: null,
    orderBy: null
)
```

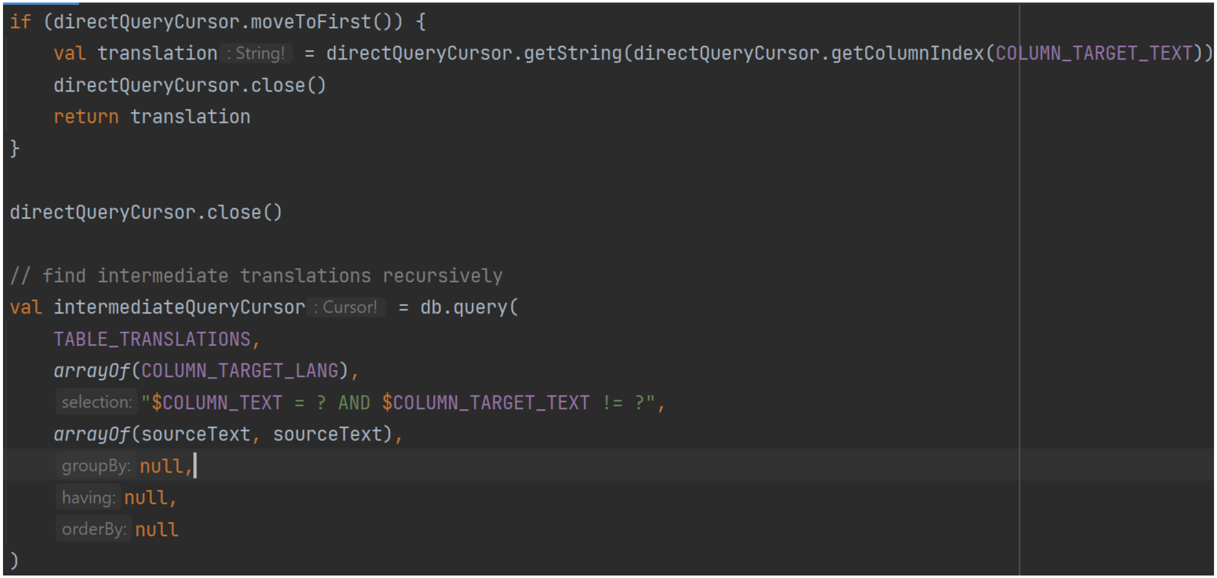


```
if (reverseCursor.count == 0) {
    // Insert the reverse translation
    val reverseContentValues = ContentValues()
    reverseContentValues.put(COLUMN_TEXT, targetText)
    reverseContentValues.put(COLUMN_HASH, targetTextHash) // Use the separate hash
    reverseContentValues.put(COLUMN_TARGET_LANG, sourceLangCode)
    reverseContentValues.put(COLUMN_TARGET_TEXT, sourceText)

    db.insert(TABLE_TRANSLATIONS, nullColumnHack: null, reverseContentValues)
}

reverseCursor.close()
```

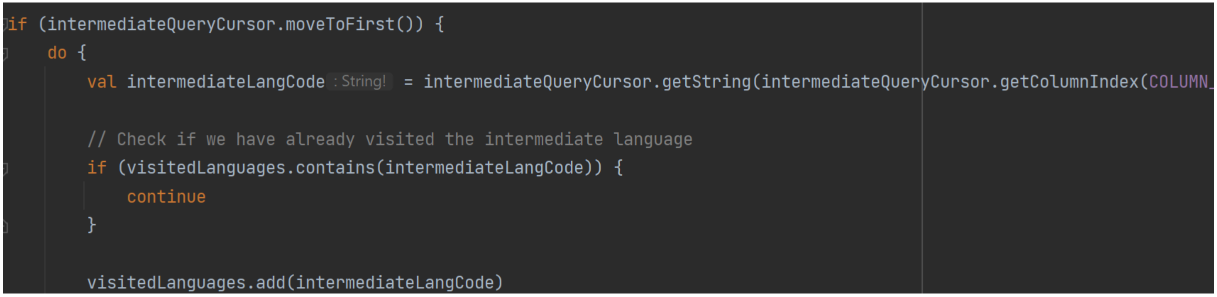
Insert the relation, then insert its reverse if it does not exist.



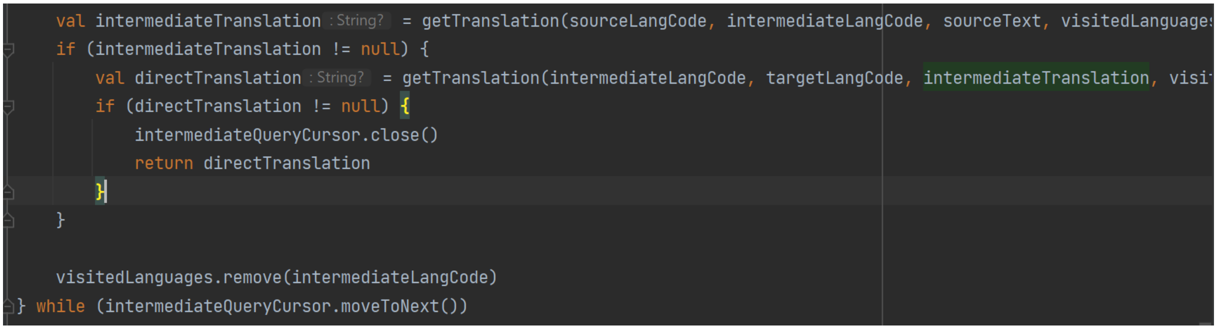
```
fun getTranslation(sourceLangCode: String, targetLangCode: String, sourceText: String, visitedLanguages: MutableSe
    val db : SQLiteDatabase! = readableDatabase
    val sourceTextHash : String = generateHash(sourceText)
    print("hash = $sourceTextHash")
    printTableContents()

    // check if a direct translation exists
    val directSelection = "($COLUMN_HASH = ? AND $COLUMN_TARGET_LANG = ?) OR ($COLUMN_TEXT = ? AND $COLUMN_TARGET_LA
    val directSelectionArgs : Array<String> = arrayOf(sourceTextHash, targetLangCode, sourceTextHash, sourceLangCode)

    val directQueryCursor : Cursor! = db.query(
        TABLE_TRANSLATIONS,
        arrayOf(COLUMN_TARGET_TEXT),
        directSelection,
        directSelectionArgs,
        groupBy: null,
        having: null,
        orderBy: null
    )
```



```
if (directQueryCursor.moveToFirst()) {
    val translation : String! = directQueryCursor.getString(directQueryCursor.getColumnIndex(COLUMN_TARGET_TEXT))
    directQueryCursor.close()
    return translation
}

directQueryCursor.close()

// find intermediate translations recursively
val intermediateQueryCursor : Cursor! = db.query(
    TABLE_TRANSLATIONS,
    arrayOf(COLUMN_TARGET_LANG),
    selection: "$COLUMN_TEXT = ? AND $COLUMN_TARGET_TEXT != ?",
    arrayOf(sourceText, sourceText),
    groupBy: null,
    having: null,
    orderBy: null
)
```



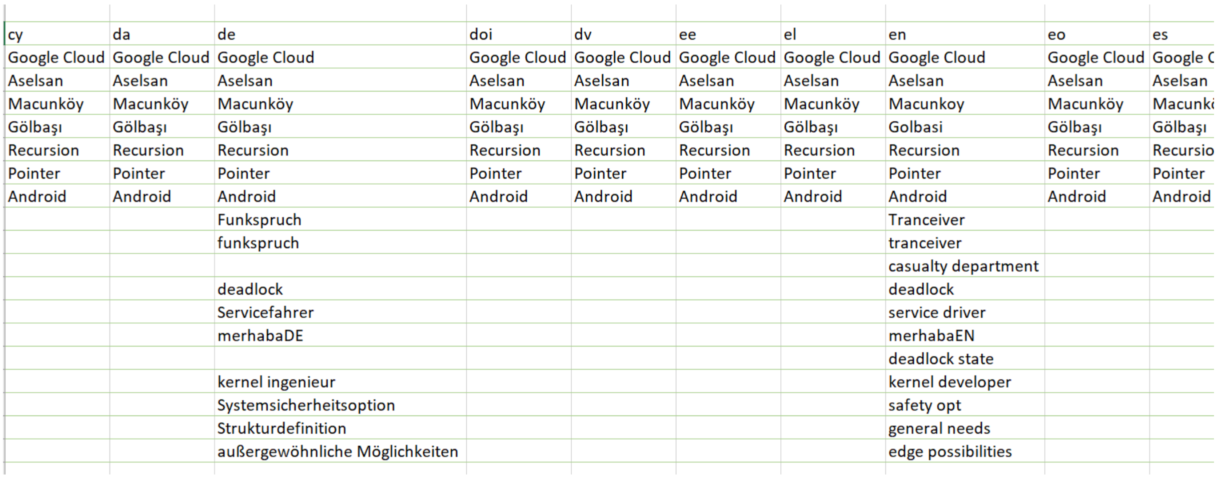
```
if (intermediateQueryCursor.moveToFirst()) {
    do {
        val intermediateLangCode : String! = intermediateQueryCursor.getString(intermediateQueryCursor.getColumnIndex(COLUMN_

        // Check if we have already visited the intermediate language
        if (visitedLanguages.contains(intermediateLangCode)) {
            continue
        }

        visitedLanguages.add(intermediateLangCode)
```



```
    val intermediateTranslation : String? = getTranslation(sourceLangCode, intermediateLangCode, sourceText, visitedLanguage
    if (intermediateTranslation != null) {
        val directTranslation : String? = getTranslation(intermediateLangCode, targetLangCode, intermediateTranslation, visi
        if (directTranslation != null) {
            intermediateQueryCursor.close()
            return directTranslation
        }
    }

    visitedLanguages.remove(intermediateLangCode)
} while (intermediateQueryCursor.moveToNext())
```

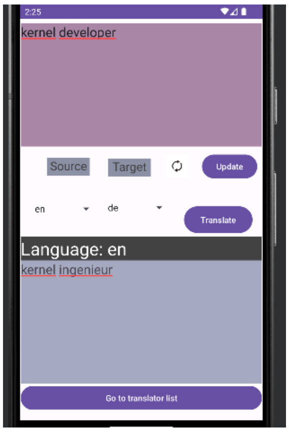
Try to find direct relation. If not found, then traverse that row using intermediate translations. In order to not visit same translation again put them in on a set.

Usage



cy	da	de	doi	dv	ee	el	en	eo	es
Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google C
Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan
Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunkoy	Macunköy	Macunk
Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Golbasi	Gölbaşı	Gölbaşı
Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursio
Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer
Android	Android	Android	Android	Android	Android	Android	Android	Android	Android
		Funkspruch					Tranceiver		
		funkspruch					tranceiver		
							casualty department		
		deadlock					deadlock		
		Servicefahrer					service driver		
		merhabaDE					merhabaEN		
							deadlock state		
		kernel ingenieur					kernel developer		
		Systemsicherheitsoption					safety opt		
		Strukturdefinition					general needs		
		außergewöhnliche Möglichkeiten					edge possibilities		

Some part of the current glossary

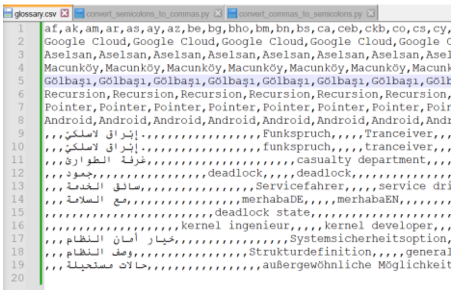


kernel developer

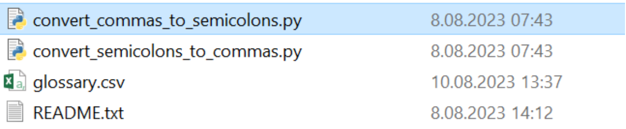
Language: en

kernel ingenieur

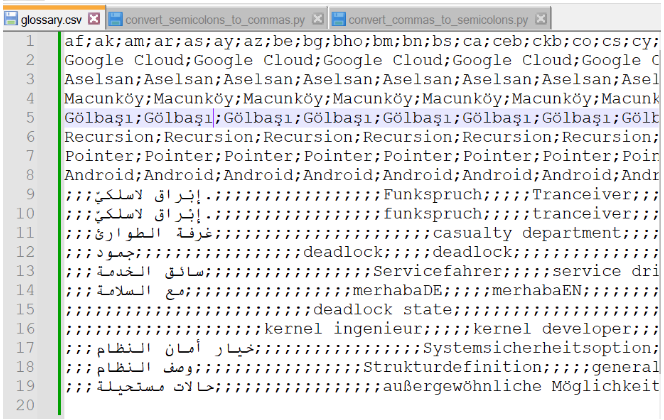
Translates correctly


glossary.csv
convert_semicolons_to_commas.py
convert_commas_to_semicolons.py

```
af,ak,am,ar,as,ay,az,be,bg,bho,bm,bn,bs,ca,ceb,ckb,co,cs,cy,
Google Cloud,Google Cloud,Google Cloud,Google Cloud,Google C
Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Aselsan,Asel
Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunk
Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölb
Recursion,Recursion,Recursion,Recursion,Recursion,Recursion,
Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Poir
Android,Android,Android,Android,Android,Android,Android,Andr
,,,إبْراق لاسلكيّ.,,,,,,,,,,,,,,,,,Funkspruch,,,,,Tranceiver,,,
,,,إبْراق لاسلكيّ.,,,,,,,,,,,,,,,,,funkspruch,,,,,tranceiver,,,
,,,غرفة الطوارئ,,,,,,,,,,,,,,,,,,,,,,casualty department,,,,
,,,جمود,,,,,,,,,,,,,,,,,deadlock,,,,,deadlock,,,,,,,,,,,,,,
,,,سائق الخدمة,,,,,,,,,,,,,,,,,Servicefahrer,,,,,service dri
,,,مع السلامة,,,,,,,,,,,,,,,,,merhabaDE,,,,,merhabaEN,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,deadlock state,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,kernel ingenieur,,,,,kernel developer,,,
,,,خيار أمان النظام,,,,,,,,,,,,,,,,,Systemsicherheitsoption,
,,,وصف النظام,,,,,,,,,,,,,,,,,Strukturdefinition,,,,,general
,,,حالات مستحيلة,,,,,,,,,,,,,,,,,außergewöhnliche Möglichkeit
```

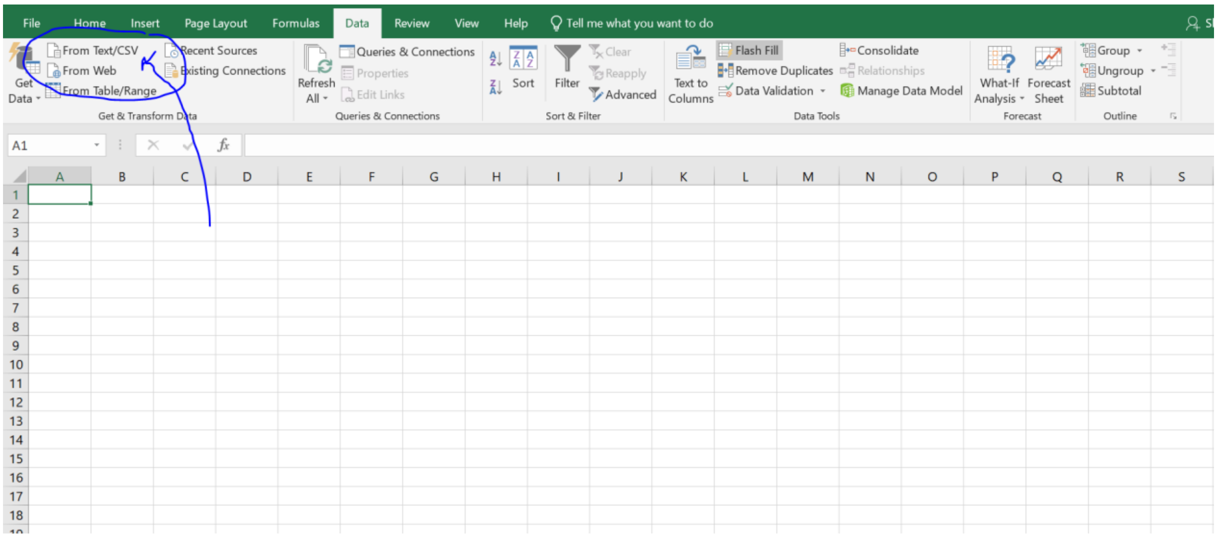


convert_commas_to_semicolons.py	8.08.2023 07:43
convert_semicolons_to_commas.py	8.08.2023 07:43
glossary.csv	10.08.2023 13:37
README.txt	8.08.2023 14:12

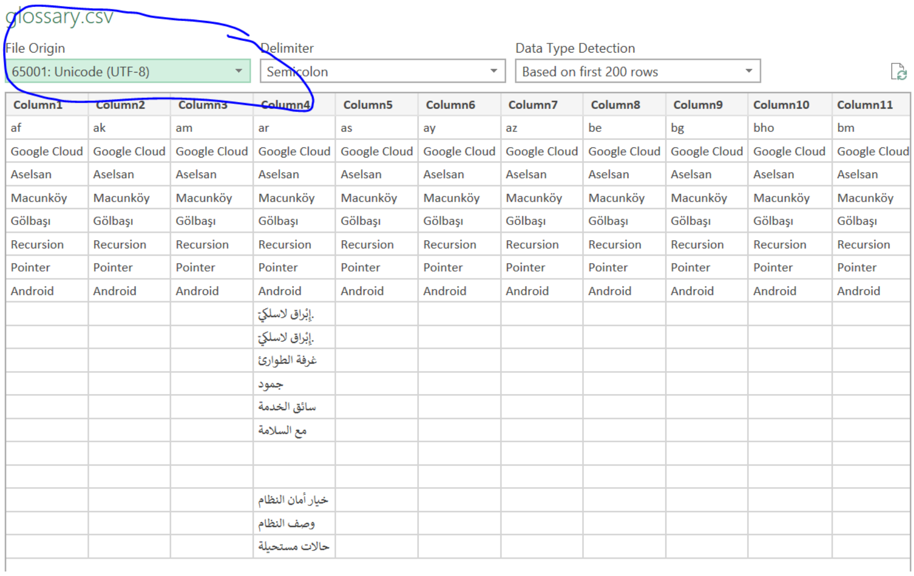

glossary.csv
convert_semicolons_to_commas.py
convert_commas_to_semicolons.py

```
af;ak;am;ar;as;ay;az;be;bg;bho;bm;bn;bs;ca;ceb;ckb;co;cs;cy;
Google Cloud;Google Cloud;Google Cloud;Google Cloud;Google C
Aselsan;Aselsan;Aselsan;Aselsan;Aselsan;Aselsan;Aselsan;Asel
Macunköy;Macunköy;Macunköy;Macunköy;Macunköy;Macunköy;Macunk
Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölbaşı;Gölb
Recursion;Recursion;Recursion;Recursion;Recursion;Recursion;
Pointer;Pointer;Pointer;Pointer;Pointer;Pointer;Pointer;Poin
Android;Android;Android;Android;Android;Android;Android;Andr
;;;إبْراق لاسلكيّ.;;;;;;;;;;;;;;;;;Funkspruch;;;;;Tranceiver;;;
;;;إبْراق لاسلكيّ.;;;;;;;;;;;;;;;;;funkspruch;;;;;tranceiver;;;
;;;غرفة الطوارئ;;;;;;;;;;;;;;;;;;;;;;casualty department;;;;
;;;جمود;;;;;;;;;;;;;;;;;deadlock;;;;;deadlock;;;;;;;;;;;;;;
;;;سائق الخدمة;;;;;;;;;;;;;;;;;Servicefahrer;;;;;service dri
;;;مع السلامة;;;;;;;;;;;;;;;;;merhabaDE;;;;;merhabaEN;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;deadlock state;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;kernel ingenieur;;;;;kernel developer;;;
;;;خيار أمان النظام;;;;;;;;;;;;;;;;;Systemsicherheitsoption;
;;;وصف النظام;;;;;;;;;;;;;;;;;Strukturdefinition;;;;;general
;;;حالات مستحيلة;;;;;;;;;;;;;;;;;außergewöhnliche Möglichkeit
```

To open it on excel commas must be converted to semicolon.



Open blank excel, and click this button.



glossary.csv

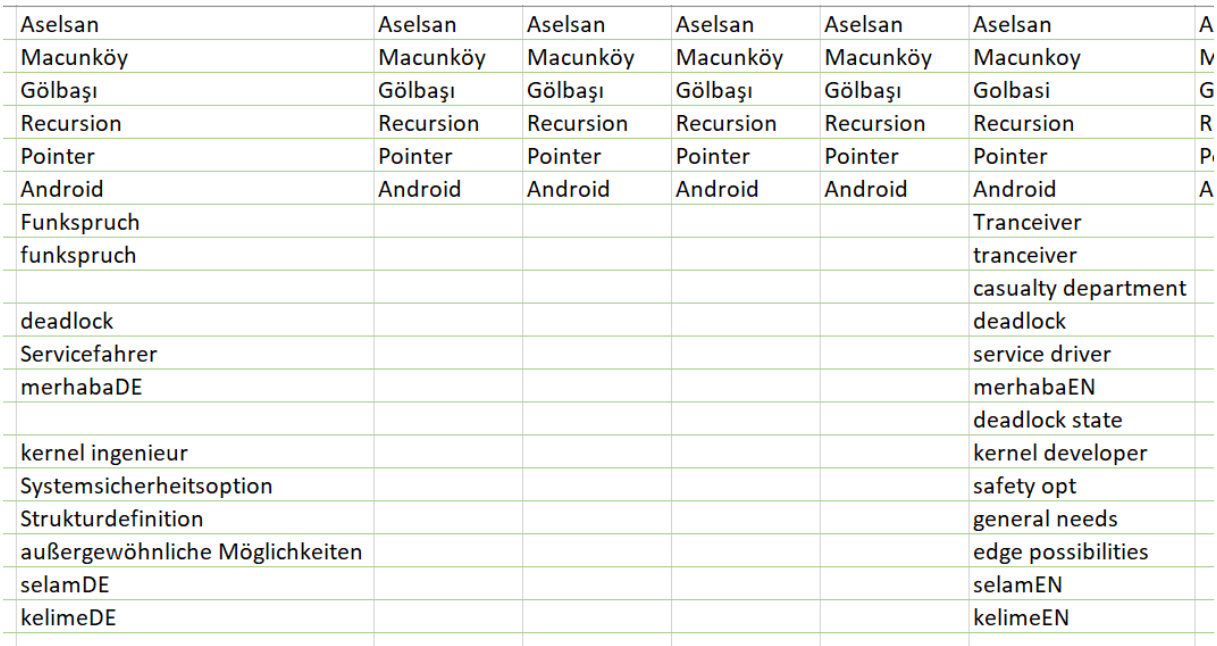
File Origin: 65001: Unicode (UTF-8)

Delimiter: Semicolon

Data Type Detection: Based on first 200 rows

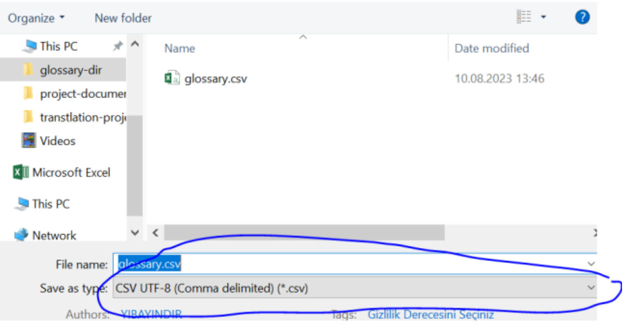
Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11
af	ak	am	ar	as	ay	az	be	bg	bho	bm
Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud	Google Cloud
Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan
Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunköy
Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı
Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	Recursion
Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	Pointer
Android	Android	Android	Android	Android	Android	Android	Android	Android	Android	Android
			إبْراق لاسلكيّ.							
			إبْراق لاسلكيّ.							
			غرفة الطوارئ							
			جمود							
			سائق الخدمة							
			مع السلامة							
			خيار أمان النظام							
			وصف النظام							
			حالات مستحيلة							

Choose UTF-8 encoding.



Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	Aselsan	A
Macunköy	Macunköy	Macunköy	Macunköy	Macunköy	Macunkoy	M
Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Gölbaşı	Golbasi	G
Recursion	Recursion	Recursion	Recursion	Recursion	Recursion	R
Pointer	Pointer	Pointer	Pointer	Pointer	Pointer	P
Android	Android	Android	Android	Android	Android	A
Funkspruch					Tranceiver	
funkspruch					tranceiver	
					casualty department	
deadlock					deadlock	
Servicefahrer					service driver	
merhabaDE					merhabaEN	
					deadlock state	
kernel ingenieur					kernel developer	
Systemsicherheitsoption					safety opt	
Strukturdefinition					general needs	
außergewöhnliche Möglichkeiten					edge possibilities	
selamDE					selamEN	
kelimeDE					kelimeEN	

kelimeDE, kelimeEN, kelimeTR, kelimeAR, are added.



glossary.csv 10.08.2023 13:46

File name: glossary.csv

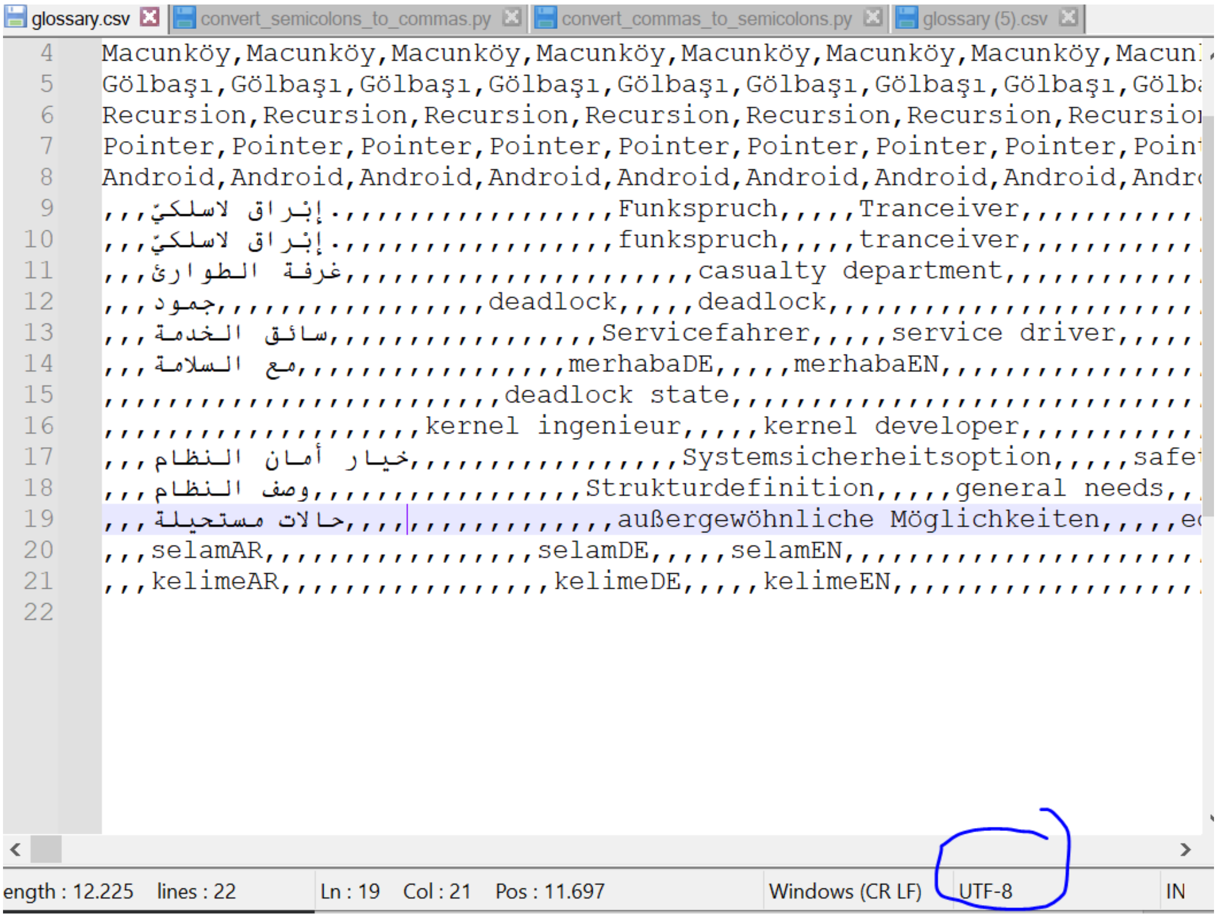
Save as type: CSV UTF-8 (Comma delimited) (*.csv)

Authors: YBAYINDIR Tags: Gizlilik Derecesini Seçiniz

Save the file in this format.

Close the excel.

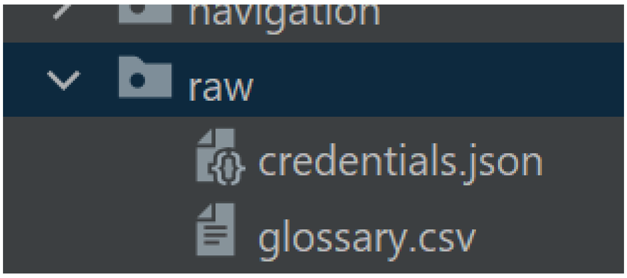
Run semicolon to comma script.


glossary.csv
convert_semicolons_to_commas.py
convert_commas_to_semicolons.py
glossary (5).csv

```
Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macunköy,Macun
Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölbaşı,Gölba
Recursion,Recursion,Recursion,Recursion,Recursion,Recursion,Recursio
Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Pointer,Poin
Android,Android,Android,Android,Android,Android,Android,Android,Andr
,,,إبراق لاسلكيّ.,,,,,,,,,,,,,,,,,Funkspruch,,,,Tranceiver,,,,,,,,,,,
,,,إبراق لاسلكيّ.,,,,,,,,,,,,,,,,,funkspruch,,,,tranceiver,,,,,,,,,,,
,,,غرفة الطوارئ,,,,,,,,,,,,,,,,,,,,,,casualty department,,,,,,,,,,,,
,,,جمود,,,,,,,,,,,,,,,,,deadlock,,,,deadlock,,,,,,,,,,,,,,,,,,,,,,,
,,,سائق الخدمة,,,,,,,,,,,,,,,,,Servicefahrer,,,,service driver,,,,,
,,,مع السلامة,,,,,,,,,,,,,,,,,merhabaDE,,,,merhabaEN,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,deadlock state,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,kernel ingenieur,,,,kernel developer,,,,,,,,,,,
,,,خيار أمان النظام,,,,,,,,,,,,,,,,,Systemsicherheitsoption,,,,safe
,,,وصف النظام,,,,,,,,,,,,,,,,,Strukturdefinition,,,,general needs,,
,,,حالات مستحيلة,,,,,,,,,,,,,,,,,außergewöhnliche Möglichkeiten,,,,e
,,,selamAR,,,,,,,,,,,,,,,,,selamDE,,,,selamEN,,,,,,,,,,,,,,,,,,,,,,
,,,kelimeAR,,,,,,,,,,,,,,,,,kelimeDE,,,,kelimeEN,,,,,,,,,,,,,,,,,,,
```

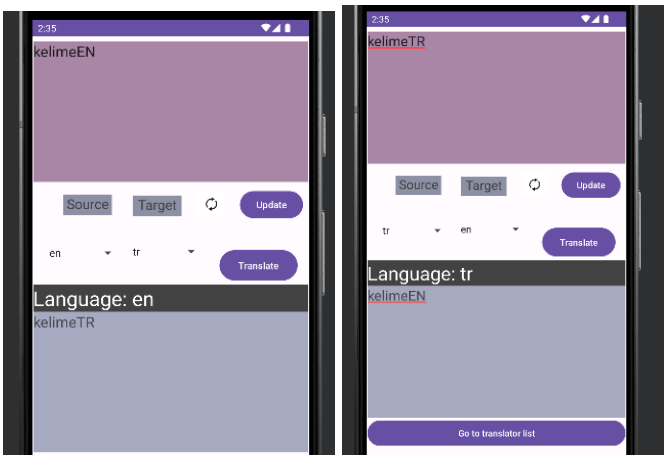
ength : 12.225   lines : 22   Ln : 19   Col : 21   Pos : 11.697   Windows (CR LF)   UTF-8   IN

After running, it must be UTF-8. Without BOM.



Add new file to that location.

After adding it to the end, program must be restarted.



kelimeEN

en tr

Language: en

kelimeTR

kelimeTR

tr en

Language: tr

kelimeEN

Custom glossary works.

TODO, and Unchecked Parts

- TODO: Add loading screen while database is updating.
- TODO: Implement more optimized database schema, and lookup mechanism.
- TODO: Add user messages when download fail or success.
- TODO: Prevent program from crashing if csv format is wrong.
- TODO: Inform user if update fails.

Requirements Result

Requirement Type	Requirement	MET	AVG MET	NOT MET
Language Translation with Glossary	Translation must be done according to the given glossary, but it is not required to detect glossary words inside input text like Google Glossary. Direct translation is enough.	X		
	If source text does not match with glossary, ML-KIT model must be used.	X		
	Translated result text must be semantically accurate.		X	
	Target text must consist of alphabets of the target language.	X		
Language Identification	Entered text must be identified.		X	
	Generally, those texts will be few words sentence or just word. They must be identified accurately.		X	
User Interface	It must be responsive, understandable.	X		

	Source, and target languages can be decided by user, or source language might be auto-identified. Decision must be decidable by user.	X		
	Identification must be done after each key stroke.	X		
	Translation must only be done after button click.	X		
	Reverse translation must be done with reverse button.	X		
	Result and source texts must be readable.	X		
	There must be update button to load glossary to database.	X		
	User must see loading screen while database is updating.			X
Functional	Application must work without internet connection except downloading the models.	X		
	User must be informed if a download is started, and ends (fail or success).			X
	User must be informed if update fails.			X
	Program must not crash if csv file is in wrong format, and user must be informed.			X
	Program must be able to parse comma separated, utf-8 format csv files.	X		
	Database update, and queries must not take too much time.		X	
	Glossaries must not be held in process memory, if data structure implementation won't be used.	X		

Future Suggestions

- Cache mechanism can be added for faster search.
- With using pre-glossaries, may be keywords inside texts can be translated, too.

2. Update Logs

09/08/2023 – Burak Kocausta – Documentation is created.
09/08/2023 – Burak Kocausta – Requirement, and requirement results are completed.
10/08/2023 – Burak Kocausta – Usages, TODO parts are completed.
10/08/2023 – Burak Kocausta – 1.0 version is completed.