

Java Mapping for Constants

Here are the sample [constant definitions](#) once more:

Slice

```
const bool      AppendByDefault = true;
const byte      LowerNibble = 0x0f;
const string    Advice = "Don't Panic!";
const short     TheAnswer = 42;
const double    PI = 3.1416;

enum Fruit { Apple, Pear, Orange };
const Fruit     FavoriteFruit = Pear;
```

Here are the generated definitions for these constants:

Java

```
public interface AppendByDefault {
    boolean value = true;
}

public interface LowerNibble {
    byte value = 15;
}

public interface Advice {
    String value = "Don't Panic!";
}

public interface TheAnswer {
    short value = 42;
}

public interface PI {
    double value = 3.1416;
}

public interface FavoriteFruit {
    Fruit value = Fruit.Pear;
}
```

As you can see, each Slice constant is mapped to a Java interface with the same name as the constant. The interface contains a member named `value` that holds the value of the constant.

See Also

- [Constants and Literals](#)
- [Java Mapping for Identifiers](#)
- [Java Mapping for Modules](#)
- [Java Mapping for Built-In Types](#)
- [Java Mapping for Enumerations](#)
- [Java Mapping for Structures](#)
- [Java Mapping for Sequences](#)
- [Java Mapping for Dictionaries](#)
- [Java Mapping for Exceptions](#)