[Работа с файлами на Android](http://ru.stackoverflow.com/questions/40851/%d0%a0%d0%b0%d0%b1%d0%be%d1%82%d0%b0-%d1%81-%d1%84%d0%b0%d0%b9%d0%bb%d0%b0%d0%bc%d0%b8-%d0%bd%d0%b0-android)

Как осуществить проверку существует ли файл или нет? Если сущ, то произвести его чтение, иначе же создать, и произвести запись...

Пытался создать файл таким образом:

File myF = new File(Environment.getDataDirectory(),"file.txt");

try

{

myF.createNewFile();

text.setText(myF.getAbsolutePath());

}

catch (IOException e1)

{

text.setText(e1.toString());

}

В итоге пишет java.io.IOException: Permission denied

Да и в любом случае при применение метода createNewFile() вылезает это...

После долгих блужданий по инету я нашел решение своей проблемы...

изначально в manifest-файл приложения нужно было добавить следующую строку

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"></uses-permission>

Далее все происходит так:

File myF = new File(Environment.getExternalStorageDirectory()+"/data/","file.txt");

if(!myF.exists())

{

try

{

String str = "0";

myF.createNewFile();

try

{

FileOutputStream fOut = openFileOutput(myF.getAbsolutePath(),0);

OutputStreamWriter outStr = new OutputStreamWriter(fOut);

outStr.write(str);

outStr.flush();

outStr.close();

}

catch (Throwable t)

{

Toast.makeText(getApplicationContext(), "Exception: " + t.toString(), Toast.LENGTH\_LONG).show();

}

}

catch (IOException e1)

{

text.setText(e1.toString());

}

}

else

{

text.setText("Файл есть!");

}

Но теперь вылезает:

java.lang.IllegalArgumentException: File mnt//sdcard//data//file.txt contains a path separator

Полазив нашел что "openFileOutput не моет принимать путь, а только имя файла.."

Но как тогда обратится к созданному файлу? Подскажите..

Спасибо.

**ОТВЕТ**

File file = new File(path);

if (!file.exists()) {

file.createNewFile()

// и писать

FileOutputStream fOut = openFileOutput(file.getAbsolutePath(), MODE\_WORLD\_READABLE);

OutputStreamWriter osw = new OutputStreamWriter(fOut);

osw.write("сюда писать инфу");

osw.flush();

osw.close();

} else {

// читать

FileInputStream fIn = openFileInput(file.getAbsolutePath());

InputStreamReader isr = new InputStreamReader(fIn);

char[] inputBuffer = new char[bufSize];

isr.read(inputBuffer);

String read = new String(inputBuffer);

}

Флаг MODE\_WORLD\_READABLE позволяет другим приложениям получить доступ к файлу с правами чтения. Есть еще MODE\_PRIVATE - только создатель получает доступ, и MODE\_WORLD\_WRITEABLE - все приложения могут писать в этот файл. Флаги определены в классе Context.

Второй ОТВЕТ

В итоге нашелся такой выход:

File myF = new File(Environment.getExternalStorageDirectory()+"/data/","file.txt");

if(!myF.exists())

{

try

{

String str = "500";

myF.createNewFile();

try

{

FileWriter fWr = new FileWriter(myF);

fWr.write(str);

fWr.flush();

fWr.close();

}

catch (Throwable t)

{

Toast.makeText(getApplicationContext(), "Exception: " + t.toString(), Toast.LENGTH\_LONG).show();

}

text.setText("Бюджет - "+str+" р.");

}

catch (Exception e1)

{

text.setText(e1.toString());

}

}

else

{

try {

FileReader fRd = new FileReader(myF);

BufferedReader reader = new BufferedReader(fRd);

String str;

StringBuffer buffer = new StringBuffer();

while ((str = reader.readLine()) != null)

{

buffer.append(str + "\n");

}

fRd.close();

text.setText("Бюджет - "+buffer.toString()+"р.");

}

catch (Throwable t)

{

Toast.makeText(getApplicationContext(),"Exception: " + t.toString(),Toast.LENGTH\_LONG).show();

}

}