BackupDataInput

added in API level 8
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Summary: Methods (#pubmethods) | Inherited Methods (#inhmethods) | [Expand All] (#)

```
public class BackupDataInput
extends Object (https://developer.android.com/reference/java/lang/Object.html)
```

java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)
↳ android.app.backup.BackupDataInput

Provides the structured interface through which a `BackupAgent` (https://developer.android.com/reference/android/app/backup/BackupAgent.html) reads information from the backup data set, via its `onRestore()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor)) method. The data is presented as a set of "entities," each representing one named record as previously stored by the agent's `onBackup()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor)) implementation. An entity is composed of a descriptive header plus a byte array that holds the raw data saved in the remote backup.

The agent must consume every entity in the data stream, otherwise the restored state of the application will be incomplete.

Example

A typical `onRestore()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor)) implementation might be structured something like this:

```
public void onRestore(BackupDataInput data, int appVersionCode,
                      ParcelFileDescriptor newState) {
    while (data.readNextHeader()) {
        String key = data.getKey();
        int dataSize = data.getDataSize();

        if (key.equals(MY_BACKUP_KEY_ONE)) {
            // process this kind of record here
            byte[] buffer = new byte[dataSize];
            data.readEntityData(buffer, 0, dataSize); // reads the entire entity at once

            // now 'buffer' holds the raw data and can be processed however
            // the agent wishes
            processBackupKeyOne(buffer);
        } else if (key.equals(MY_BACKUP_KEY_TO_IGNORE) {
            // a key we recognize but wish to discard
            data.skipEntityData();
        } // ... etc.
    }
}
```

Summary

Public methods	
`int`	`getDataSize` (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#getDataSize())`()` Report the size in bytes of the data associated with the current entity in the restore stream.
`String` (https://developer.android.com/reference/java/lang/String.html)	`getKey` (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#getKey())`()` Report the key associated with the current entity in the restore stream
`int`	`readEntityData` (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#readEntityData(byte[], int, int))`(byte[] data, int offset, int size)` Read a record's raw data from the restore stream.
`boolean`	`readNextHeader`

	(https://developer.android.com/reference/android/app/backup/BackupDataInput.html#readNextHeader())() Extract the next entity header from the restore stream.
void	skipEntityData (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#skipEntityData())() Consume the current entity's data without extracting it into a buffer for further processing.

Inherited methods
(#)From class java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)

Public methods

getDataSize

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int getDataSize ()
```

Report the size in bytes of the data associated with the current entity in the restore stream.

Returns	
int	The size of the record's raw data, in bytes

Throws	
IllegalStateException (https://developer.android.com/reference/java/lang/IllegalStateException.html)	if the next record header has not yet been read

getKey

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
String (https://developer.android.com/reference/java/lang/String.html) getKey ()
```

Report the key associated with the current entity in the restore stream

Returns	
String (https://developer.android.com/reference/java/lang/String.html)	the current entity's key string

Throws	
IllegalStateException (https://developer.android.com/reference/java/lang/IllegalStateException.html)	if the next record header has not yet been read

readEntityData

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int readEntityData (byte[] data,
                int offset,
                int size)
```

Read a record's raw data from the restore stream. The record's header must first have been processed by the readNextHeader() (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#readNextHeader()) method. Multiple calls to this method may be made in order to process the data in chunks; not all of it must be read in a single call. Once all of the raw data for the current entity has been read, further calls to this method will simply return zero.

Parameters	
data	byte: An allocated byte array of at least 'size' bytes

`offset`	`int`: Offset within the 'data' array at which the data will be placed when read from the stream
`size`	`int`: The number of bytes to read in this pass

Returns	
`int`	The number of bytes of data read. Once all of the data for this entity has been read, further calls to this method will return zero.

Throws	
`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an error occurred when trying to read the restore data stream

readNextHeader

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
boolean readNextHeader ()
```

Extract the next entity header from the restore stream. After this method return success, the `getKey()` (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#getKey()) and `getDataSize()` (https://developer.android.com/reference/android/app/backup/BackupDataInput.html#getDataSize()) methods can be used to inspect the entity that is now available for processing.

Returns	
`boolean`	`true` when there is an entity ready for consumption from the restore stream, `false` if the restore stream has been fully consumed.

Throws	
`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an error occurred while reading the restore stream

skipEntityData

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void skipEntityData ()
```

Consume the current entity's data without extracting it into a buffer for further processing. This allows a `BackupAgent` (https://developer.android.com/reference/android/app/backup/BackupAgent.html) to efficiently discard obsolete or otherwise uninteresting records during the restore operation.

Throws	
`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an error occurred when trying to read the restore data stream