

I'm not robot  reCAPTCHA

Continue

Byte to string android

Public final class bytes can extend the number implementation similar to<code>java.lang.Object</code>↳<code>java.lang.number</code>↳<code>java.lang.byte</code>.Byte class to lapping the value of raw type bytes in an object. Byte-type objects contain a single field with a byte type. The class also provides several ways to convert bytes to strings and strings, as well as other constants and methods that are useful when processing bytes. int BYTES The number of bytes used to represent byte values in the form of two complementary binaries. Byte MAX_VALUE holds the maximum value a byte can have, 27-1. A constant MIN_VALUE -27 that holds the minimum value a byte can have. int SIZE The number of bits used to represent byte values in the form of two complementary binaries. A public static final class<code>Byte</code>; type class instance that represents the default type byte. Byte (byte value) Generates a newly allocated byte object that represents the specified byte value. Bytes (strings) generate newly assigned byte objects that represent byte values represented by string parameters. Byte Byte () Returns the value of this byte in bytes. Static int comparison (byte x, byte y) compares two byte values by numbers. Int compareTo (bytes another byte) number comparing two byte objects. Decodes static byte decoding (string nm) strings bytes. Returns this byte as a double value after a double valueate () magnified raw conversion. Boolean equals (object obj) Compares this object to the specified object. Float Float Value () Returns this byte value as a float after a magnified raw transformation. int Hashcode () Returns the hash code for this byte. Same as the result of calling int value (). A static int hashcode (byte value) returns a hash code for a byte value. Bytes.Compatible with hashcode(). IntValue() Returns the value of this byte as int after a magnified raw conversion. Returns the value of this byte long after a long Value() magnified raw transformation. Static byte parsing Bytes (String s) parses string arguments with signed bytes of decimal. String s (int radix) parses string arguments bytes signed from the radix specified by the second argument. A short short value () returns the value of this byte short after the enlarged raw conversion. String toString() Returns a string object that represents this byte value. Static string toString(byte b) Returns a new string object that represents the specified byte. Static int toUnsignedInt (byte x) converts arguments to int by unsigned renditions. Long (byte x), which is unserrd at static length, converts arguments long to unserrd transformations. Static byte valueOfOf (string) returns a byte object that holds a given value in the specified string. Returns a byte object that holds a value extracted from a static byte value (string s, int radix)<code>Byte</code>; <code>Byte</code>;Specify a string when parsing with a given radix in the second argument. Static byte value (byte b) Returns a byte instance that represents the specified byte value. Class java.lang.number byteValue() Returns the value of a number specified by bytes associated with rounding or severing. Returns the value of a specified number that can contain abstract double value () rounding. Abstract float value () Returns the value of a number specified by a float that may involve rounding. Abstract intValue() Returns the value of a specified number associated with rounding or sminging as int. Abstract Long Value() Returns the value of a specified number that can contain rounding or sminging. Short Short Value () Returns the value of a specified number associated with rounding or cutting short. The class java.lang.Object Clone () makes and returns a copy of this object. Equal Boolean (object obj) indicates whether another object is equal to this object. Void finish () When a gabage collector determines that there is no longer a reference to the object, it is called from the object's savage collector. Final Class <code>?</code> <code>getClass()</code> Returns the runtime class of this object. Returns a string representation of a string toString() object. Final void wait (long time, int nanos) The current thread will wait until another thread calls the notification () or notifyAll() method for this object, or some other thread interferences with the current thread or becomes a certain amount of real-time. The final void wait (long time) waits for the current thread to pass, or for another thread to call the notification () method or notifyAll() method for this object. Wait for the final void () The current thread will wait until another thread calls the notification () method or notifyAll() method for this object. Constant public static final INT BYTES The number of bytes used to represent byte values in the form of two complementary binaries. Constant value: 1 (0x0000000001) Public static final byte MAX_VALUE constant holding the maximum value that a byte can have, 27-1. Constant value: 127 (0x000000007f) is a constant that holds the minimum MIN_VALUE a public static final byte or byte, -27. Constant value: -128 (0xfffffff80) The number of bits used to represent byte values in the form of two complementary binaries in a public static final int SIZE. Constant value: 8 (0x0000000008) field public static final class<code>Byte</code>; type a class instance that represents the base type byte. Public Constructor Public Bytes (Byte Values) Newly Configured <code>Byte</code>;A byte object that represents the specified byte value. Parameter value bytes: The value to be expressed in bytes. Public Bytes (String s) generates a newly assigned byte object that represents the byte value represented by the string parameter. The string is converted to a byte value in exactly the way that the parseByte method for radix 10 uses. Parameter String: The string to be converted to a byte reference. parseByte (java.lang.String, int) public method Returns this byte value as a public byte byte (). Converts the numeric value represented by this object to bytes after conversion. Public static int comparison (byte x, byte y) compares two byte values by numbers. The returned value is the Byte.valueOf(x).compareTo (Byte.valueOf(y)) parameter x bytes: the second byte compares the value 0 if x == y. A value of x <code>y</code> is less than 0, and x <code>y</code> public int comparison (Byte anotherByte) is greater than 0 when comparing two byte objects by numbers. Parameter Different byte bytes: You can compare bytes. If this byte is equal to the argument byte, it returns a value of 0. If this byte is less than the argument byte, the value is less than 0. If this byte is numericly larger than the argument byte (signed comparison), the value is greater than 0. Decodes a public static byte decoding (string nm) string in bytes. Accepts decimal points, hexagons, and eighty numbers given by the following grammar: DecodableString: Signature per signature decimal number0x six-digit number 0X six-digit number signal selection # HexDigits Signopt 0 OctalDigits symbol: - + decimal point. The naked eye, Opal Digitals, and OctalDigits are not defined in section 3.10.1 of the Java language™ and character sequences that follow optional symbols and/or radix classifiers (0x, 0X, # 또는 leading 0) are parsed as in the Byte.parseByte method with marked radix (10, 16, or 8). This sequence of characters must represent a positive value and is thrown outside of the NumberFormat example. If the first character in the specified string is a minus symbol, the result is invalidated. White space characters are not allowed in strings. Parameter nm string: The string to decode. Returns a byte object that holds the byte value represented by the nm reference: parseByte (java.lang.String, int) public double value () returns the value of this byte as a double value after the enlarged raw conversion. After conversion, double the numeric value represented by this object and convert it to double. The public Boolean compares this object to the specified object. The result is only if the argument is not null and is a byte object that contains the same byte value as this object. Parameter obj object: object to compare to return The fact that the objects are the same; Otherwise false. Returns this byte value as a float after a public float float value () magnified raw conversion. After conversion, plot the numeric values represented by this object and convert them to float types. Public int hashcode () returns the hash code for this byte; Same as the result of calling int value (). If you return a hash code value for this byte public static hashcode () returns the hash code for the byte value. Bytes.Compatible with hashcode(). Parameter value byte: The value that returns the hash code value for the byte value. Public int intValue (returns the value of this byte as int) after the enlarged raw conversion. The numeric value displayed by this object after conversion is int. Returns the value of this byte long after the public long value () magnified raw conversion. Returns the numeric value represented by this object as a long input after conversion. Public static byte parsing (string s) parses string arguments with signed bytes of decimal. All characters in the string must be decimal places, except that the first character can be the ASCII minus sign '-' (u002D) to represent a negative value or ASCII plus '+' (u002B). The resulting byte value is returned exactly as the argument and radix 10 provided as arguments to the parseByte (java.lang.String, int) method. Parameter s string: A string containing a separate representation parses the string argument bytes signed from the radix specified by the second argument by separating the byte value represented by the argument from the decimal public static byte pass byte (string s, int radix). The characters in the string must all be numeric, and depending on whether the number in the specified radix (character digit (char, int) returns a non-negative value, the first character can represent a positive value to represent a negative value or ASCII plus '+' (u002B), except that the first character may be the ASCII minus sign '-' (u002D). The resulting byte value is returned. If some of the following situations occur, an exception of the numberformat example type is thrown: the first argument is null or a string of length 0. Radix is less than Character.MIN_RADIX or a character. MAX_RADIX more than 36. All characters in a string are not numbers in the specified radix, except that the first character can be the minus sign '-' (u002D) or '+' (u002B) if the string is longer than length 1. The value represented by the string is not a byte-type value. Parameter s string: A string containing a byte representation to parse the line: Tadic uses radix to use while parsing byte values represented by string arguments at a given radix public short value () byte bytes. After the enlarged raw conversion. Returns the numeric value represented by this object after conversion in a short format with a short value. Public string toString() Returns a string object that represents this byte value. The value is converted to a signed decimal representation and the byte value is returned as a string as provided as an argument to the toString (byte) method. Returns a string representation of the value of this object as a string in the default 10. The public static string toString (byte b) returns a new string object that represents the specified byte. Radix is estimated to be 10. Parameter b bytes: The bytes to be converted are also converted to a string string representation of the specified byte: the argument is converted to int by a public static int toUnsignedInt(byte x) unsigned conversion. In a conversion that is not unnamed as int, the high order 24 bits of the int are 0, and the lower order 8 bits are the same as the bits of the byte argument. Therefore, the 0 and fudji byte values are numerically mapped to the same int value, and the negative byte value is mapped to the same int value as input plus 28. Parameter x bytes: The value of converting to unsigned int Return converts the argument converted to int by an unsigned conversion to intototunsignedLong (byte x), converting the argument to a long one with an unsigned rendition. In an unseged conversion, a long 56-bit high-order 56-bit is 0, and a suborder 8 bit is the same as a bit of a byte argument. Therefore, the value of 0 and footage bytes is mapped to the same long value as the number, and the negative byte value is mapped to the same long value as input plus 28. Parameter x bytes: An unserrd transition to a public static byte value that converts an argument converted to a long, unserrd long return returns a byte object that holds a given value in the specified string. Arguments are interpreted to represent bytes of a signed decimal point just as arguments were provided in the state (java.lang.String) method. The result is a byte object that represents the byte value specified in the string. In other words, this method returns the same byte object as the value of the new byte (Byte.parseByte(s)) parameter string: a string (string s, int radix) that parses a byte byte object that holds a value represented by a string argument. The first argument is interpreted as representing a byte signed in the radix specified by the second argument, and the argument is interpreted exactly as provided in the parseByte (java.lang.String, int) method. The result is a byte object that represents the byte value specified in the string. In other words, this method returns the same object as the byte object. Of: A string of new bytes (Byte.parseByte(s, radix)) parameters: a line of radix int: a radix that is used to interpret the return byte object of a byte holding a value represented by the string argument of the specified radix. Public static byte value (byte b) Returns a byte instance that represents the specified byte value. If you don't need a new byte instance, this method should typically be used in the default settings for constructor Byte (bytes), because all byte values are cached, which has the potential to provide much better space and time performance. Parameter b byte: Byte value. returns a .b instance that represents b.b.

