Dart Programming - Symbol

Symbols in Dart are opaque, dynamic string name used in reflecting out metadata from a library. Simply put, symbols are a way to store the relationship between a human readable string and a string that is optimized to be used by computers.

Reflection is a mechanism to get metadata of a type at runtime like the number of methods in a class, the number of constructors it has or the number of parameters in a function. You can even invoke a method of the type which is loaded at runtime.

In Dart reflection specific classes are available in the **dart:mirrors** package. This library works in both web applications and command line applications.

Syntax

```
Symbol obj = new Symbol('name');
// expects a name of class or function or library to reflect
```

The **name** must be a valid public Dart member name, public constructor name, or library name.

Example

Consider the following example. The code declares a class **Foo** in a library **foo_lib**. The class defines the methods **m1, m2,** and **m3**.

Foo.dart

```
library foo_lib;
// libarary name can be a symbol

class Foo {
   // class name can be a symbol
   m1() {
      // method name can be a symbol
      print("Inside m1");
   }
   m2() {
      print("Inside m2");
   }
   m3() {
      print("Inside m3");
   }
}
```

The following code loads **Foo.dart** library and searches for Foo class, with help of Symbol type. Since we are reflecting the metadata from the above library the code imports **dart:mirrors** library.

FooSymbol.dart

```
import 'dart:core';
import 'dart:mirrors';
import 'Foo.dart';

main() {
   Symbol lib = new Symbol("foo_lib");
   //library name stored as Symbol

   Symbol clsToSearch = new Symbol("Foo");
   // class name stored as Symbol

   if(checkIf_classAvailableInlibrary(lib, clsToSearch))

   // searches Foo class in foo_lib library
      print("class found..");
}

bool checkIf_classAvailableInlibrary(Symbol libraryName, Symbol className) {
   MirrorSystem mirrorSystem = currentMirrorSystem();
   LibraryMirror libMirror = mirrorSystem.findLibrary(libraryName);

   if (libMirror != null) {
      print("Found Library");
      print("checkng...class details..");
      print("No of classes found is : ${libMirror.declarations.length}");
      libMirror.declarations.forEach((s, d) => print(s));

      if (libMirror.declarations.containsKey(className)) return true;
      return false;
   }
}
```

Note that the line libMirror.declarations.forEach((s, d) => print(s)); will iterate across every declaration in the library at runtime and prints the declarations as type of **Symbol**.

This code should produce the following **output** –

```
Found Library
checkng...class details..
No of classes found is : 1
Symbol("Foo") // class name displayed as symbol
class found.
```

Example: Display the number of instance methods of a class

Let us now consider displaying the number of instance methods in a class. The predefined class **ClassMirror** helps us to achieve the same.

```
import 'dart:core';
import 'dart:mirrors';
import 'Foo.dart';

main() {
   Symbol lib = new Symbol("foo_lib");
   Symbol clsToSearch = new Symbol("Foo");
   reflect_InstanceMethods(lib, clsToSearch);
}
void reflect_InstanceMethods(Symbol libraryName, Symbol className) {
   MirrorSystem mirrorSystem = currentMirrorSystem();
   LibraryMirror libMirror = mirrorSystem.findLibrary(libraryName);

   if (libMirror != null) {
      print("Found Library");
      print("checkng...class details..");
      print("No of classes found is : ${libMirror.declarations.length}");
      libMirror.declarations.forEach((s, d) => print(s));
      libMirror.declarations.forEach((s, d) => print(s));

      if (libMirror.declarations.containsKey(className)) print("found class"
      ClassMirror classMirror = libMirror.declarations[className];

      print("No of instance methods found is ${classMirror.instanceMembers.le
      classMirror.instanceMembers.forEach((s, v) => print(s));
   }
}
```

This code should produce the following **output** –

```
Found Library
checkng...class details..
No of classes found is : 1
Symbol("Foo")
found class
No of instance methods found is 8
Symbol("==")
Symbol("hashCode")
Symbol("toString")
Symbol("noSuchMethod")
Symbol("runtimeType")
Symbol("m1")
Symbol("m2")
Symbol("m3")
```

Convert Symbol to String

You can convert the name of a type like class or library stored in a symbol back to string using **MirrorSystem** class. The following code shows how you can convert a symbol to a string.

Live Demo

```
import 'dart:mirrors';
void main(){
   Symbol lib = new Symbol("foo_lib");
   String name_of_lib = MirrorSystem.getName(lib);

   print(lib);
   print(name_of_lib);
}
```

It should produce the following **output** –

```
Symbol("foo_lib")

foo_lib
```