

תרגול מס' 5: המתרגם

שימוש במחלקות קיימות

- תכנית המתרגמת קטעי טקסט לשפה אחרת
- הקלט: קובץ המכיל את קטעי הטקסט וכן את השפה אליה רוצים לתרגם



- האם כבר יש שירות תרגום שאנחנו יכולים להשתמש בו?
- אולי נוכל לעשות שימוש ב Microsoft Translate?
- כיצד קוראים מקבצים?
- מה הפורמט של הקלט?
- נצטרך להחליט

■ כצעד ראשון נפתור בעיה הרבה יותר פשוטה

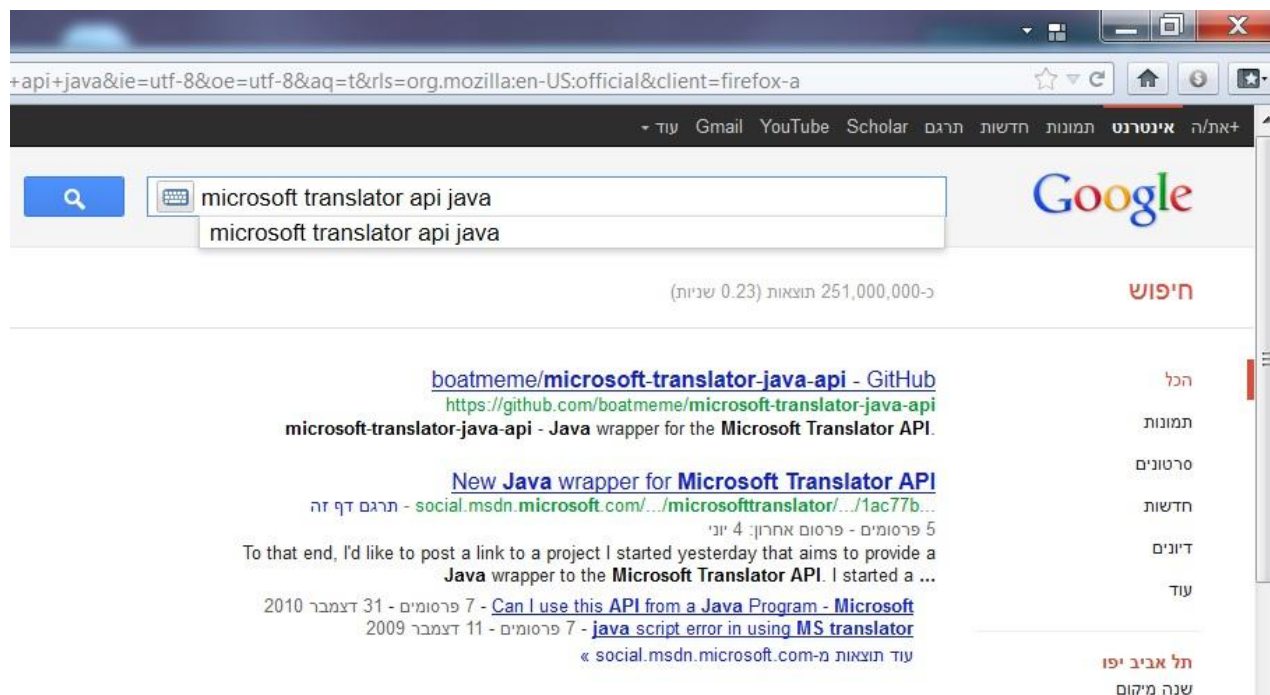
■ תכנית שמתרגמת את המילה "Hello" מאנגלית לצרפתית

■ יש: שימוש בשירות תרגום

■ אין: קלט, טקסט, עבודה עם קבצים, פורמט

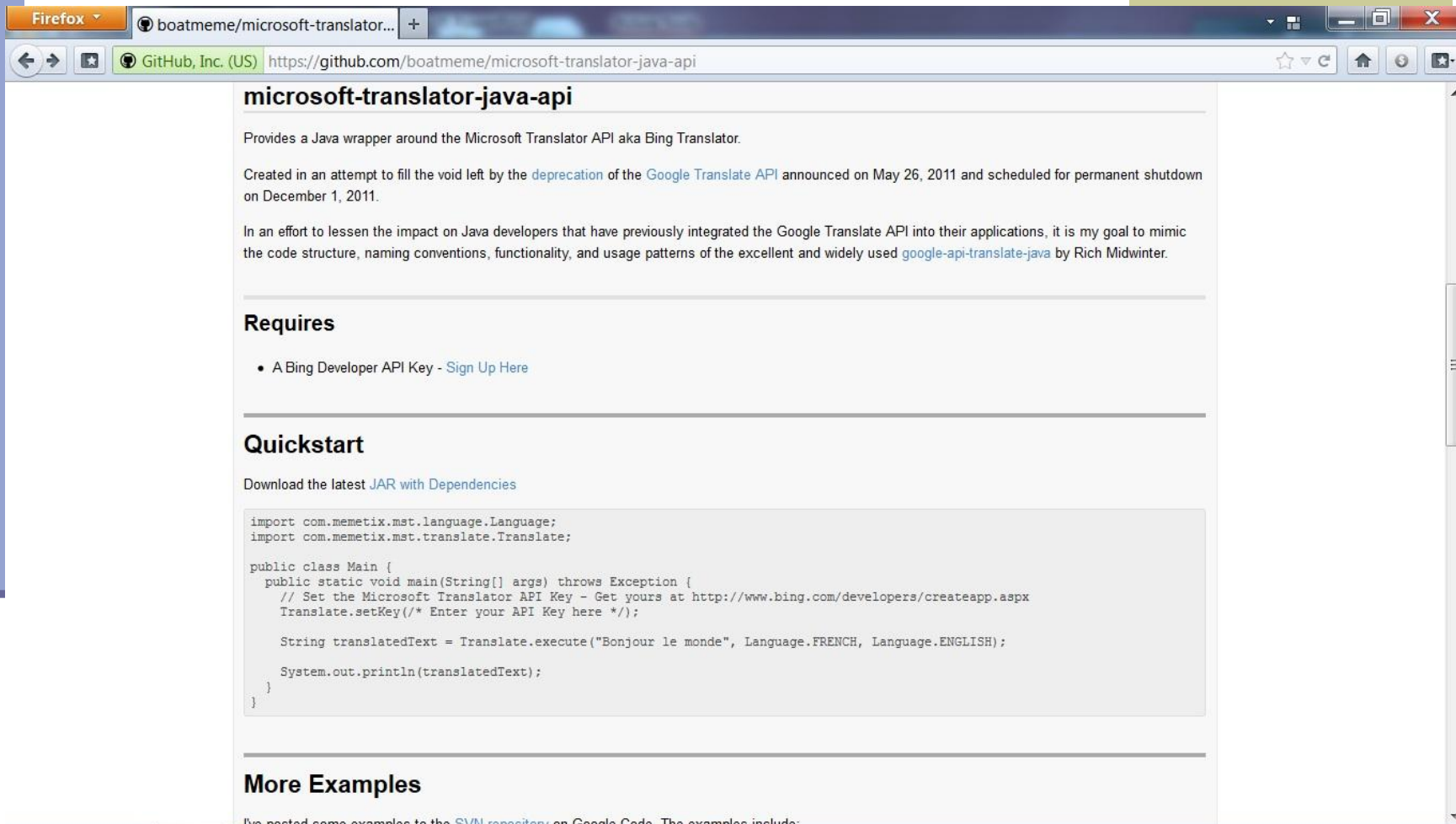
API – Application Programming Interface

ממשק המאפשר לאפליקציה לתקשר עם תוכנה אחרת



תכנות מתקדם בשפת Java
אוניברסיטת תל אביב

Eureka!



The screenshot shows a Firefox browser window with the address bar displaying 'https://github.com/boatmeme/microsoft-translator-java-api'. The page content includes:

microsoft-translator-java-api

Provides a Java wrapper around the Microsoft Translator API aka Bing Translator.

Created in an attempt to fill the void left by the [deprecation](#) of the [Google Translate API](#) announced on May 26, 2011 and scheduled for permanent shutdown on December 1, 2011.

In an effort to lessen the impact on Java developers that have previously integrated the Google Translate API into their applications, it is my goal to mimic the code structure, naming conventions, functionality, and usage patterns of the excellent and widely used [google-api-translate-java](#) by Rich Midwinter.

Requires

- A Bing Developer API Key - [Sign Up Here](#)

Quickstart

Download the latest [JAR with Dependencies](#)

```
import com.memetix.mst.language.Language;
import com.memetix.mst.translate.Translate;

public class Main {
    public static void main(String[] args) throws Exception {
        // Set the Microsoft Translator API Key - Get yours at http://www.bing.com/developers/createapp.aspx
        Translate.setKey(/* Enter your API Key here */);

        String translatedText = Translate.execute("Bonjour le monde", Language.FRENCH, Language.ENGLISH);

        System.out.println(translatedText);
    }
}
```

More Examples

I've posted some examples to the [SVN repository](#) on Google Code. The examples include:

שלב א'

```
public class TranslatorEngine1 {
    private static final String APIkey = "Your Bing Developer API key
    goes here";

    public static void main(String[] args) throws Exception {
        Translate.setKey(APIkey);

        String TranslatedText = Translate.execute("Hello",
        Language.ENGLISH, Language.FRENCH);
        System.out.println(TranslatedText);
    }
}
```

אינטראקציה עם המשתמש

- נתחיל להתקדם עקב בצד אגודל אל היעד שלנו
- קלט מהמשתמש יינתן בשורת הפקודה
 - פרמטר ראשון: המילה לתרגום
 - פרמטר שני: שפת המקור
 - פרמטר שלישי: שפת היעד

שלב ב'

```
public class TranslatorEngine2 {
    private static final String APIkey = "Your Bing Developer API key
    goes here";

    public static void main(String[] args) throws Exception {
        Translate.setKey(APIkey);

        String TranslatedText = Translate.execute(args[0],
            Language.fromString(args[1]),
            Language.fromString(args[2]));
        System.out.println(TranslatedText);
    }
}
```

קריאת קלט

■ נקרא קלט מהמשתמש (console)

■ עדיין מילה אחת

■ אין שימוש בקבצים

■ נשתמש במחלקה Scanner

■ מה הפורמט של הקלט?

המחלקה Scanner

- סורק טקסט פשוט
- "שובר" את הקלט לרכיביו השונים (מילה, מספר וכדומה)
- בעת יצירה מקבל כפרמטר מהיכן לקרוא את הקלט

```
Scanner s = new Scanner(System.in);  
int anInt = s.nextInt();  
float aFloat = s.nextFloat();  
String aString = s.next();  
String aLine = s.nextLine();
```

פורמט הקלט

מהו הפרוטוקול המשותף שחולקים האפליקציה והמשתמש לצורך התקשורת ביניהם

■ איזה מידע דרוש

■ כיצד הוא מקודד (מספר, מחרוזת, ...)

■ מה סדר הפרמטרים

■ נבחר: `<target-lang>` `<source-lang>` `<word>`

דוגמא

קרא מ - standard input

```
Scanner s = new Scanner(System.in);  
System.out.println("enter line:");  
while (s.hasNext())  
    System.out.println(s.next());
```

קרא את ה - Token הבא

שלב ג'

```
public class TranslatorEngine3 {
    private static final String APIkey = "Your Bing Developer API key
    goes here";

    public static void main(String[] args) throws Exception {
        Translate.setKey(APIkey);

        Scanner s = new Scanner(System.in);
        String[] fragments = s.nextLine().split(" ");
        String TranslatedText = Translate.execute(fragments[0],
            Language.fromString(fragments[1]),
            Language.fromString(fragments[2]));
        System.out.println(TranslatedText);
    }
}
```

קבצים

- במקום לקרוא את שורת הקלט מהמשתמש
נקרא אותה מקובץ
- קובץ מיצג ע"י המחלקה File
- נאתחל את האובייקט עם המסלול (path) לקובץ

```
File f =
```

```
new File("C:\\Software1\\example.txt");
```

מסלול (Path) לקובץ

מסלול יחסי – Relative path ■

`new File("example.txt")` ■

ב- eclipse המיקום הנוכחי הוא Project root ■

מסלול מלא – Absolute path ■

`new File("C:\\Software1\\example.txt")` ■


```
public class TranslatorEngine4 {
    private static final String APIkey = "API key";
    private static final String FILE_NAME = "example.txt";

    public static void main(String[] args) throws Exception {
        Translate.setKey(APIkey);

        Scanner s = new Scanner(new File(FILE_NAME));
        String[] fragments = s.nextLine().split(" ");
        String TranslatedText = Translate.execute(fragments[0],
            Language.fromString(fragments[1]),
            Language.fromString(fragments[2]));
        System.out.println(TranslatedText);
    }
}
```

קלטים מרובים

■ מספר שורות קלט מקובץ

■ נקרא מספר קלטים עד לסוף הקובץ

■ שימוש ב**nextLine** ו**hasNextLine**

שלב ה'

```
public class TranslatorEngine5 {
    private static final String APIkey = "API key";
    private static final String FILE_NAME = "...";

    public static void main(String[] args) throws Exception {
        Translate.setKey(APIkey);

        Scanner s = new Scanner(new File(FILE_NAME));
        while (s.hasNextLine()) {
            String[] fragments = s.nextLine().split(" ");
            System.out.println(Translate.execute(fragments[0],
                Language.fromString(fragments[1]),
                Language.fromString(fragments[2])));
        }
    }
}
```

פיסקה

■ פיסקה ולא רק מילה אחת

■ מה יהיה הפורמט החדש?

■ נבחר:

`<source-lang>#<target-lang>#<paragraph>`

Example - Scanner

Set delimiters

```
String input = "1 fish 2 fish red fish blue fish";
Scanner s =
    new Scanner(input).useDelimiter(" *fish *");
while (s.hasNext())
    System.out.println(s.next());
s.close();
```

שלב ו'

```
public class TranslatorEngine6 {  
    private static final String APIkey = "API key";  
    private static final String FILE_NAME = "...";  
    public static void main(String[] args) throws Exception {  
        Translate.setKey(APIkey);  
  
        Scanner s = new Scanner(new File(FILE_NAME));  
        s.useDelimiter("#");  
        Language from = Language.fromString(s.next());  
        Language to = Language.fromString(s.next());  
        s.skip("#");  
        StringBuffer text = new StringBuffer();  
        while (s.hasNextLine()) {  
            text.append(s.nextLine());  
            text.append(' ');  
        }  
        System.out.println(Translate.execute(text.toString(), from, to));  
    }  
}
```

לאן עכשיו?

- תרגום מספר קבצים
- מספר פסקאות בקובץ יחיד
- לוותר על חלק מהפרמטרים
 - זיהוי אוטומטי של שפת הקלט
- שימוש בשמות משמעותיים לשפות
- GUI
- ...

מה חסר?

■ טיפול בשגיאות

- חסרים פרמטרים
- שפות לא מוגדרות
- מנגנון התרגום נכשל
- קובץ לא קיים
- ...