

 免费电子书

学习

android-asyncTask

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
asyncTask

.....	1
1: android-asyncTask	2
.....	2
Examples	2
AsyncTask	2
2: AsyncTask	6
.....	6
Examples	6
AsyncTask	6
.....	9

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [android-asyncTask](#)

It is an unofficial and free android-asyncTask ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-asyncTask.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: android-asyncTask

android-asyncTask。

android-asyncTask。 android-asyncTask。

Examples

AsyncTask

AsyncTaskUI。 /。 AsyncTask。 AsyncTask。 。

AsyncTask。 4。 **onPreExecute** UI。 **doInBackground** UI。 **onProgressUpdate** UI。
onPostExecute。

AsyncTask。

1

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button = (FloatingActionButton) findViewById(R.id.btn);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                executeAsyncTaskOperation();
            }
        });
    }

    private void executeAsyncTaskOperation() {
        new CustomAsyncTask(this).execute();
    }

    private static class CustomAsyncTask extends AsyncTask<Void, Void, String> {

        private Context context;
        private ProgressDialog progressDialog;

        public CustomAsyncTask(Context context) {
            this.context = context;
        }

        @Override
```



```

private void executeAsyncTaskOperation() {
    Boolean[] bools = new Boolean[10];
    for (int k = 0; k < 10; k++) {
        if (k % 2 == 0) {
            bools[k] = true;
        } else {
            bools[k] = false;
        }
    }
    new CustomAsyncTask(this).execute(bools);
}

private static class CustomAsyncTask extends AsyncTask<Boolean, Void, Integer> {

    private Context context;
    private ProgressDialog progressDialog;

    public CustomAsyncTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        progressDialog = ProgressDialog.show(context, "Please wait...", "Loading data from
web");
    }

    @Override
    protected Integer doInBackground(Boolean... params) {
        int count = 0;
        try {
            Thread.sleep(1000);
            Log.d(CustomAsyncTask.class.getCanonicalName(), "doInBackground");
            for (Boolean param : params) {
                if (param) {
                    count++;
                }
            }
        } catch (Exception exc) {
            Log.e(CustomAsyncTask.class.getCanonicalName(), "exception");
            count = 0;
        }
        return count;
    }

    @Override
    protected void onPostExecute(Integer s) {
        if (progressDialog != null && progressDialog.isShowing()) {
            progressDialog.dismiss();
        }
        if (s != null && s > 0) {
            Toast.makeText(context, "finished loading: " + s + " tasks",
Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(context, "finished unsuccessfully!", Toast.LENGTH_LONG).show();
        }
    }
}
}

```


2: AsyncTask

AsyncTask

Examples

AsyncTask

◦ ◦

```
public class MainActivity extends AppCompatActivity {

    private static AtomicBoolean inWork;
    private CustomAsyncTask asyncTask;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        inWork = new AtomicBoolean(false);
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();

                executeAsyncTaskOperation();
            }
        });
    }

    private void executeAsyncTaskOperation() {
        Boolean[] bools = new Boolean[10];
        for (int k = 0; k < 10; k++) {
            if (k % 2 == 0) {
                bools[k] = true;
            } else {
                bools[k] = false;
            }
        }
        asyncTask = new CustomAsyncTask(this);
        asyncTask.execute(bools);
    }

    //pressing the home button while the task is running will trigger the onStop being called.
    @Override
    protected void onStop() {
        if (asyncTask.getStatus() == AsyncTask.Status.RUNNING) {
            asyncTask.cancel(true);
        }
        super.onStop();
    }
}
```



```

private static class CustomAsyncTask extends AsyncTask<Boolean, Void, Integer> {

    private Context context;
    private ProgressDialog progressDialog;

    public CustomAsyncTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onCancelled() {
        inWork.set(false);
        if (progressDialog != null && progressDialog.isShowing()) {
            progressDialog.dismiss();
            Log.d(CustomAsyncTask.class.getCanonicalName(), "progressdialog is
dismissed.");
        }
    }

    @Override
    protected void onPreExecute() {
        progressDialog = ProgressDialog.show(context, "Please wait...", "Loading data from
web");
    }

    @Override
    protected Integer doInBackground(Boolean... params) {
        int count = 0;
        inWork.set(true);
        try {
            Thread.sleep(1000);
            Log.d(CustomAsyncTask.class.getCanonicalName(), "doInBackground");
            if (!isCancelled()) {
                for (Boolean param : params) {
                    if (param) {

                        count++;
                    }
                }
            } else {
                Log.d(CustomAsyncTask.class.getCanonicalName(), "doInBackground is
cancelled.");
            }

        } catch (Exception exc) {
            Log.e(CustomAsyncTask.class.getCanonicalName(), "exception");
            count = 0;
        }
        return count;
    }

    @Override
    protected void onPostExecute(Integer s) {
        if (!isCancelled()) {
            inWork.set(false);
            if (progressDialog != null && progressDialog.isShowing()) {
                progressDialog.dismiss();
            }
            if (s != null && s > 0) {
                Toast.makeText(context, "finished loading: " + s + " tasks",
Toast.LENGTH_LONG).show();
            }
        }
    }
}

```

```
        } else {
            Toast.makeText(context, "finished unsuccessfully!",
Toast.LENGTH_LONG).show();

        }
    } else {
        Log.d(CustomAsyncTask.class.getCanonicalName(), "onPostExecute is
cancelled.");
    }
}
}
}
```

AsyncTask <https://riptutorial.com/zh-CN/android-async-task/topic/8783/async-task>

S. No		Contributors
1	android-asyncTask	Andrei T , Community , rossettistone
2	AsyncTask	Andrei T