



# Java API

Last edited time	@March 19, 2025 9:53 AM
Summary	(1) class/inheritance (2) API

## Class/Interface

Aa Name	</> Type	Package	Inheritance	Description	proper
<u>Set&lt;E&gt;</u>	Interface	java.util	Set → Collection → Object	不允許重複元素的集合。	HashSet (排序)、Tree (排序)、LinkedHas (插入順序)
<u>List&lt;E&gt;</u>	Interface	java.util	List → Collection → Object	允許重複元素，元素有索引，可依順序存取。	ArrayList (列實作)、LinkedList (結串列實作)
<u>Queue&lt;E&gt;</u>	Interface	java.util	Queue → Collection → Object	先進先出 (FIFO) 的資料結構。	LinkedList, ArrayDeque
<u>Properties</u>	Class	java.util	Properties → Hashtable<Object, Object> → Map → Object	Key-Value 組合，用於讀取設定檔 (.properties)。	Hashtable 類，Key Value 皆為 String，使用 .properties 件。
<u>Collection&lt;E&gt;</u>	Interface	java.util	Collection → Object	容納多個元素的集合介面，提供基本操作方法。	主要子介面：List (有順序)、Set (無重複)、Queue (FIFO)。
<u>Map&lt;K, V&gt;</u>	Interface	java.util	Map → Object	以 Key-Value 形式儲存資料，Key 唯一，不允許重複。	HashMap (排序)、Tree (排序)、LinkedHas (插入順序)
<u>Math</u>	Class	java.lang	Math → Object	數學運算類別，提供常見的數學函數，如指數、對數、三角函數等。	final 類別，無法實例化；方法皆為 static。
<u>System</u>	Class	java.lang	System → Object	提供標準輸入/輸出、系統屬性、時間計算等功能。	final 類別，無法實例化；屬性：System.out, System.err, System.in
<u>StringBuffer</u>	Class	java.lang	StringBuffer → Object	可變字串類別，允許修改字串內容 (增、刪、改)。適合頻繁變動的字串操作。	預設容量 16，可自動擴展；支援 append(), insert(), replace(), delete()。

Aa Name	</> Type	Package	Inheritance	Description	proper
<b>Spring data jpa</b>	Interface	org.springframework.data.jpa.repository	JpaRepository<T, ID> → PagingAndSortingRepository<T, ID> → CrudRepository<T, ID>	支援 <b>Specification</b> API與動態查詢方法，例如 findByUsernameAndEmail(String username, String email)	

## Java API Methods

Return	Aa Name	Static	Parameters	Class	Package	Description
int	<b>compareTo</b>	<input type="checkbox"/>	T o	Comparable<T>	java.lang	比較當前物件與指定物件的大小，返回負數、0 或正數
Double	<b>valueOf</b>	<input type="checkbox"/>	String str	Double	java.lang	返回一個表示指定字符串的 Double 物件。將字符串解析為 double 值並包裝在 Double 物件中。
double	<b>doubleValue</b>	<input type="checkbox"/>	()	Double	java.lang	將 Double 物件轉換為 double 基本數據類型。返回由 Double 物件表示的 double 值。
long	<b>Math.round</b>	<input checked="" type="checkbox"/>	double d	Math	java.lang	四捨五入 d，回傳最接近的長整數值。通常用來處理小數點後的精度。
boolean	<b>equals</b>	<input type="checkbox"/>	Object obj	Object	java.lang	比較兩個物件是否相等。如果兩物件屬於同一個類別，則比較它們的屬性值是否相同。
Class	<b>getClass</b>	<input type="checkbox"/>	()	Object	java.lang	返回該物件的類型，返回 Class 型別的物件，表示該物件的類別資訊。
int	<b>hashCode</b>	<input type="checkbox"/>	()	Object	java.lang	返回物件的 hash code，用來比較物件是否相等。通常與 equals() 方法搭配使用。
String	<b>toString</b>	<input type="checkbox"/>	()	Object	java.lang	返回物件的字串表示，通常格式為 getClass().getName() + "@" + Integer.toHexString(hashCode())。
String	<b>substring</b>	<input type="checkbox"/>	int x, int y	String	java.lang	取得字串從 x 到 y-1 的子字串。
int	<b>indexOf</b>	<input type="checkbox"/>	String st1	String	java.lang	回傳 st1 在字串中第一次出現的索引。
String	<b>trim</b>	<input type="checkbox"/>	()	String	java.lang	移除字串前後的空白。
int	<b>lastIndexOf</b>	<input type="checkbox"/>	String st1	String	java.lang	回傳 st1 在字串中最後一次出現的索引。
boolean	<b>endsWith</b>	<input type="checkbox"/>	String suffix	String	java.lang	檢查字串是否以指定的 suffix 結尾。
String[]	<b>split</b>	<input type="checkbox"/>	String regex	String	java.lang	根據指定的正則表達式 (regex) 將字串拆分成多個子字串，並返回包含這些子字串的字串陣列 (String[])。
StringBuffer	<b>append</b>	<input type="checkbox"/>	String str	StringBuffer	java.lang	將指定的字串 str 添加到 StringBuffer 的末尾。
StringBuffer	<b>reverse</b>	<input type="checkbox"/>	()	StringBuffer	java.lang	反轉 StringBuffer 中的字串。

Return	Name	Static	Parameters	Class	Package	Description
StringBuffer	<a href="#">insert</a>	<input type="checkbox"/>	(int offset, String str)	StringBuffer	java.lang	在指定的偏移位置 <code>offset</code> 插入字串 <code>str</code> 。
StringBuffer	<a href="#">delete</a>	<input type="checkbox"/>	(int start, int end)	StringBuffer	java.lang	刪除 <code>start</code> 到 <code>end</code> 位置的字元。
StringBuffer	<a href="#">deleteCharAt</a>	<input type="checkbox"/>	int index	StringBuffer	java.lang	刪除指定位置 <code>index</code> 的單一字元。
StringBuffer	<a href="#">replace</a>	<input type="checkbox"/>	(int start, int end, String str)	StringBuffer	java.lang	將從 <code>start</code> 開始到 <code>end</code> （不包含 <code>end</code> ）的字元替換為字串 <code>str</code> 。
void	<a href="#">setCharAt</a>	<input type="checkbox"/>	(int index, char ch)	StringBuffer	java.lang	將索引 <code>index</code> 的字元替換成 <code>ch</code>
int	<a href="#">capacity</a>	<input type="checkbox"/>	()	StringBuffer	java.lang	傳回此 <code>StringBuffer</code> 物件的容量
void	<a href="#">e.printStackTrace</a>	<input type="checkbox"/>	()	Throwable	java.lang	打印出錯誤的堆疊跟蹤信息，幫助定位異常發生的地方。
boolean	<a href="#">add</a>	<input type="checkbox"/>	(E e)	Collection<E>	java.util	在集合內新增元素 <code>e</code> 。
boolean	<a href="#">containsAll</a>	<input type="checkbox"/>	(Collection<?> c)	Collection<E>	java.util	檢查集合是否包含 <code>c</code> 內的所有元素。
Collection<?> c	<a href="#">iterator</a>	<input type="checkbox"/>	()	Collection<E>	java.util	傳回 <code>Iterator</code> 物件，用於遍歷集合。
boolean	<a href="#">retainAll</a>	<input type="checkbox"/>	Collection<?> c	Collection<E>	java.util	只保留 <code>c</code> 內的元素，其餘刪除。
Object[]	<a href="#">toArray</a>	<input type="checkbox"/>	()	Collection<E>	java.util	將集合轉換為 <code>Object</code> 陣列。
T[]	<a href="#">toArray</a>	<input type="checkbox"/>	T[] a	Collection<E>	java.util	將集合轉換為陣列，若 <code>a</code> 足夠大則存入 <code>a</code> ，否則新建陣列存放元素。
int	<a href="#">compare</a>	<input type="checkbox"/>	(T o1, T o2)	Comparator<T>	java.util	比較兩個物件的大小，返回負數、0 或正數
boolean	<a href="#">isLeapYear</a>	<input type="checkbox"/>	int year	GregorianCalendar	java.util	判斷 <code>year</code> 是否為閏年，若是回傳 <code>true</code> ，否則回傳 <code>false</code> 。
boolean	<a href="#">hasNext</a>	<input type="checkbox"/>	()	Iterator<E>	java.util	判斷是否還有下一個元素可取出。
E	<a href="#">next</a>	<input type="checkbox"/>	()	Iterator<E>	java.util	取出集合內的下一個元素。
void	<a href="#">remove</a>	<input type="checkbox"/>	()	Iterator<E>	java.util	從集合內移除最近取出的元素。
boolean	<a href="#">containsKey</a>	<input type="checkbox"/>	Object key	Map<K,V>	java.util	檢查 Map 是否包含指定的 Key。
boolean	<a href="#">containsValue</a>	<input type="checkbox"/>	Object value	Map<K,V>	java.util	檢查 Map 是否包含指定的 Value。
Set<Map.Entry<K,V>>	<a href="#">entrySet</a>	<input type="checkbox"/>	()	Map<K,V>	java.util	傳回包含所有 Key-Value 對的 Set 集合。
Set<K>	<a href="#">keySet</a>	<input type="checkbox"/>	()	Map<K,V>	java.util	傳回所有 Key 組成的 Set 集合。
void	<a href="#">putAll</a>	<input type="checkbox"/>	(Map<? extends K, ? extends V> m)	Map<K,V>	java.util	新增多對 Key-Value 進入 Map。
boolean	<a href="#">isEmpty</a>	<input type="checkbox"/>	()	Map<K,V>	java.util	檢查 Map 是否為空。
void	<a href="#">remove</a>	<input type="checkbox"/>	Object key	Map<K,V>	java.util	移除指定 Key 及對應的 Value。
void	<a href="#">put</a>	<input type="checkbox"/>	(K key, V value)	Map<K,V>	java.util	新增一對 Key-Value 進入 Map。
V	<a href="#">get</a>	<input type="checkbox"/>	Object key	Map<K,V>	java.util	取得指定 Key 對應的 Value。

Return	Name	Static	Parameters	Class	Package	Description
Collection<V>	<a href="#">values</a>	<input type="checkbox"/>	()	Map<K,V>	java.util	取得 Map 內的所有 Value，並以 Collection 形式返回。
T	<a href="#">orElseThrow</a>	<input type="checkbox"/>	Supplier<? extends X> exceptionSupplier	Optional<T>	java.util	避免 null 錯誤，當 Optional 物件為空時拋出異常
boolean	<a href="#">instanceof</a>	<input type="checkbox"/>	物件 instanceof 類別	Operators	KeyWords	在 運行時 檢查某個物件是否是某個類別或其子類別的實例，返回 true 或 false。
<T extends Throwable> T	<a href="#">assertThrows</a>	<input type="checkbox"/>	(Class<T> expectedType, Executable executable)	Assertions	org.junit.jupiter.api	測試是否拋出特定異常
void	<a href="#">assertEquals</a>	<input type="checkbox"/>	(expected, actual, message)	Assertions	org.junit.jupiter.api	斷言兩個值相等
S	<a href="#">save</a>	<input type="checkbox"/>	S entity	Entity	spring-data-jpa	保存一個實體，若已存在則更新
Iterable<S>	<a href="#">saveAll</a>	<input type="checkbox"/>	Iterable<S> entities	Entity	spring-data-jpa	批量保存多個實體
Iterable<S>	<a href="#">findAll</a>	<input type="checkbox"/>	()	Entity	spring-data-jpa	查詢所有實體
Optional<S>	<a href="#">findById</a>	<input type="checkbox"/>	ID id	Entity	spring-data-jpa	根據 ID 查詢單個實體
Iterable<S>	<a href="#">findAllById</a>	<input type="checkbox"/>	Iterable<ID> ids	Entity	spring-data-jpa	根據一組 ID 查詢對應的實體
void	<a href="#">deleteAllById</a>	<input type="checkbox"/>	(Iterable<? extends ID> ids)	Entity	spring-data-jpa	根據一組 ID 刪除多個實體
void	<a href="#">deleteById</a>	<input type="checkbox"/>	ID id	Entity	spring-data-jpa	根據 ID 刪除一個實體
void	<a href="#">delete</a>	<input type="checkbox"/>	T entity	Entity	spring-data-jpa	刪除指定的實體
void	<a href="#">deleteAll</a>	<input type="checkbox"/>	(Iterable<? extends T> entities)	Entity	spring-data-jpa	批量刪除多個實體
void	<a href="#">deleteAll</a>	<input type="checkbox"/>	()	Entity	spring-data-jpa	刪除所有實體
long	<a href="#">count</a>	<input type="checkbox"/>	()	Entity	spring-data-jpa	查詢資料庫中實體的數量
boolean	<a href="#">existsById</a>	<input type="checkbox"/>	ID id	Entity	spring-data-jpa	判斷某個 ID 是否存在對應的實體
Page<T>	<a href="#">findAll</a>	<input type="checkbox"/>	Pageable pageable	Entity	spring-data-jpa	分頁查詢所有實體
Iterable<T>	<a href="#">findAll</a>	<input type="checkbox"/>	Specification<T> spec	Entity	spring-data-jpa	使用 JPA 的規範查詢 (Specification) 進行查詢
Iterable<T>	<a href="#">findAll</a>	<input type="checkbox"/>	Sort sort	Entity	spring-data-jpa	根據排序條件查詢所有實體
Optional<T>	<a href="#">findOne</a>	<input type="checkbox"/>	Specification<T> spec	Entity	spring-data-jpa	使用規範查詢獲得一個實體
void	<a href="#">deleteAllInBatch</a>	<input type="checkbox"/>	()	Entity	spring-data-jpa	批量刪除所有實體
void	<a href="#">deleteInBatch</a>	<input type="checkbox"/>	(Iterable<T> entities)	Entity	spring-data-jpa	批量刪除指定的多個實體
Iterable<T>	<a href="#">findAll</a>	<input type="checkbox"/>	(Specification<T> spec, Sort sort)	Entity	spring-data-jpa	使用規範查詢和排序條件相結合
Page<T>	<a href="#">findAll</a>	<input type="checkbox"/>	(Specification<T> spec, Pageable pageable)	Entity	spring-data-jpa	使用規範查詢和分頁查詢相結合
S	<a href="#">saveAndFlush</a>	<input type="checkbox"/>	(S entity)	Entity	spring-data-jpa	保存並立即刷新資料庫
void	<a href="#">flush</a>	<input type="checkbox"/>	()	Entity	spring-data-jpa	刷新當前上下文，確保所有的變更已被同步到資料庫
Optional<T>	<a href="#">findByIdAndStatus</a>	<input type="checkbox"/>	(ID id, String status)	Entity	spring-data-jpa	根據 ID 和其他條件查詢 (可自定義方法)

