TopicDetailFragment Private Class

This document specifies the contents of the private classes in the Java Resume Application. Under normal circumstances, the average user would not have access to this information because the classes are private.

Inheritance Hierarchy

android.app

Fragment

Namespace: com.example.resume

Constructors

Properties

Methods

Name	Description
addElement(Context, ViewGroup, RelativeLayout.LayoutParams, String, int, int)	Adds TextView element to parent ViewGroup.
chooseFormat(int)	Selects TextView formatting based on the number of the CSV column stored in the resource array
chooseFormat(String)	Selects TextView formatting based on the name of the CSV column
formatFragment(Context, ViewGroup, ArrayList<List<String>>)	Determines which fragment to display and calls the appropriate methods to retrieve and show content
readAssets(int, String)	Reads a file from the assets directory.
readAssets(String strFileName)	Reads a file from the assets directory.
readInputCSVFile(InputStream is)	Reads a comma-separated-value file.
readInputFile(InputStream is)	Attaches the array adapter to the current view.
readRawFile(int intFrag, String)	Reads a raw resource file.
readRawFile(String)	Reads a raw resource file.
readSDCardFile(int, String)	Reads a file from the SD card.
readSDCardFile(String strFileName)	Reads a file from the SD card.

Remarks

This class contains most of the custom code for the resume application.

This application displays resume information in several different fragments. The resume is separated into six sections, or topics.

The main application activity displays a list of views from which the user can select; each view is associated with a resume topic. There is a correspondence between the position of the item in the list array and the name of the fragment.

After the application determines which fragment to use, the fragment is located in the resource XML file directory and inflated. The id of the inflated object is used in later processing. Only one topic fragment is completely contained in the XML file; the other fragments require that content be retrieved from assets, raw resources, or the SD card. The fragments still use XML files, but they contain no content. Instead, they contain ScrollView objects. The ScrollView object allows the view to scroll vertically if the fragment contains more content than can be displayed.

When the user selects a topic, the information that is displayed programmatically is positioned top to bottom, left to right across the display. The RelativeLayout object within the ScrollView object is more flexible than a LinearLayout or a TableLayout, and the effect is that the CSV columns are concatenated across the display. The size of each TextView object that contains CSV information is computed based on items like the number of characters to display, or the size of the font that is used. The TextView object is positioned on the screen by its top left corner.

For the most part, content that is stored in a CSV file is displayed using a normal font. However, some comma-separated values require special formatting, based on the name of the column. These are as follows:

- BUSINESS: Bold font
- TITLE: Italicized
- URL: Display as URL, and allow the application to hyperlink to the item when selected
- TOOLSANDTECHNIQUES: Prefaced by the text “Tools and Techniques: "

The topic fragment that is completely contained in the XML file also contains URLs and an email address. Consequently, there is some duplication of methods between this class and the ListActivity class. This is because the contents of the XML files are processed by the main activity, whereas the TextViews that build URLs programmatically are associated with the DetailFragment class.

When testing with an emulator, ensure that it has the capacity to start the activities or intents as requested.

This class is initially generated by the IDE when it creates a new Master/Detail Flow Android Application Project. It associates the subclasses or superclasses of the application to the classes provided by Eclipse, the Java language, and various product vendors.

Edition Information

Edition 1.0

Platforms

Eclipse Android version 4.2.2

Other devices that support Android minimum target version 17

See Also

TopicDetailActivity Class
TopicDetailFragment Class

TopicDetailFragment.addElement(Context, ViewGroup, RelativeLayout.LayoutParams, String, int, int) Method

Adds a single TextView element to the parent ViewGroup.

Namespace: com.example.resume

Syntax

Parameters

Context
Type: android.content.Context
The interface to global information about the environment in which the application is running. It allows access to application-specific resources and classes, in addition to up-calls for operations like launching activities or receiving intents.

ViewGroup
Type: android.view.ViewGroup
The group of views to which a new View object will be appended.

RelativeLayout.LayoutParams
Type: android.widget.RelativeLayout
The layout parameters for the object that will be displayed inside a RelativeLayout object. This includes items like margin padding, text color, and typeface.

String
Type: java.lang
The text to display in the View that will be added to the ViewGroup.

Int
Type: java.lang
A number that ensures that the new View object will have a unique id.

Int
Type: java.lang
The type of formatting that is required for this Comma-Separated-Value column. A value of zero specifies that no special formatting is requested.

Return Value

Type: int
The id of the element that was added to the parent View.

Remarks

Enumerations are not allowed at this level of the application, because they contain static fields. The CSV columns are handled as though an enumeration were present.

This method must be updated if another CSV column that requires formatting is added to the program. It must also be updated if the background colors of the application are changed, so that text will still display against the background.

See Also

FormatFragment Method
RelativeLayout Class, located at
http://developer.android.com/reference/android/widget/RelativeLayout.html
RelativeLayout.LayoutParams Class, located at
http://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams.html
ViewGroup Class, located at http://developer.android.com/reference/android/view/ViewGroup.html

TopicDetailFragment.chooseFormat(int) Method

Selects TextView formatting based on the number of the CSV column stored in the resource array.

Namespace: com.example.resume

Syntax

Parameters

Int
Type: java.lang
The index of the resource array that contains the name of the CSV column.

Return Value

Type:String
The name of the CSV column. If no special formatting is requested for this CSV column, an empty string is returned.

Remarks

At the present time, the CSV columns that can request special formatting are as follows:

- business
- title
- URL
- Experience
- ToolsAndTechniques

Lookups are case-insensitive.

See Also

chooseFormat(String)
addElement Method
FormatFragment Method

TopicDetailFragment.chooseFormat(String) Method

Selects TextView formatting based on the name of the CSV column.

Namespace: com.example.resume

Syntax

Parameters

String
Type: java.lang
The name of the CSV column that may require special formatting.

Return Value

Type: int
The index of the CSV column name that can be formatted. If the CSV column name is not in the resource array, no special formatting is required and the method returns a value of zero.

Remarks

At the present time, the CSV columns that can request special formatting are as follows:

- business
- Title
- URL
- Experience
- ToolsAndTechniques

Lookups are case-insensitive.

See Also

LinearLayout Class
RelativeLayout Class
TableLayout Class
ScrollView Class

TopicDetailFragment.formatFragment(Context, ViewGroup, ArrayList<List<String>>) Method

Programmatically locate content and format it for display on the device screen for the fragment that is associated with this resume topic.

Namespace: com.example.resume

Syntax

Parameters

Context
Type: android.content.Context
The interface to global information about the environment in which the application is running. It allows access to application-specific resources and classes, in addition to up-calls for operations like launching activities or receiving intents. The activity object inherits the Context object.

ViewGroup
Type: android.view.ViewGroup
The group of views to which a new View object will be appended.

ArrayList<List<String>>
Type: java.util.ArrayList
The Comma-Separated Value file that is located on the SD Card.

Return Value

This method does not return a value.

Remarks

The ViewGroup object contains a ScrollView that was inflated previous to the call to this method. The ScrollView contains a RelativeLayout that can contain the TextView objects that display the information. A RelativeLayout.LayoutParameters object is instantiated to control dynamic location of TextView objects on the screen.

The array that contains the CSV file is processed one line at a time, and within that line, one CSV column at a time. A TextView object is instantiated for each content element, and a RelativeLayout object that will specify its formatting parameters is also instantiated.

The column names from the CSV file are saved in a separate List<String> object, and are used to determine whether special formatting is required. It is suggested that a CSV file that contains many columns should put the ones that need multiline content at the end of the record. Landscape formatting will display the text across the screen; portrait formatting will insert the display equivalent of a carriage return and line feed after every other column.

The first TextView of the CSV row is aligned with the left side of the parent view. Space is added after every line within the CSV file.

See Also

LinearLayout Class
RelativeLayout Class
TableLayout Class
ScrollView Class

TopicDetailFragment.readAssets(int, String) Method

Reads a file from the assets directory.

Namespace: com.example.resume

Syntax

Parameters

Int
Type: java.lang
The number of the selection within the list item array that represents the fragment to display.

String
Type: java.lang.String
The name of the file that is located in the assets directory of the application. The filename includes the dot-3 suffix ".txt".

Return Value

Type: ArrayList<List<String>>
The Comma-Separated Value file that is located in the application assets directory.

Remarks

Arrays are counted 0 .. <size>, but selections are counted 1 .. #numberOfSelections. Consequently, the third selection of the list item array is associated with array position 2.

As of the time of this deployment, this method is not used. It primarily exists to retain method name overloading for set of read methods.

This method instantiates an InputStream object for use in methods that perform I/O functions for the application.

Text files in the /assets directory are copied verbatim into local storage when the program is installed.

See Also

readAssets(String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(String)
readSDCardFile (String)

TopicDetailFragment.readAssets(String) Method

Reads a file from the assets directory.

Namespace: com.example.resume

Syntax

Parameters

String
Type: java.lang.String
The name of the file that is located in the assets directory of the application. The filename includes the dot-3 suffix ".txt".

Return Value

Type: String
The Comma-Separated Value file that is located in the application assets directory.

Remarks

This method instantiates an InputStream object for use in methods that perform I/O functions for the application.

Text files in the /assets directory are copied verbatim into local storage when the program is installed.

See Also

readAssets(int, String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(int, String)
readSDCardFile (int, String)

TopicDetailFragment.readInputCSVFile(InputStream) Method

Reads a comma-separated-value file.

Namespace: com.example.resume

Syntax

Parameters

InputStream
Type: java.io.InputStream
The InputStream object that has been associated with the file to read.

Return Value

Type: ArrayList<List<String>>
The Comma-Separated Value file.

Remarks

The reading process is buffered so the input file can be separated with newline characters that the method can recognize.

See Also

readAssets(String)
readInputFile(InputStream)
readRawFile(String)
readSDCardFile (String)

TopicDetailFragment.readInputFile(InputStream) Method

Reads a CSV file from the assets directory.

Namespace: com.example.resume

Syntax

Parameters

InputStream
Type: java.io.InputStream
The InputStream object that has been associated with the file to read.

Return Value

Type: String
The Comma-Separated Value file.

Remarks

The reading process is buffered so the input file can be separated with newline characters that the method can recognize.

See Also

readAssets(String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(String)
readSDCardFile (String)

TopicDetailFragment.readRawFile(int, String) Method

Reads a file from the raw resources directory.

Namespace: com.example.resume

Syntax

Parameters

Int
Type: java.lang
The number of the selection within the list item array that represents the fragment to display. This parameter is not used except to retain method name overloading for the set of read methods.

String
Type: java.lang.String
The name of the file that is located in the assets directory of the application. The filename does not include a dot-3 suffix.

Return Value

Type: ArrayList<List<String>>
The Comma-Separated Value file that is located in the application raw resources directory.

Remarks

Files that should be treated as project resources are located in the /raw subdirectory of the /res resources directory. They are associated with resource ids when the application is synchronized by the IDE.

See Also

readAssets(String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(String)
readSDCardFile (String)

TopicDetailFragment.readRawFile (String) Method

Reads a file from the raw resources directory.

Namespace: com.example.resume

Syntax

Parameters

String
Type: java.lang.String
The name of the file that is located in the assets directory of the application. The filename does not include a dot-3 suffix.

Return Value

Type: String
The Comma-Separated Value file that is located in the application raw resources directory.

Remarks

Files that should be treated as project resources are located in the /raw subdirectory of the /res resources directory. They are associated with resource ids when the application is synchronized by the IDE.

See Also

readAssets(int, String)
readAssets(String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(int, String)
readSDCardFile (String)

TopicDetailFragment.readSDCardFile (int, String) Method

Reads a file from the SD card in the device.

Namespace: com.example.resume

Syntax

Parameters

Int
Type: java.lang
The number of the selection within the list item array that represents the fragment to display. This parameter is not used except to retain method name overloading for the set of read methods.

String
Type: java.lang.String
The name of the file that is located in the assets directory of the application. The filename does not include a dot-3 suffix.

Return Value

Type: ArrayList<List<String>>
The Comma-Separated Value file that is located in the SD card of the device.

Remarks

The SD card or other external directory of the device is located using the full path to the file and the Environment API.

See Also

readAssets(String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(String)
readSDCardFile (String)

TopicDetailFragment.readSDCardFile (String) Method

Reads a file from the SD card in the device.

Namespace: com.example.resume

Syntax

Parameters

String
Type: java.lang.String
The name of the file that is located in the assets directory of the application. The filename does not include a dot-3 suffix.

Return Value

Type: String
The Comma-Separated Value file that is located in the SD card in the device.

Remarks

The SD card or other external directory of the device is located using the full path to the file and the Environment API.

See Also

readAssets(String)
readInputCSVFile (InputStream)
readInputFile(InputStream)
readRawFile(String)
readSDCardFile (String)
readAssets(int, String)