Java: API

Shirley B. Chu

June 16, 2020

De La Salle University
College of Computer Studies

API

- Application Programming Interface (API)
- In Java, it is a collection of packages, classes and interfaces that can be used in creating applications.
- Documentation of the Java API is also available online.

OVERVIEW MODULE PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

ALL CLASSES

Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification

This document is divided into two sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose c
start with `java`.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all
are in modules whose names start with `jdk`.

All Modules | **Java SE** | **JDK** | **Other Modules**

Module	Description
java.base	Defines the foundational APIs of the Java SE Platform.
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
java.datatransfer	Defines the API for transferring data between and within applications.
java.desktop	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, ir
java.instrument	Defines services that allow agents to instrument programs running on the JVM.
java.logging	Defines the Java Logging API.

Module java.base

Package java.util

Class Scanner

java.lang.Object
    java.util.Scanner

All Implemented Interfaces:

Closeable, AutoCloseable, Iterator<String>

```
public final class Scanner
extends Object
implements Iterator<String>, Closeable
```

A simple text scanner which can parse primitive types and strings using regular expressions.

A `Scanner` breaks its input into tokens using a delimiter pattern, which by default matches whitespac
types using the various `next` methods.

For example, this code allows a user to read a number from `System.in`:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

As another example, this code allows `long` types to be assigned from entries in a file `myNumbers`:

Module java.base

Package java.util

Class Scanner

java.lang.Object
    java.util.Scanner

All Implemented Interfaces:

Closeable, AutoCloseable, Iterator<String>

```
public final class Scanner
extends Object
implements Iterator<String>, Closeable
```

A simple text scanner which can parse primitive types and strings using regular expressions.

A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespac
types using the various next methods.

For example, this code allows a user to read a number from System.in:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

As another example, this code allows long types to be assigned from entries in a file myNumbers:

Module java.base

Package java.util → package to import

Class Scanner

java.lang.Object
    java.util.Scanner

All Implemented Interfaces:

Closeable, AutoCloseable, Iterator<String>

```
public final class Scanner
extends Object
implements Iterator<String>, Closeable
```

A simple text scanner which can parse primitive types and strings using regular expressions.

A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespac
types using the various next methods.

For example, this code allows a user to read a number from System.in:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

As another example, this code allows long types to be assigned from entries in a file myNumbers:

Module java.base

Package java.util

Class Scanner

java.lang.Object
    java.util.Scanner

All Implemented Interfaces:

`Closeable`, `AutoCloseable`, `Iterator<String>`

```
public final class Scanner
extends Object
implements Iterator<String>, Closeable
```

A simple text scanner which can parse primitive types and strings using regular expressions.

A `Scanner` breaks its input into toke description of the class hich by default matches whitespac
types using the various `next` methods.

For example, this code allows a user to read a number from `System.in`:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

As another example, this code allows `long` types to be assigned from entries in a file `myNumbers`:

Since:

1.5

Constructor Summary

Constructors

Constructor	Description
`Scanner(File source)`	Constructs a new Scanner tha
`Scanner(File source, String charsetName)`	Constructs a new Scanner tha
`Scanner(File source, Charset charset)`	Constructs a new Scanner tha
`Scanner(InputStream source)`	Constructs a new Scanner tha
`Scanner(InputStream source, String charsetName)`	Constructs a new Scanner tha
`Scanner(InputStream source, Charset charset)`	Constructs a new Scanner tha
`Scanner(Readable source)`	Constructs a new Scanner tha
`Scanner(String source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source, String charsetName)`	Constructs a new Scanner tha

Since:

1.5 → earliest JDK version

Constructor Summary

Constructors

Constructor	Description
`Scanner(File source)`	Constructs a new Scanner tha
`Scanner(File source, String charsetName)`	Constructs a new Scanner tha
`Scanner(File source, Charset charset)`	Constructs a new Scanner tha
`Scanner(InputStream source)`	Constructs a new Scanner tha
`Scanner(InputStream source, String charsetName)`	Constructs a new Scanner tha
`Scanner(InputStream source, Charset charset)`	Constructs a new Scanner tha
`Scanner(Readable source)`	Constructs a new Scanner tha
`Scanner(String source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source, String charsetName)`	Constructs a new Scanner tha

Since:

1.5

Constructor Summary

Constructors

Constructor	Description
`Scanner(File source)`	Constructs a new Scanner tha
`Scanner(File source, String charsetName)`	Constructs a new Scanner tha
`Scanner(File source, Charset charset)`	Constructs a new Scanner tha
`Scanner(InputStream source)`	Constructs a new Scanner tha
`Scanner(InputStream sourc`	a new Scanner tha
`Scanner(InputStream source, Charset charset)`	Constructs a new Scanner tha
`Scanner(Readable source)`	Constructs a new Scanner tha
`Scanner(String source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source, String charsetName)`	Constructs a new Scanner tha

available constructors, for instantiation

Since:

1.5

Constructor Summary

Constructors

Constructor	Description
`Scanner(File source)`	Constructs a new Scanner tha
`Scanner(File source, String charsetName)`	Constructs a new Scanner tha
`Scanner(File source, Charset charset)`	Constructs a new Scanner tha
`Scanner(InputStream source)`	Constructs a new Scanner tha
`Scanner(InputStream source, String charsetName)`	Constructs a new Scanner tha
`Scanner(InputStream source, Charset charset)`	Constructs a new Scanner tha
`Scanner(Readable source)`	Constructs a new Scanner tha
`Scanner(String source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source, String charsetName)`	Constructs a new Scanner tha

Since:

1.5

Constructor Summary

Constructors

Constructor	Description
`Scanner(File source)`	Constructs a new Scanner tha
`Scanner(File source, String charsetName)`	Constructs a new Scanner tha
`Scanner(File source, Charset charset)`	Constructs a new Scanner tha
`Scanner(InputStream source)`	s a new Scanner tha
`Scanner(InputStream source, String charsetName)`	Constructs a new Scanner tha
`Scanner(InputStream source, Charset charset)`	Constructs a new Scanner tha
`Scanner(Readable source)`	Constructs a new Scanner tha
`Scanner(String source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source)`	Constructs a new Scanner tha
`Scanner(ReadableByteChannel source, String charsetName)`	Constructs a new Scanner tha

`System.in` (keyboard)

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	close()	Closes this scanner.
Pattern	delimiter()	[illegible]
String	next()	Finds and returns the next complete tol
boolean	nextBoolean()	Scans the next token of the input into a
byte	nextByte()	Scans the next token of the input as a b
byte	nextByte(int radix)	Scans the next token of the input as a b
double	nextDouble()	Scans the next token of the input as a d
float	nextFloat()	Scans the next token of the input as a f
int	nextInt()	Scans the next token of the input as an
int	nextInt(int radix)	Scans the next token of the input as an
String	nextLine()	Advances this scanner past the current

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	close()	Closes this scanner.
Pattern	delimiter()	Returns the Pattern this Scanner is
String	next()	Finds and returns the next complete tol
boolean	nextBoolean()	Scans the next token of the input into a
byte	nextByte()	Scans the next token of the input as a b
byte	nextByte(int radix)	Scans the next token of the input as a b
double	nextDouble()	Scans the next token of the input as a d
float	nextFloat()	Scans the next token of the input as a f
int	nextInt()	Scans the next token of the input as an
int	nextInt(int radix)	Scans the next token of the input as an
String	nextLine()	Advances this scanner past the current

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	close()	Closes this scanner.
Pattern	delimiter()	[illegible]
String	next()	Finds and returns the next complete tol
boolean	nextBoolean()	Scans the next token of the input into a
byte	nextByte()	Scans the next token of the input as a b
byte	nextByte(int radix)	Scans the next token of the input as a b
double	nextDouble()	Scans the next token of the input as a d
float	nextFloat()	Scans the next token of the input as a f
int	nextInt()	Scans the next token of the input as an
int	nextInt(int radix)	Scans the next token of the input as an
String	nextLine()	Advances this scanner past the current

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	close()	Closes this scanner.
Pattern	delimiter()	[illegible]
String	next()	Finds and returns the next complete to
boolean	nextBoolean()	Scans the next token of the input into a
byte	nextByte()	Scans the next token of the input as a b
byte	nextByte(int radix)	Scans the next token of the input as a b
double	nextDouble()	Scans the next token of the input as a d
float	nextFloat()	Scans the next token of the input as a f
int	nextInt()	Scans the next token of the input as an
int	nextInt(int radix)	Scans the next token of the input as an
String	nextLine()	Advances this scanner past the current

```
import java.util.*;
public class SimpleSample
{
    public static void main (String[] args)
    {
        Scanner kb = new Scanner (System.in);
        int nOne, nTwo;

        System.out.print ("Enter a number:  ");
        nOne = kb.nextInt ();


        System.out.print ("Enter another number:  ");
        nTwo = kb.nextInt ();

        System.out.println ("Sum is " + (nOne + nTwo));
        kb.close ();
    }
}
```

```
import java.util.*;
public class SimpleSample
{
    public static void main (String[] args)
    {
        Scanner kb = new Scanner (System.in);
        int nOne, nTwo;

        System.out.print ("Enter a number:  ");
        nOne = kb.nextInt ();


        System.out.print ("Enter another number:  ");
        nTwo = kb.nextInt ();

        System.out.println ("Sum is " + (nOne + nTwo));
        kb.close ();
    }
}
```

Package java.util

Class Scanner

```
import java.util.*;
public class SimpleSample
{
    public static void main (String[] args)
    {
        Scanner kb = new Scanner (System.in);
        int nOne, nTwo;

        System.out.print ("Enter a number:  ");
        nOne = kb.nextInt ();


        System.out.print ("Enter another number:  ");
        nTwo = kb.nextInt ();

        System.out.println ("Sum is " + (nOne + nTwo));
        kb.close ();
    }
}
```

Package java.util

Class Scanner

```java
import java.util.*;
public class SimpleSample
{
    public static void main (String[] args)
    {
        Scanner kb = new Scanner (System.in);
        int nOne, nTwo;

        System.out.print ("Enter a number:  ");
        nOne = kb.nextInt ();


        System.out.print ("Enter another number:  ");
        nTwo = kb.nextInt ();

        System.out.println ("Sum is " + (nOne + nTwo));
        kb.close ();
    }
}
```

```java
Scanner(InputStream source)
```

```java
import java.util.*;
public class SimpleSample
{
    public static void main (String[] args)
    {
        Scanner kb = new Scanner (System.in);
        int nOne, nTwo;

        System.out.print ("Enter a number:  ");
        nOne = kb.nextInt ();
```

```
int      nextInt()
```

```java
        System.out.print ("Enter another number:  ");
        nTwo = kb.nextInt ();

        System.out.println ("Sum is " + (nOne + nTwo));
        kb.close ();
    }
}
```

```
import java.util.*;
public class SimpleSample
{
    public static void main (String[] args)
    {
        Scanner kb = new Scanner (System.in);
        int nOne, nTwo;

        System.out.print ("Enter a number:  ");
        nOne = kb.nextInt ();


        System.out.print ("Enter another number:  ");
        nTwo = kb.nextInt ();

        System.out.println ("Sum is " + (nOne + nTwo));
        kb.close ();
    }
}
```

```
void        close()
```

☺ **Thank you!** ☺