MP3Java

Generated by Doxygen 1.12.0

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

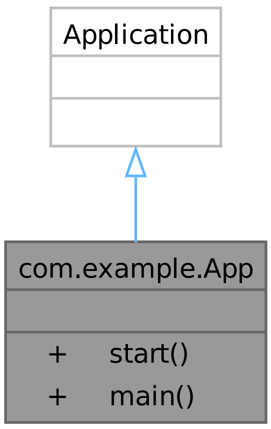
Here is a list of all documented files with brief descriptions:

Chapter 4

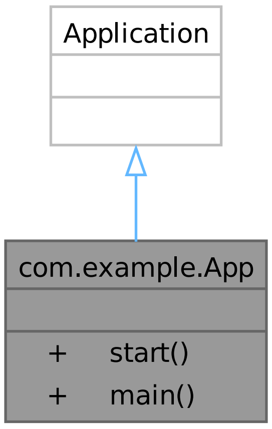
Class Documentation

4.1 com.example.App Class Reference

Inheritance diagram for com.example.App:



Collaboration diagram for com.example.App:



Public Member Functions

- void start (Stage stage) throws UnsupportedTagException, InvalidDataException, IOException

Static Public Member Functions

- static void main (String[ ] args)

4.1.1 Detailed Description

This class conatins the main method

Definition at line 13 of file App.java.

4.1.2 Member Function Documentation

4.1.2.1 main()

```
static void com.example.App.main (
             String[ ] args)  [static]
```

Definition at line 39 of file App.java.

```
                                        {
           launch(args);
    }
```

4.1.2.2 start()

```
void com.example.App.start (
             Stage stage) throws UnsupportedTagException, InvalidDataException, IOException
```

javaFx requeres this start() method to be overwritten This start() called in main.

Parameters

stage	No idea what is this and how it is works

Definition at line 21 of file App.java.

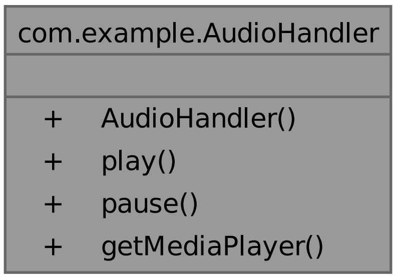
```
00021                                                                                                                                      {
00022         // INICIALIZÁLÁS
00023         // FileHandler fileHandler = new FileHandler();
00024 
00025         // // Beolvassa a megadott dir össze `.mp3` fájlját és kollekcióba rakja őket.
00026         // fileHandler.readDir("/home/i3hunor/Suli/Prog3/nagyHF/Fasz/mp3java/src/main/resources");
00027         // // A kiválasztott lejátszandó track
00028         // Track track = fileHandler.trackList.get(2);
00029 
00030         // AudioHandler audioHandler = new AudioHandler(track);
00031         // Itt hozom létre a tui handlert !!
00032         GuiHandler tuiHandler = new GuiHandler();
00033 
00034         // TUI handler kezeli az inputut és onnan hívja meg a megfelelő play/pause metódusokat
00035         // Creates a new thread that wll run the tuiHandler
00036         new Thread(tuiHandler::init).start();
00037     }
```

The documentation for this class was generated from the following file:

- src/main/java/com/example/App.java

4.2 com.example.AudioHandler Class Reference

Collaboration diagram for com.example.AudioHandler:



Public Member Functions

- AudioHandler (Track track)
- void play ()
- void pause ()
- MediaPlayer getMediaPlayer ()

4.2.1 Detailed Description

Class that handle every audio related method

Definition at line 11 of file AudioHandler.java.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 AudioHandler()

```
com.example.AudioHandler.AudioHandler (
            Track track)
```

Contructor that initializes the mediaPlayer with a track

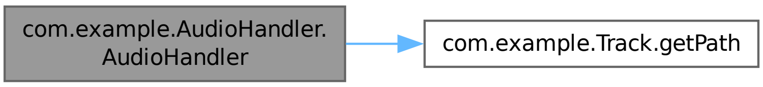
Parameters

track	dal amit le akarok játszani

Definition at line 18 of file AudioHandler.java.

```
                                     {
        String path = track.getPath();
        Media media = new Media(new File(path).toURI().toString());
        mediaPlayer = new MediaPlayer(media);
    }
```

Here is the call graph for this function:



4.2.3 Member Function Documentation

4.2.3.1 getMediaPlayer()

```
MediaPlayer com.example.AudioHandler.getMediaPlayer ()
```

Definition at line 41 of file AudioHandler.java.

```
                                        {
        return mediaPlayer;
    }
```

4.2.3.2 pause()

```
void com.example.AudioHandler.pause ()
```

Calls pause() methond on the mediaPlayer

Definition at line 36 of file AudioHandler.java.

```
00036                           {
00037         mediaPlayer.pause();
00038         System.out.println("Paused.");
00039     }
```

4.2.3.3 play()

```
void com.example.AudioHandler.play ()
```

Calls the play() method on the initialized media player Works as a resume() too

Definition at line 28 of file AudioHandler.java.

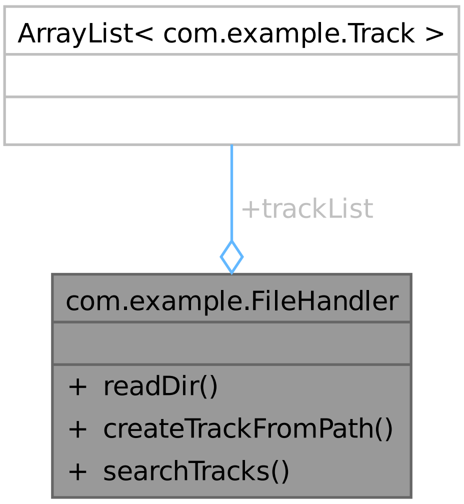
```
00028                          {
00029         mediaPlayer.play();
00030         System.out.println("Playing...");
00031     }
```

The documentation for this class was generated from the following file:

- src/main/java/com/example/AudioHandler.java

4.3 com.example.FileHandler Class Reference

Collaboration diagram for com.example.FileHandler:



Public Member Functions

- void readDir (String path) throws IOException, UnsupportedTagException, InvalidDataException
- Track createTrackFromPath (String path) throws IOException, UnsupportedTagException, InvalidData←Exception
- ArrayList< Track > searchTracks (String pattern, String path) throws IOException, UnsupportedTag←Exception, InvalidDataException

Public Attributes

- ArrayList< Track > trackList = new ArrayList<>()

4.3.1 Detailed Description

FileHandler - Minden aminek fileokhoz van köze itt van kezelve

Definition at line 19 of file FileHandler.java.

4.3.2 Member Function Documentation

4.3.2.1 createTrackFromPath()

```
Track com.example.FileHandler.createTrackFromPath (
            String path) throws IOException, UnsupportedTagException, InvalidDataException
```

Megadott útvonalból létrehozza a Track objektumot amit utána el tudok tárolni egy kollekcióban. Csak azért kell, hogy a readDir()-ben konstruálni tudjak metadatából Tracket amit pedig kollekcióba tudok elhelyezni

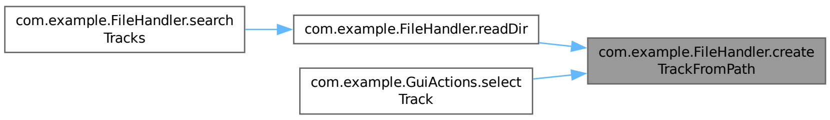
Parameters

path	- a dal elérési útvonala

Definition at line 57 of file FileHandler.java.

```
                             {
        // Inicializálás
        Mp3File mp3File = new Mp3File(path);
        long length = 0;
        String artist = null;
        String title = null;
        String album = null;

        // Megnézem, hogy milyen típusú tagjei vannak
        if(mp3File.hasId3v1Tag()){
            ID3v1 v1Tag = mp3File.getId3v1Tag();
            length = mp3File.getLengthInSeconds();
            artist = v1Tag.getArtist();
            title = v1Tag.getTitle();
            album = v1Tag.getAlbum();
        }
        else if(mp3File.hasId3v2Tag()){
            ID3v2 v2Tag = mp3File.getId3v2Tag();
            length = mp3File.getLengthInSeconds();
            artist = v2Tag.getArtist();
            title = v2Tag.getTitle();
            album = v2Tag.getAlbum();
        }
        else System.err.println("Se v1 se v2 tagjei nincsenek a fájlnak.");

        // Trakc konstruálás, lényegi rész
        return new Track(path, title, artist, album, length);
    } // end of createTrackFromPath()
```

Here is the caller graph for this function:


com.example.FileHandler.search Tracks
com.example.FileHandler.readDir
com.example.GuiActions.select Track
com.example.FileHandler.create TrackFromPath

4.3.2.2 readDir()

```
void com.example.FileHandler.readDir (
            String path) throws IOException, UnsupportedTagException, InvalidDataException
```

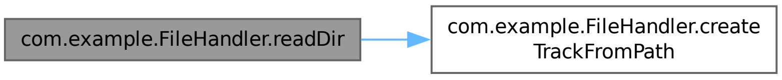
Parameters

path	dal elérési útvonala A megadott dir végigpásztázása mp3 fájlok után kutatva A megtalált mp3 fájlokból egyesével lértehoz egy T //rack objektumot

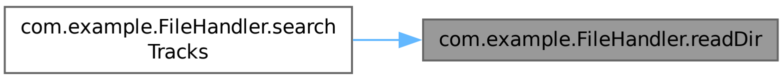
Definition at line 29 of file FileHandler.java.

```
                          {
        File dir = new File(path);
        File[] listOfFiles = dir.listFiles();
        String name;

        if(listOfFiles != null){
            for(int i = 0; i < listOfFiles.length; i++){
                name = listOfFiles[i].getName();

                if(listOfFiles[i].isFile() && name.endsWith(".mp3")){
                    Track track = createTrackFromPath(listOfFiles[i].getAbsolutePath());
                    trackList.add(track);
                }
            } // end of for loop
        }
        // Error handling
        else System.out.println("Hiba a mappa pásztázásakor");
    } // end of readDir()
```

Here is the call graph for this function:


com.example.FileHandler.readDir
com.example.FileHandler.create TrackFromPath

Here is the caller graph for this function:


com.example.FileHandler.search Tracks
com.example.FileHandler.readDir

4.3.2.3 searchTracks()

```
ArrayList< Track > com.example.FileHandler.searchTracks (
            String pattern,
            String path) throws IOException, UnsupportedTagException, InvalidDataException
```

Parameters

pattern	keresett cím stringben megadva

Returns

megtalált dal/dalok kollekciója TODO toLoweCase() Nem teljesen értem mi a retek ez a kód, de lambda Search method

Parameters

pattern	amit keresek, guiban megadtam
path	az absolute path-ja a vizsgált dir-nek.

Returns

matching trackek

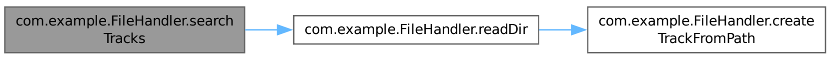
Exceptions

IOException	
UnsupportedTagException	
InvalidDataException	

Definition at line 111 of file FileHandler.java.

```
    {
        trackList.clear();
        readDir(path);

        String lowerCasepattern = pattern.toLowerCase();

        ArrayList<Track> matchingTracks = new ArrayList<>();

        for (Track t : trackList) {
            if (t.getTitle().toLowerCase().contains(lowerCasepattern) ||
                t.getArtist().toLowerCase().contains(lowerCasepattern)) {
                matchingTracks.add(t);
            }
        }

        return matchingTracks;
    }
```

Here is the call graph for this function:


com.example.FileHandler.search Tracks
com.example.FileHandler.readDir
com.example.FileHandler.create TrackFromPath

4.3.3 Member Data Documentation

4.3.3.1 trackList

```
ArrayList<Track> com.example.FileHandler.trackList = new ArrayList<>()
```

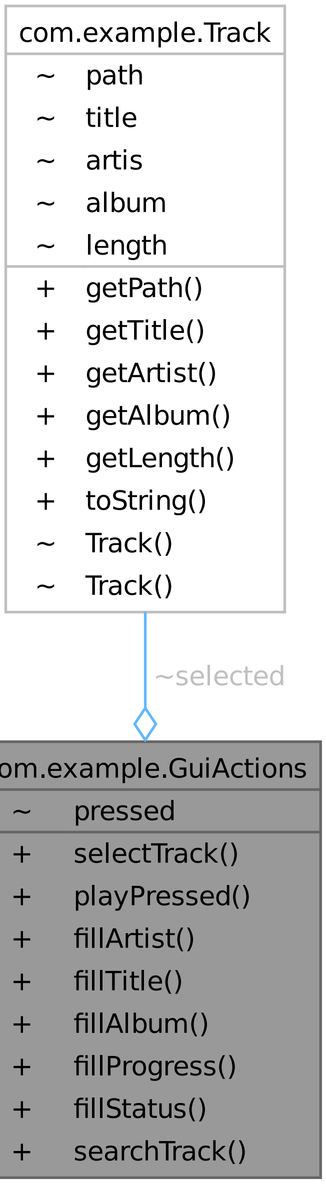
Definition at line 22 of file FileHandler.java.

The documentation for this class was generated from the following file:

- src/main/java/com/example/FileHandler.java

4.4 com.example.GuiActions Class Reference

Collaboration diagram for com.example.GuiActions:



Public Member Functions

- Track selectTrack (JFrame frame) throws UnsupportedTagException, InvalidDataException, IOException
- void playPressed ()
- void fillArtist (JTextField field)
- void fillTitle (JTextField field)
- void fillAlbum (JTextField field)

- void fillProgress (JTextField field)
- String fillStatus ()
- ArrayList< Track > searchTrack (String pattern) throws UnsupportedTagException, InvalidDataException, IOException

Package Attributes

- boolean pressed = false
- Track selected

4.4.1 Detailed Description

Minden guiHandlerben helyet foglaló gombnak és fieldnek itt vannak definiálva a listenerjei és azok metódusai

Definition at line 19 of file GuiActions.java.

4.4.2 Member Function Documentation

4.4.2.1 fillAlbum()

```
void com.example.GuiActions.fillAlbum (
            JTextField field)
```

Definition at line 86 of file GuiActions.java.

```
                                           {
        if(selected != null){
            String album = selected.getAlbum();
            field.setText(album);
            System.out.println("Album filed filled");
        }
        else field.setText("NA");
    }
```

4.4.2.2 fillArtist()

```
void com.example.GuiActions.fillArtist (
            JTextField field)
```

Definition at line 68 of file GuiActions.java.

```
                                            {
        if(selected != null){
            String artist = selected.getArtist();
            field.setText(artist);
            System.out.println("Artist filed filled");
        }
        else field.setText("NA");
    }
```

4.4.2.3 fillProgress()

```
void com.example.GuiActions.fillProgress (
            JTextField field)
```

Definition at line 95 of file GuiActions.java.

```
                                              {
        if(selected != null){
           long len = selected.getLength();
           int curr = (int) audioHandler.getMediaPlayer().getCurrentTime().toSeconds();
           field.setText(curr + "/" + len);
           // System.out.println("Length of the track: " + len);
        }
        else field.setText("NA");
     }
```

4.4.2.4 fillStatus()

```
String com.example.GuiActions.fillStatus ()
```

Definition at line 105 of file GuiActions.java.

```
00105                                    {
00106         if(selected != null){
00107             String status = audioHandler.getMediaPlayer().getStatus().toString();
00108             System.out.println("Status filled");
00109             return status;
00110         }
00111         return "Not selected";
00112     }
```

4.4.2.5 fillTitle()

```
void com.example.GuiActions.fillTitle (
            JTextField field)
```

Definition at line 77 of file GuiActions.java.

```
00077                                               {
00078         if(selected != null){
00079             String artist = selected.getTitle();
00080             field.setText(artist);
00081             System.out.println("Titke filed filled");
00082         }
00083         else field.setText("NA");
00084     }
```

4.4.2.6 playPressed()

```
void com.example.GuiActions.playPressed ()
```

Definition at line 57 of file GuiActions.java.

```
00057                                   {
00058         if (!pressed) {
00059             audioHandler.play();
00060             pressed = true;
00061         }
00062         else {
00063             audioHandler.pause();
00064             pressed = false;
00065         }
00066     }
```

4.4.2.7 searchTrack()

```
ArrayList< Track > com.example.GuiActions.searchTrack (
            String pattern) throws UnsupportedTagException, InvalidDataException, IOException
```

Definition at line 114 of file GuiActions.java.

```
00114
     {
00115         return fileHandler.searchTracks(pattern,
     "/home/i3hunor/Suli/Prog3/nagyHF/Fasz/mp3java/src/main/resources");
00116     }
```

4.4.2.8 selectTrack()

```
Track com.example.GuiActions.selectTrack (
            JFrame frame) throws UnsupportedTagException, InvalidDataException, IOException
```

A dal/file kiválasztásáért felel

Parameters

frame	

Returns

selected track

Exceptions

UnsupportedTagException	
InvalidDataException	
IOException	

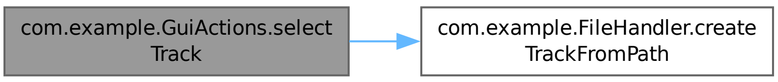
curr directoryt nyitja meg

csak mp3 fájlokat látsz

Definition at line 33 of file GuiActions.java.

```
    {
        JFileChooser fileChooser = new JFileChooser();

        String currentDirectory = System.getProperty("user.dir");
        fileChooser.setCurrentDirectory(new File(currentDirectory));

        fileChooser.setFileFilter(new javax.swing.filechooser.FileNameExtensionFilter("MP3 Files",
"mp3"));

        int result = fileChooser.showOpenDialog(frame);

        if (result == JFileChooser.APPROVE_OPTION) {
            File selectedFile = fileChooser.getSelectedFile();

            selected = fileHandler.createTrackFromPath(selectedFile.getAbsolutePath());

            audioHandler = new AudioHandler(selected);

            return selected;
        }

        return null;
    }
```

Here is the call graph for this function:


com.example.GuiActions.select Track
com.example.FileHandler.create TrackFromPath

4.4.3 Member Data Documentation

4.4.3.1 pressed

```
boolean com.example.GuiActions.pressed = false  [package]
```

Definition at line 22 of file GuiActions.java.

4.4.3.2 selected

```
Track com.example.GuiActions.selected [package]
```

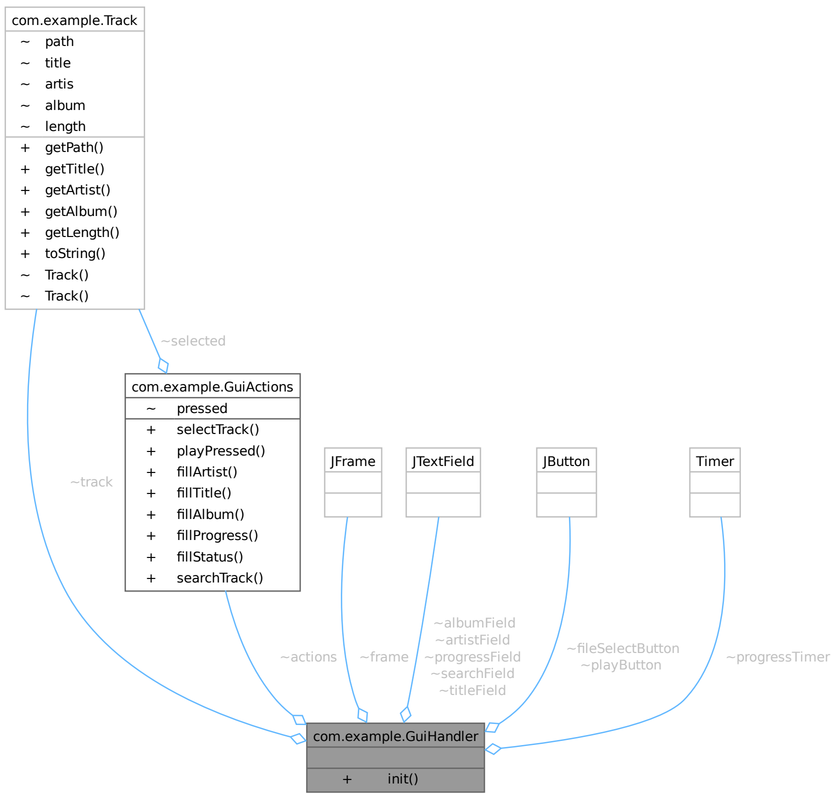
Definition at line 23 of file GuiActions.java.

The documentation for this class was generated from the following file:

- src/main/java/com/example/GuiActions.java

4.5 com.example.GuiHandler Class Reference

Collaboration diagram for com.example.GuiHandler:



Public Member Functions

- void init ()

Package Attributes

- GuiActions actions = new GuiActions()
- Track track
- JFrame frame
- JTextField searchField
- JTextField artistField
- JTextField titleField
- JTextField albumField
- JTextField progressField
- JButton playButton
- JButton fileSelectButton
- Timer progressTimer

4.5.1 Detailed Description

Definition at line 17 of file GuiHandler.java.

4.5.2 Member Function Documentation

4.5.2.1 init()

```
void com.example.GuiHandler.init ()
```

Definition at line 25 of file GuiHandler.java.

```
                        {
        initFrame();
        initComponents();
        configureProgressTimer();
    }
```

4.5.3 Member Data Documentation

4.5.3.1 actions

```
GuiActions com.example.GuiHandler.actions = new GuiActions() [package]
```

Definition at line 18 of file GuiHandler.java.

4.5.3.2 albumField

```
JTextField com.example.GuiHandler.albumField [package]
```

Definition at line 21 of file GuiHandler.java.

4.5.3.3 artistField

```
JTextField com.example.GuiHandler.artistField [package]
```

Definition at line 21 of file GuiHandler.java.

4.5.3.4 fileSelectButton

```
JButton com.example.GuiHandler.fileSelectButton  [package]
```

Definition at line 22 of file GuiHandler.java.

4.5.3.5 frame

```
JFrame com.example.GuiHandler.frame  [package]
```

Definition at line 20 of file GuiHandler.java.

4.5.3.6 playButton

```
JButton com.example.GuiHandler.playButton  [package]
```

Definition at line 22 of file GuiHandler.java.

4.5.3.7 progressField

```
JTextField com.example.GuiHandler.progressField  [package]
```

Definition at line 21 of file GuiHandler.java.

4.5.3.8 progressTimer

```
Timer com.example.GuiHandler.progressTimer  [package]
```

Definition at line 23 of file GuiHandler.java.

4.5.3.9 searchField

```
JTextField com.example.GuiHandler.searchField  [package]
```

Definition at line 21 of file GuiHandler.java.

4.5.3.10 titleField

```
JTextField com.example.GuiHandler.titleField  [package]
```

Definition at line 21 of file GuiHandler.java.

4.5.3.11 track

```
Track com.example.GuiHandler.track  [package]
```

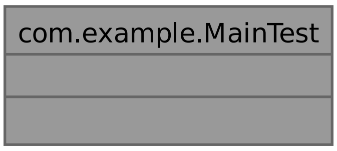
Definition at line 19 of file GuiHandler.java.

The documentation for this class was generated from the following file:

- src/main/java/com/example/GuiHandler.java

4.6 com.example.MainTest Class Reference

Collaboration diagram for com.example.MainTest:



4.6.1 Detailed Description

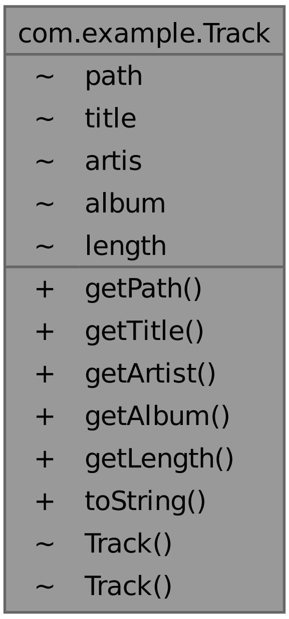
Definition at line 7 of file MainTest.java.

The documentation for this class was generated from the following file:

- src/test/java/com/example/MainTest.java

4.7 com.example.Track Class Reference

Collaboration diagram for com.example.Track:



Public Member Functions

- String getPath ()
- String getTitle ()
- String getArtist ()
- String getAlbum ()
- long getLength ()
- String toString ()

Package Functions

- Track (String path, String title, String artist, String album, long length)

  Sec-ben megadott dal hossz.
- Track ()

  Paraméter nélküli konstruktor.

Package Attributes

- String path
- String title

  Absolute path that is set by the FileHandler.readDir()
- String artis
- String album
- long length

4.7.1 Detailed Description

Definition at line 3 of file Track.java.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Track() [1/2]

```
com.example.Track.Track (
              String path,
              String title,
              String artist,
              String album,
              long length)  [package]
```

Sec-ben megadott dal hossz.

Konstruktor

Definition at line 11 of file Track.java.

```
00011                                                                               {
00012         this.path = path;
00013         this.title = title;
00014         this.artis = artist;
00015         this.album = album;
00016         this.length = length;
00017     }
```

4.7.2.2 Track() [2/2]

```
com.example.Track.Track ()  [package]
```

Paraméter nélküli konstruktor.

Definition at line 20 of file Track.java.

```
00020            {
00021         this.path = null;
00022         this.title = null;
00023         this.artis = null;
00024         this.album = null;
00025         this.length = 0;
00026     }
```

4.7.3 Member Function Documentation

4.7.3.1 getAlbum()

```
String com.example.Track.getAlbum ()
```

Definition at line 46 of file Track.java.

```
00046                             {
00047         return album;
00048     }
```

4.7.3.2 getArtist()

```
String com.example.Track.getArtist ()
```

Definition at line 42 of file Track.java.

```
00042                           {
00043         return artis;
00044     }
```

4.7.3.3 getLength()

```
long com.example.Track.getLength ()
```

Definition at line 50 of file Track.java.

```
00050                         {
00051         return length;
00052     }
```

4.7.3.4 getPath()

```
String com.example.Track.getPath ()
```

The fileHandler class Readdi method sets this attribute

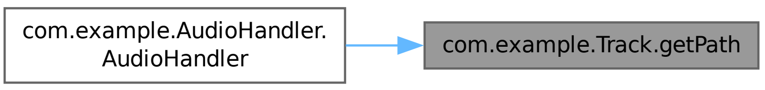
Returns

path absolute path to the song

Definition at line 34 of file Track.java.

```
00034                         {
00035         return path;
00036     }
```

Here is the caller graph for this function:



4.7.3.5 getTitle()

```
String com.example.Track.getTitle ()
```

Definition at line 38 of file Track.java.

```
00038                          {
00039         return title;
00040     }
```

4.7.3.6 toString()

```
String com.example.Track.toString ()
```

Definition at line 55 of file Track.java.

```
00055                                             {
00056             return "title=" + title + "’, artist=’" + artis + " ";
00057         }
```

4.7.4 Member Data Documentation

4.7.4.1 album

```
String com.example.Track.album  [package]
```

Definition at line 7 of file Track.java.

4.7.4.2 artis

```
String com.example.Track.artis  [package]
```

Definition at line 6 of file Track.java.

4.7.4.3 length

```
long com.example.Track.length  [package]
```

Definition at line 8 of file Track.java.

4.7.4.4 path

```
String com.example.Track.path  [package]
```

Definition at line 4 of file Track.java.

4.7.4.5 title

```
String com.example.Track.title  [package]
```

Absolute path that is set by the FileHandler.readDir()

Definition at line 5 of file Track.java.

The documentation for this class was generated from the following file:

- src/main/java/com/example/Track.java

Chapter 5

File Documentation

5.1 App.java

```
package com.example;

import java.io.IOException;

import com.mpatric.mp3agic.InvalidDataException;
import com.mpatric.mp3agic.UnsupportedTagException;

import javafx.application.Application;
import javafx.stage.Stage;
public class App extends Application {

    @Override
    public void start(Stage stage) throws UnsupportedTagException, InvalidDataException, IOException {
        // INICIALIZÁLÁS
        // FileHandler fileHandler = new FileHandler();

        // // Beolvassa a megadott dir össze `.mp3` fájlját és kollekcióba rakja őket.
        // fileHandler.readDir("/home/i3hunor/Suli/Prog3/nagyHF/Fasz/mp3java/src/main/resources");
        // // A kiválasztott lejátszandó track
        // Track track = fileHandler.trackList.get(2);

        // AudioHandler audioHandler = new AudioHandler(track);
        // Itt hozom létre a tui handlert !!
        GuiHandler tuiHandler = new GuiHandler();

        // TUI handler kezeli az inputut és onnan hívja meg a megfelelő play/pause metódusokat
        // Creates a new thread that wll run the tuiHandler
        new Thread(tuiHandler::init).start();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

5.2 AudioHandler.java

```
package com.example;

import java.io.File;

import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;

public class AudioHandler {
    private MediaPlayer mediaPlayer;

    public AudioHandler(Track track) {
        String path = track.getPath();
        Media media = new Media(new File(path).toURI().toString());
        mediaPlayer = new MediaPlayer(media);
    }

    public void play() {
        mediaPlayer.play();
        System.out.println("Playing...");
    }

    public void pause() {
        mediaPlayer.pause();
        System.out.println("Paused.");
    }

    public MediaPlayer getMediaPlayer(){
        return mediaPlayer;
    }

} // END OF AUDIOHANDLER
```

5.3 FileHandler.java

```
package com.example;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.stream.Collectors;

import javax.swing.plaf.basic.BasicSliderUI.TrackListener;

import com.mpatric.mp3agic.ID3v1;
import com.mpatric.mp3agic.ID3v2;
import com.mpatric.mp3agic.InvalidDataException;
import com.mpatric.mp3agic.Mp3File;
import com.mpatric.mp3agic.UnsupportedTagException;

public class FileHandler {

    // Ebben a kollekcióban  tárolom a dalokat
    public ArrayList<Track> trackList = new ArrayList<>();

    public void readDir(String path) throws IOException, UnsupportedTagException,
     InvalidDataException{
        File dir = new File(path);
        File[] listOfFiles = dir.listFiles();
        String name;

        if(listOfFiles != null){
            for(int i = 0; i < listOfFiles.length; i++){
                name = listOfFiles[i].getName();

                if(listOfFiles[i].isFile() && name.endsWith(".mp3")){
                    Track track = createTrackFromPath(listOfFiles[i].getAbsolutePath());
                    trackList.add(track);
                }
            } // end of for loop
        }
        // Error handling
        else System.out.println("Hiba a mappa pásztázásakor");
    } // end of readDir()


     public Track createTrackFromPath(String path) throws IOException, UnsupportedTagException,
     InvalidDataException {
        // Inicializálás
        Mp3File mp3File = new Mp3File(path);
        long length = 0;
        String artist = null;
        String title = null;
        String album = null;

        // Megnézem, hogy milyen típusú tagjei vannak
        if(mp3File.hasId3v1Tag()){
            ID3v1 v1Tag = mp3File.getId3v1Tag();
            length = mp3File.getLengthInSeconds();
            artist = v1Tag.getArtist();
            title = v1Tag.getTitle();
            album = v1Tag.getAlbum();
        }
        else if(mp3File.hasId3v2Tag()){
            ID3v2 v2Tag = mp3File.getId3v2Tag();
            length = mp3File.getLengthInSeconds();
            artist = v2Tag.getArtist();
            title = v2Tag.getTitle();
            album = v2Tag.getAlbum();
        }
        else System.err.println("Se v1 se v2 tagjei nincsenek a fájlnak.");

        // Trakc konstruálás, lényegi rész
        return new Track(path, title, artist, album, length);
    } // end of createTrackFromPath()

    // /**
    //  * @param pattern keresett cím stringben megadva
    //  * @return megtalált dal/dalok kollekciója
    //  * TODO toLoweCase()
    //  * Nem teljesen értem mi a retek ez a kód, de lambda
    //  */
    // public ArrayList<Track> search(String pattern) {
    //     return trackList.stream()
    //             .filter(track ->
    //                 track.getTitle().contains(pattern)
    //                 || track.getArtist().contains(pattern))
    //             .collect(Collectors.toCollection(ArrayList::new));
    // } // end of search()


    public ArrayList<Track> searchTracks(String pattern, String path) throws IOException,
UnsupportedTagException, InvalidDataException {
        trackList.clear();
        readDir(path);

        String lowerCasepattern = pattern.toLowerCase();

        ArrayList<Track> matchingTracks = new ArrayList<>();

        for (Track t : trackList) {
            if (t.getTitle().toLowerCase().contains(lowerCasepattern) ||
                t.getArtist().toLowerCase().contains(lowerCasepattern)) {
                matchingTracks.add(t);
            }
        }

        return matchingTracks;
    }



} // end of fileHandler class
```

5.4 GuiActions.java

```
package com.example;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;

import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JTextField;

import com.mpatric.mp3agic.InvalidDataException;
import com.mpatric.mp3agic.UnsupportedTagException;

public class GuiActions {
    private FileHandler fileHandler = new FileHandler();
    private AudioHandler audioHandler;
    boolean pressed = false;
    Track selected;

    public Track selectTrack(JFrame frame) throws UnsupportedTagException, InvalidDataException,
IOException {
        JFileChooser fileChooser = new JFileChooser();

        String currentDirectory = System.getProperty("user.dir");
        fileChooser.setCurrentDirectory(new File(currentDirectory));

        fileChooser.setFileFilter(new javax.swing.filechooser.FileNameExtensionFilter("MP3 Files",
"mp3"));

        int result = fileChooser.showOpenDialog(frame);

        if (result == JFileChooser.APPROVE_OPTION) {
            File selectedFile = fileChooser.getSelectedFile();

            selected = fileHandler.createTrackFromPath(selectedFile.getAbsolutePath());

            audioHandler = new AudioHandler(selected);

            return selected;
        }

        return null;
    }

    public void playPressed(){
        if (!pressed) {
            audioHandler.play();
            pressed = true;
        }
        else {
            audioHandler.pause();
            pressed = false;
        }
    }

    public void fillArtist(JTextField field){
        if(selected != null){
             String artist = selected.getArtist();
             field.setText(artist);
             System.out.println("Artist filed filled");
        }
        else field.setText("NA");
   }

    public void fillTitle(JTextField field){
        if(selected != null){
             String artist = selected.getTitle();
             field.setText(artist);
             System.out.println("Titke filed filled");
        }
        else field.setText("NA");
   }

    public void fillAlbum(JTextField field){
        if(selected != null){
             String album = selected.getAlbum();
             field.setText(album);
             System.out.println("Album filed filled");
        }
        else field.setText("NA");
   }

    public void fillProgress(JTextField field){
        if(selected != null){
            long len = selected.getLength();
            int curr = (int) audioHandler.getMediaPlayer().getCurrentTime().toSeconds();
            field.setText(curr + "/" + len);
            // System.out.println("Length of the track: " + len);
        }
        else field.setText("NA");
    }

     public String fillStatus(){
        if(selected != null){
             String status = audioHandler.getMediaPlayer().getStatus().toString();
             System.out.println("Status filled");
             return status;
        }
        return "Not selected";
    }

    public ArrayList<Track> searchTrack(String pattern) throws UnsupportedTagException, InvalidDataException, IOException {
        return fileHandler.searchTracks(pattern, "/home/i3hunor/Suli/Prog3/nagyHF/Fasz/mp3java/src/main/resources");
    }


}
```

5.5 GuiHandler.java

```
package com.example;

import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.io.IOException;
import java.util.ArrayList;

import javax.swing.JButton;
import javax.swing.JFrame;
```

```java
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.Timer;

import com.mpatric.mp3agic.InvalidDataException;
import com.mpatric.mp3agic.UnsupportedTagException;

public class GuiHandler {
    GuiActions actions = new GuiActions();
    Track track;
    JFrame frame;
    JTextField searchField, artistField, titleField, albumField, progressField;
    JButton playButton, fileSelectButton;
    Timer progressTimer;

    public void init() {
        initFrame();
        initComponents();
        configureProgressTimer();
    }

    private void initFrame() {
        frame = new JFrame("Mp3Java");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 250);
        frame.setLayout(new GridBagLayout());
        setupLayout();
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    /*
     * Initeli a layoutot
     */
    private void setupLayout() {
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.weightx = 1.0;

        // Top row: Search field
        addSearchLabelAndField(gbc, 0, "Search:", searchField = new JTextField(20));

        // Artist row
        addLabelAndField(gbc, 1, "Artist:", artistField = new JTextField(20));

        // Title row
        addLabelAndField(gbc, 2, "Title:", titleField = new JTextField(20));

        // Album row
        addLabelAndField(gbc, 3, "Album:", albumField = new JTextField(20));

        // Middle row: Progress field
        gbc.gridx = 0;
        gbc.gridy = 4;
        gbc.gridwidth = 3;
        progressField = createReadOnlyField();
        frame.add(progressField, gbc);

        // Bottom row: File Select and Play/Stop buttons
        gbc.gridwidth = 1;
        gbc.gridy = 5;
        fileSelectButton = new JButton("File Select");
        gbc.gridx = 0;
        frame.add(fileSelectButton, gbc);

        playButton = new JButton("Play");
        gbc.gridx = 1;
        frame.add(playButton, gbc);

        addButtonListeners();
    }

    private void addSearchLabelAndField(GridBagConstraints gbc, int row, String label, JTextField
field) {
        gbc.gridx = 0;
        gbc.gridy = row;
        frame.add(new JLabel(label), gbc);
        field.setEditable(true); // Make searchField editable
        gbc.gridx = 1;
        gbc.gridwidth = 2;
        frame.add(field, gbc);
        gbc.gridwidth = 1;
    }

    private void addLabelAndField(GridBagConstraints gbc, int row, String label, JTextField field) {
        gbc.gridx = 0;
        gbc.gridy = row;
        frame.add(new JLabel(label), gbc);
        field.setEditable(false);
        gbc.gridx = 1;
        gbc.gridwidth = 2;
        frame.add(field, gbc);
        gbc.gridwidth = 1;
    }

    private JTextField createReadOnlyField() {
        JTextField field = new JTextField();
        field.setEditable(false);
        return field;
    }

    private void initComponents() {
        actions.fillArtist(artistField);
        actions.fillTitle(titleField);
        actions.fillAlbum(albumField); // Fill the album field if the method is available in GuiActions
    }

    private void addButtonListeners() {
        fileSelectButton.addActionListener(e -> selectTrack());
        playButton.addActionListener(e -> togglePlay());

        // Add ActionListener to the search field
        searchField.addActionListener(e -> {
            try {
                searchTrack(searchField.getText());
            } catch (UnsupportedTagException | InvalidDataException | IOException e1) {
                e1.printStackTrace();
            }
        });
    }

    private void configureProgressTimer() {
        progressTimer = new Timer(1000, e -> actions.fillProgress(progressField));
    }

    private void selectTrack() {
        try {
            track = actions.selectTrack(frame);
            initComponents();
        } catch (UnsupportedTagException | InvalidDataException | IOException e1) {
            e1.printStackTrace();
        }
    }

    private void togglePlay() {
        actions.playPressed();
        updateDynamicFields();

        String currentStatus = actions.fillStatus();
        playButton.setText(currentStatus.equals("PLAYING") ? "Pause" : "Play");

        if (actions.pressed) {
            progressTimer.start();
        } else {
            progressTimer.stop();
        }
    }

    private void updateDynamicFields() {
        actions.fillProgress(progressField);
    }

    private void searchTrack(String pattern) throws UnsupportedTagException, InvalidDataException, IOException {
       System.out.println("Searching for: " + pattern);
       ArrayList<Track> tracks = new ArrayList<>();
       tracks = actions.searchTrack(pattern);
       System.out.println(tracks);
    }


}
```

5.6 Track.java

```java
package com.example;

public class Track {
    String path;
    String title;
    String artis;
    String album;
    long length;

    Track(String path, String title, String artist, String album, long length){
        this.path = path;
        this.title = title;
        this.artis = artist;
        this.album = album;
        this.length = length;
    }

    Track(){
        this.path = null;
        this.title = null;
        this.artis = null;
        this.album = null;
        this.length = 0;
    }

    // Getterek

    public String getPath(){
        return path;
    }

    public String getTitle(){
        return title;
    }

    public String getArtist(){
        return artis;
    }

    public String getAlbum(){
        return album;
    }

    public long getLength(){
        return length;
    }

    @Override
    public String toString() {
        return "title=" + title + "', artist='" + artis + " ";
    }



} // end of Track class


```

5.7 module-info.java

```java
module com.example {
    requires javafx.controls;
    requires javafx.fxml;

    opens com.example to javafx.fxml;
    exports com.example;

    // chatgpt
    requires javafx.media;
    requires mp3agic;

    // Swing + awt
    requires java.desktop;
}
```

5.8 MainTest.java

```java
package com.example;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;
```

```

public class MainTest {

    // @Test
    // public void testAdd() {
    //     Main main = new Main();
    //     int result = main.add(1, 2);
    //     assertEquals(3, result, "The add method should add two numbers");
    // }
}

```

Index