Androidで

もっとサービスしよう！

日本Androidの会

3月28日 大阪セミナー

有山圭二 (有限会社シーリス)

目次

- 前回の要旨
- ハンズオン
  - プロジェクトの準備 (ダウンロード)
  - AIDLによるインターフェースの定義と実装
  - Service側の実装
    - AIDLで定義されたインターフェースの実装
    - バインドされた際の動作を実装
  - Activity側の実装
    - Serviceにバインドするコードを追加
    - ボタンイベントの処理を実装
- まとめ
- 参考

前回の要旨

- Serviceは、それ自身は画面を持たず、バックグラウンドで動作をします。
- 一度起動したサービスをユーザーが操作する必要がある場合、アクティビティから既存のサービスにバインドして、予め定義したインターフェースを介して行います。

ハンズオン

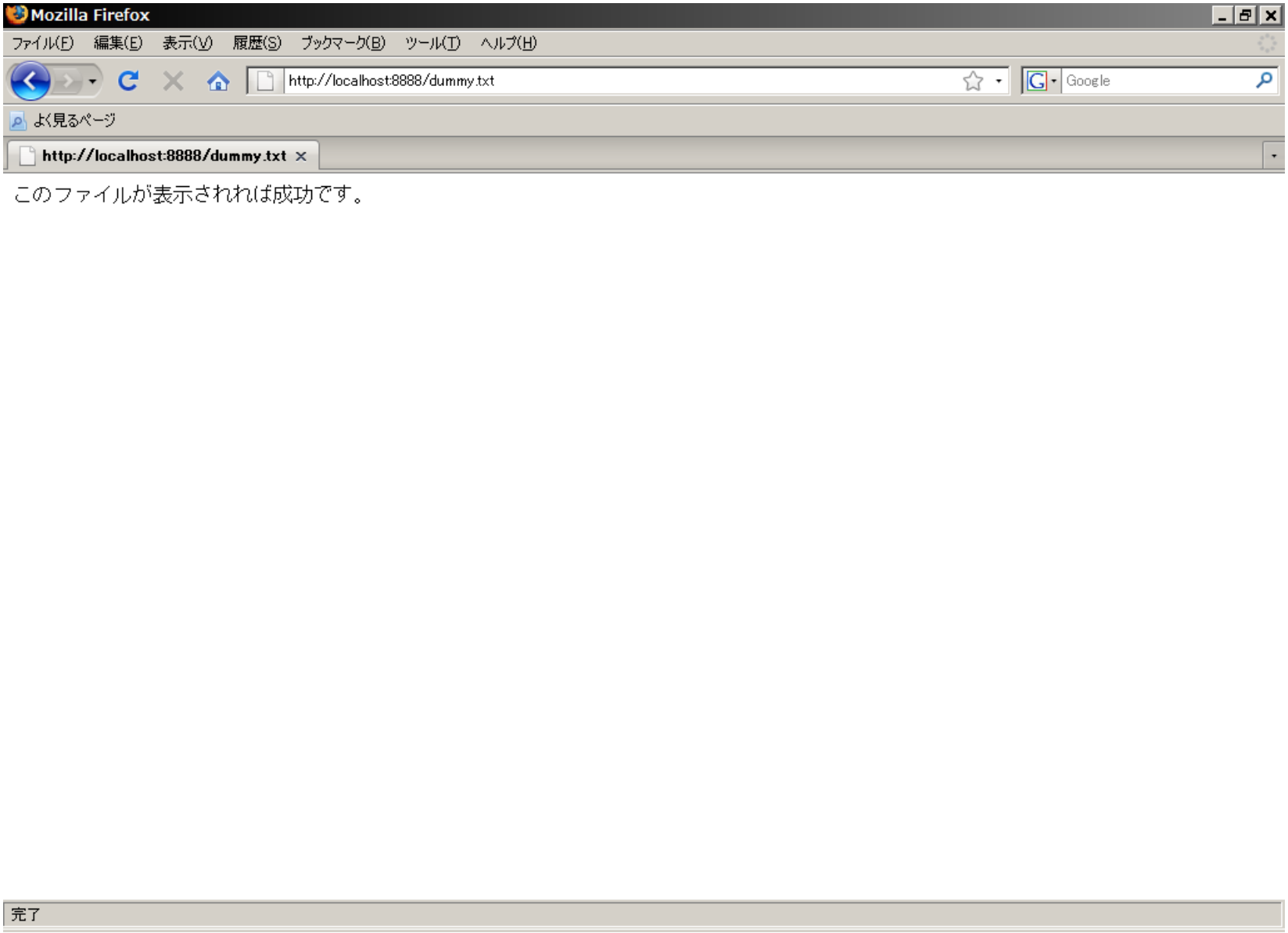
プロジェクトの準備

- **プロジェクトをダウンロード**
  - **「日本**Android**の会」のサイトより**
    http://www.android-group.jp/
- Eclipse**でインポート**

概要

- 簡単なHTTPサーバ（作りかけ）
  - 外部からブラウザで接続
- アプリケーションを起動
- adbで、ポートの転送を設定
  （エミュレータの場合）

  # adb forward tcp:8888 tcp:12700
- http://localhost:8888/dummy.txtにアクセス


Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
http://localhost:8888/dummy.txt
Google
よく見るページ
http://localhost:8888/dummy.txt
このファイルが表示されれば成功です。
完了

サービスを操作したい

- **画面を持つ**Activity**から、サーバー機能を止めたり、動かしたりしてみる。**
  - startServer()　　**サーバーを起動する**
  - stopServer()　　**サーバーを停止する**

Aidlファイルを追加する

- **パッケージ** ”jp.co.c_lis.ccl.http_server” **に**
  - IHttpServerService.aidl を新規で作成する
    - 通常のテキストファイルとして作成する。

IHttpServerService.aidlの内容

```
// Include your fully-qualified package statement.
package jp.co.c_lis.ccl.http_server;

// Declare the interface.
interface IHttpServerService {

    boolean startServer();
    boolean stopServer();

}
```

- **書き終えて、保存したら、一度プロジェクトをビルドする。**
- **エラーが無ければ、**IHttpServerService.java**が自動で生成される。**

サービス側の実装

- AIDLで定義されたインターフェースの実装
- バインドされた際の振る舞いを実装

追加するコード → HttpServerService

```
/**
 * サービスへ接続したActivityからの操作を受け取る
 * バインダインターフェース
 */
private class ServiceBinder extends IHttpServerService.Stub {
	/*
	 * (non-Javadoc)
	 * @see jp.co.c_lis.ccl.http_server.IHttpServerService#startServer()
	 */
	@Override
	public boolean startServer() throws RemoteException {
		Toast.makeText(HttpServerService.this, "startServer", Toast.LENGTH_SHORT).show();

		// サーバーを開始
		return start();
	}
	/*
	 * (non-Javadoc)
	 * @see jp.co.c_lis.ccl.http_server.IHttpServerService#stopServer()
	 */
	@Override
	public boolean stopServer() throws RemoteException {
		Toast.makeText(HttpServerService.this, "stopServer", Toast.LENGTH_SHORT).show();

		// サーバーを停止
	    return stop();
	}

}
```

追加するコード2 → HttpServerService

```
/*
 * (non-Javadoc)
 * @see android.app.Service#onBind(android.content.Intent)
 */
@Override
public IBinder onBind(Intent intent) {

        IBinder result = null;

        final String action = intent.getAction();

        if(action.equals(HttpServerService.class.getName())) {
            result = new ServiceBinder();
        }

        return result;
}
```

Activity側の実装

- **サービスへのバインド**
- **ボタンイベントの実装**

変更するコード → MainActivity

```
// サービスにバインドするインターフェース
private IHttpServerService _IHttpServerServiceBind = null;
private final ServiceConnection _serviceConnection = new ServiceConnection() {
        /*
         * (non-Javadoc)
         * @see android.content.ServiceConnection#onServiceConnected(android.content.ComponentName,
              android.os.IBinder)
         */
        @Override
        public void onServiceConnected(ComponentName arg0, IBinder arg1) {
              _IHttpServerServiceBind = IHttpServerService.Stub.asInterface(arg1);
        }

        /*
         * (non-Javadoc)
         * @see android.content.ServiceConnection#onServiceDisconnected(android.content.ComponentName)
         */
        @Override
        public void onServiceDisconnected(ComponentName arg0) {
        }
};
```

変更するコード2 → MainActivity

```
@Override
public void onCreate(Bundle savedInstanceState) {
        // 中略
        //
        // サービスの開始 – コメントアウトしてもしなくても良い
        //Intent intent = new Intent(this, HttpServerService.class);
        //this.startService(intent);

        // サービスへバインド、
            // サービスが起動していなければ起動する
            boolean serviceBindFlg = bindService(new Intent(HttpServerService.class.getName()),
           _serviceConnection, Context.BIND_AUTO_CREATE);

            if(serviceBindFlg == false) {
            Toast.makeText(this, "Cannot bind Service.", Toast.LENGTH_LONG).show();
            }

        //
        // 後略
}
```

変更するコード2 → MainActivity

```
@Override
public void onCreate(Bundle savedInstanceState) {
        // 中略
        //
        // サービスの開始 – コメントアウトしてもしなくても良い
        //Intent intent = new Intent(this, HttpServerService.class);
        //this.startService(intent);

        // サービスへバインド、
            // サービスが起動していなければ起動する
            boolean serviceBindFlg = bindService(new Intent(HttpServerService.class.getName()),
          _serviceConnection, Context.BIND_AUTO_CREATE);

            if(serviceBindFlg == false) {
            Toast.makeText(this, "Cannot bind Service.", Toast.LENGTH_LONG).show();
            }
        //
        // 後略
}
```

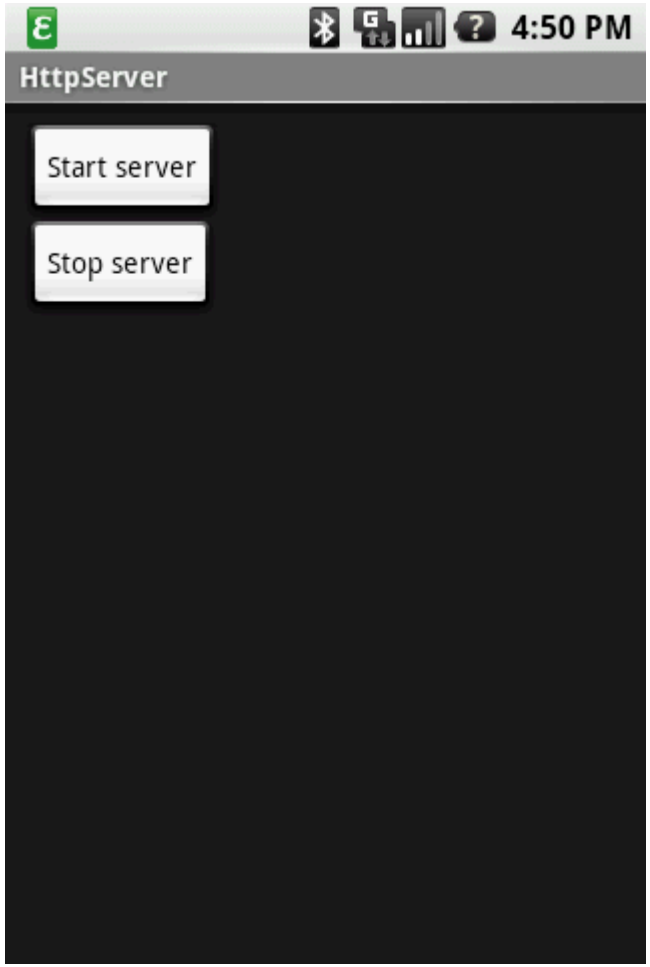
変更するコード3 → MainActivity

```
@Override
public void onCreate(Bundle savedInstanceState) {
	// 中略
	// StartServerボタン
	_mBtnStartServer.setOnClickListener(new OnClickListener() {
		@Override
		public void onClick(View arg0) {
			try {
				_IHttpServerServiceBind.startServer();
			} catch (RemoteException e) {
			}
		}
	});

		// StopServerボタン
		_mBtnStopServer.setOnClickListener(new OnClickListener() {
		@Override
		public void onClick(View arg0) {
			try {
			_IHttpServerServiceBind.stopServer();
			} catch (RemoteException e) {
			}
		}
	});
	//
	// 後略
```

エミュレーターでアプリを起動

- Start serverボタンを押すと、”startServer”と、表示されます。
- Start serverボタンを押すと、”stopServer”と、表示され、サーバー機能が停止します。



- サーバーをストップした後は、ブラウザでアクセスしてもコンテンツは表示されません。


ページ読み込みエラー - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
http://localhost:8888/dummy.txt
Google
よく見るページ
ページ読み込みエラー
接続が中断されました
ページの読み込み中にサーバへの接続がリセットされました。
ネットワーク接続の確立中にリンクが切れました。再度試してください。
再試行
完了

- **もう一度**”Start server”**ボタンを押すと、再び接続できるようになります。**

まとめ

- サービスとはシステムに常駐して、継続的な処理を行う際に利用されます。
- サービスは、AIDLでインターフェースを定義して、それを実装する事により、外部から操作する事が出来ます。

参考

- Android SDK
  http://developer.android.com/

- Android: Service : **小山 圭氏 発表資料**