

6.1

SERENITY BDD + SCREENPLAY con CUCUMBER

¡Bienvenidos a nuestra guía 6.1 del patrón Screenplay!

En esta guía, aprenderemos como utilizar datadriven en Cucumber.
Vamos a definir nuestra historia de usuario y ver como podemos pasar nuestros datos.

```
*traductor_google_datadriven.feature
1 #Author: your.email@your.domain.com
2 Feature:
3     Como un usuario
4     Quiero ingresar a google Translator
5     A traducir palabras entre diferentes lenguajes
6
7     @traducir
8     Scenario: Traducir de Ingles a Español
9         Given que Yeison quiere usar el traductor de google
10        When el traduce una palabra de inglés a español
11            |origen|destino|palabrai|
12            |inglés|español|table|
13        Then el debería ver la palabra traducida del idioma origen a idioma destino
14            |palabrae|
15            |mesa|
```

Se puede observar que en los pasos “WHEN” y THEN” se colocaron unas tablas de datos, cada una de estas tiene una línea de encabezados y una línea de datos a usar.

Ahora vamos a crear una clase en nuestro paquete “model” llamada “GoogleTraslateData” y definimos sus atributos con el mismo nombre de los títulos de los encabezados que le asignamos en la tabla de nuestro feature:

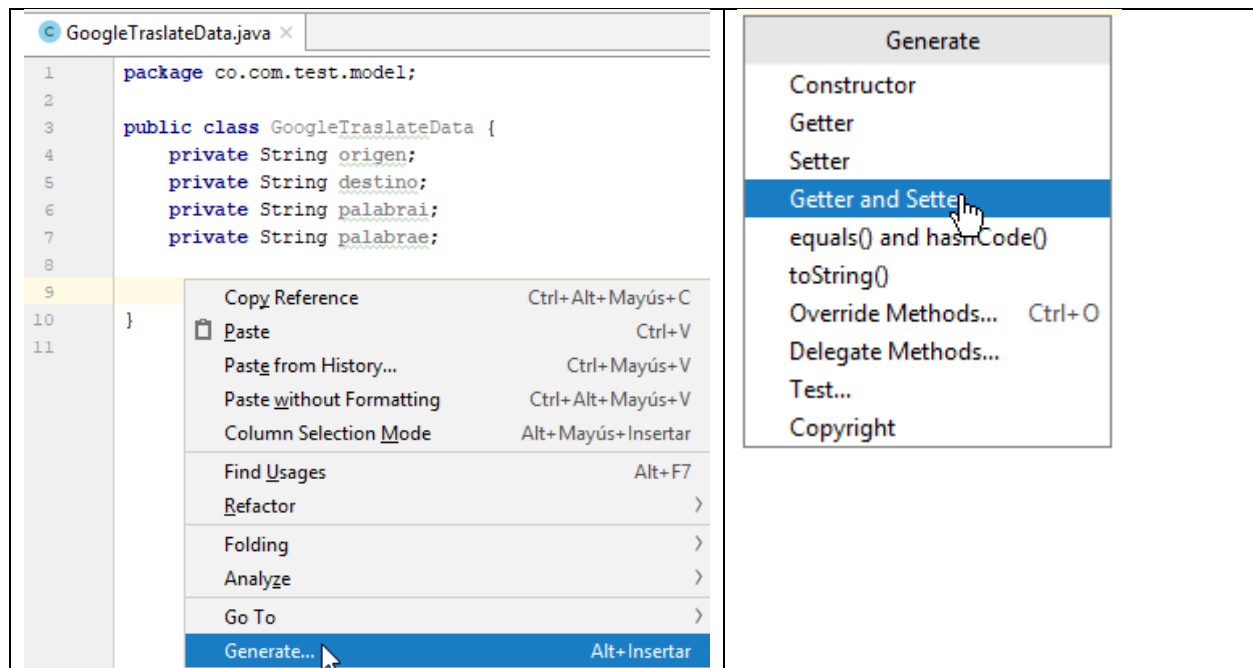
```
GoogleTraslateData.java
1 package co.com.test.model;
2
3 public class GoogleTraslateData {
4     private String origen;
5     private String destino;
6     private String palabrai;
7     private String palabrae;
8 }
9
```

```
When el traduce una palabra de inglés a español
|origen|destino|palabrai|
|inglés|español|table|
```



Una vez definidos los atributos, procedemos a generar los getter y los setter para dichos campos de esta nueva clase de la siguiente manera:

Damos click derecho en la clase y nos desplazamos hasta donde dice “**Generate**” y “**Getters and Setters**”.

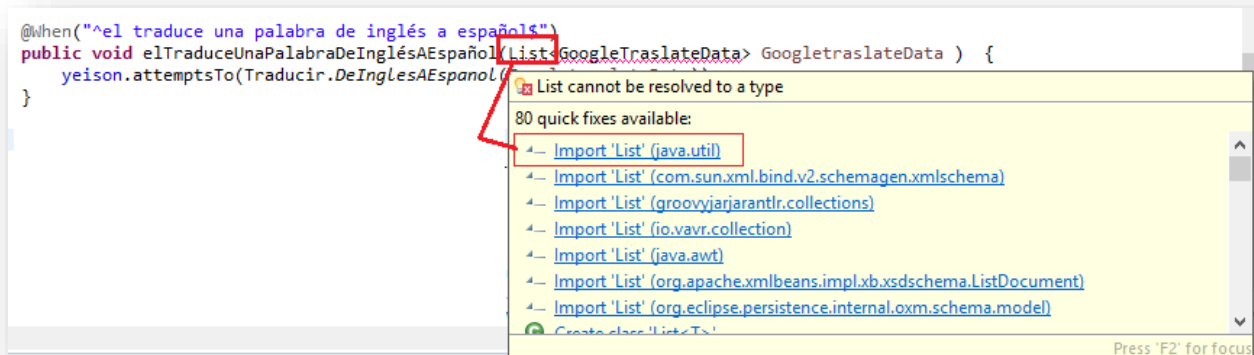


Por último, en nuestra clase StepDefinition agregamos como parámetro una lista que contenga elementos del tipo “**GoogleTraslateData**”.

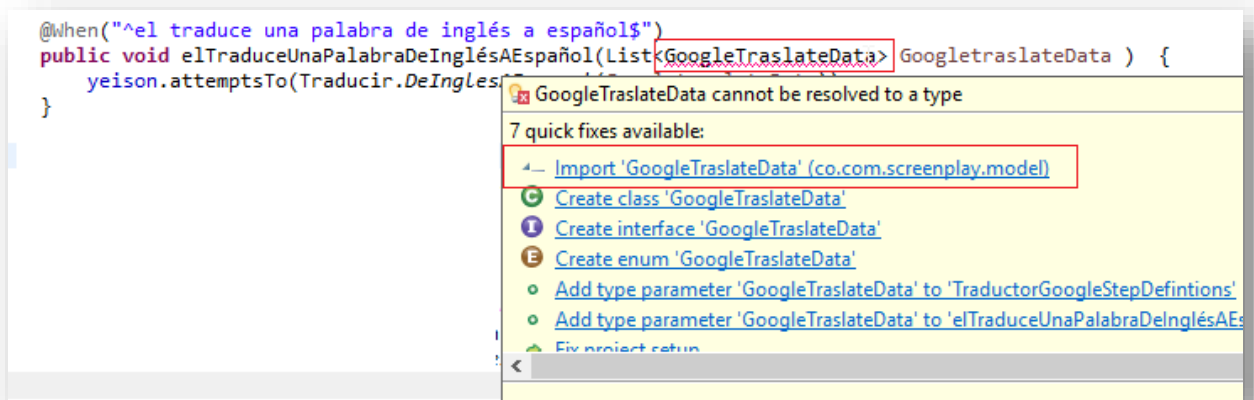
```
@When("^el traduce una palabra de inglés a español$")
public void elTraduceUnaPalabraDeInglésAEspañol(List<GoogleTraslateData> GoogletraslateData ) {
    yeison.attemptsTo(Traducir.DeIngLesAEspanol(GoogletraslateData));
}
```



Procedemos a realizar la importación de la siguientes librerías.



Y para la clase modela si:



De esta forma tendremos en esta lista toda la información que se tenía en la tabla del feature. Por último, cuando tengas un “**scenario outline**” y desees aplicar estos conceptos, podrás hacerlo armando la estructura de la tabla desde el “**feature**” como se muestra a continuación:

```
traductor_google_datadriven.feature
1 #Author: your.email@your.domain.com
2 Feature:
3     Como un usuario
4     Quiero ingresar a google Translator
5     A traducir palabras entre diferentes lenguajes
6
7     @traducir
8     Scenario Outline: Traducir de Ingles a Español
9         Given que Yeison quiere usar el traductor de google
10        When el traduce una palabra de inglés a español
11            |origen|destino|palabrai|
12            |<origen>|<destino>|<palabrai>|
13        Then el deberia ver la palabra traducida del idioma origen a idioma destino
14            |palabrae|
15            |<palabrae>|
16 Examples:
17     |origen|destino|palabrai|palabrae|
18     |inglés|español|table|mesa|
19
```

¡Ahora puedes implementarlo en tus proyectos!

