edu.mit.csail.aeolus.api

Class AeolusFileInputStream

```
java.lang.Object
  └ edu.mit.csail.aeolus.api.AeolusFileInputStream
```

```
public class AeolusFileInputStreamextends java.lang.Object
```

An Aeolus file input stream can be used for reading data from a given file.

The stream will only be created and opened if the requester labels allow reading from the file.

If the file contents are changed after the stream has been opened, the subsequent read operation from this stream will fail.

If the caller labels change and no longer allow the read, the stream will be closed.

Constructor Summary

AeolusFileInputStream(java.lang.String hostname, java.lang.String filePath) Creates a filestream for reading from the file specified by filePath on hostname	

Method Summary

void	**close**() Closes the stream.
int	**read**(byte[] buffer) Reads bytes from the file into the specified buffer.
int	**read**(byte[] buffer, int offset, int count) Reads bytes from the file into the specified buffer.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

AeolusFileInputStream

```
public AeolusFileInputStream(java.lang.String hostname,
                             java.lang.String filePath)
                      throws java.io.FileNotFoundException,
                             java.io.IOException,
                             AeolusException
```

Creates a filestream for reading from the file specified by filePath on hostname

Parameters:

`hostname` - - node where the file is stored
`filePath` - - path to the file on hostname

Throws:

`InfoFlowControlException` - - if caller labels don't allow reading from the file or reading from a higher parent directory if the file is missing
`java.io.FileNotFoundException` - - if the file does not exist on the hostname
`AeolusException` - - if a FileSystemClient can not be created
`java.io.IOException` - - if an I/O error occurs while opening the file stream (e.g. when retrieving file labels)

Method Detail

close

```
public void close()
           throws java.io.IOException,
                  AeolusException
```

Closes the stream. Any subsequent attempts to access this stream will fail with an IOException.

Throws:

`java.io.IOException` - - if this stream has already been closed
`AeolusException`

---

read

```
public int read(byte[] buffer)
         throws java.io.IOException,
                AeolusException,
                InfoFlowControlException
```

Reads bytes from the file into the specified buffer. Up to buffer.length bytes will be read from the file.

Parameters:

`buffer` - - array to store the data into

Returns:

the actual number of bytes read

Throws:

`java.io.IOException` - - if an I/O error occurred while reading from the file (e.g. the stream has been closed)

`InfoFlowControlException` - - if labels don't allow the read

`AeolusException`

---

read

```
public int read(byte[] buffer,
                int offset,
                int count)
         throws java.io.IOException,
                AeolusException,
                InfoFlowControlException
```

Reads bytes from the file into the specified buffer. Up to count bytes will be stored in the buffer starting at the given offset in the buffer.

Returns:

the actual number of bytes read or -1 if end of file has been reached

Throws:

`java.io.IOException` - - if an I/O error occurred while reading from the file (e.g. the stream has been closed)

`InfoFlowControlException` - - if labels don't allow the read

`IoException` - - if the stream is closed

`AeolusException`