

Build Own PAW Server

First of all some licensing information...

PAW components can be freely used in private or educational projects. For commercial use, please contact the author¹.

This document describes how to integrate PAW into your own application.

To make it easier, a sample project is provided which will be discussed inside this document.

If the sample project is missing please send a mail to: paw@fun2code.de

Project Structure

The sample project is a standard Android Eclipse project.

Note: There might be an error reported if the Android API level the project uses is not installed on your system. If that is the case, please install the appropriate Android version or change the API level inside the project settings.

This document will not go through all things common to all Android projects but will show what can be customized when running the server.

Here are the thing we will take a look at:

- *Manifest file*
- *Assets folder*
- *Java source files*

Manifest File

The Manifest file contains information about the application.

In this file permissions are of special interest.

Permissions set inside the sample project are the base permissions required by the server. If your app needs more than these permissions add them as needed.

The permissions that come by default are:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Assets Folder

The assets folder contains the content.zip file which includes all files the server needs to start.

This includes all the configuration and HTML files.

On server start up the ZIP file will be extracted to the INSTALL_DIR (see *Java Source Files*) location which can be SD Card or app's file location.

¹ Email: paw@fun2code.de

Java Source Files

The Java source files are located inside the src folder which belong to the package `de.fun2code.android.buildownpawserver`.

The package includes the files `BuildOwnPawServerActivity.java` and `BuildOwnPawServerService.java`.

The BuildOwnPawServerActivity Class

This class is a standard Android Activity and presents the GUI to the user. Important methods are:

- `onCreate()`
Standard Android `onCreate()` method. The `INSTALL_DIR` should be specified here. It is recommended to use external storage for testing because it makes debugging much easier. After testing is complete, the app's file directory should be used. Sample code is provided.
- `onResume()` and `onDestroy()`
These are mainly responsible to register/unregister the `ServiceListener` which calls `onServiceStart()/onServiceStop()` on service start/stop and to start and stop the service..
- `serverStart()` and `serverStop()`
These methods do basically what their names indicate, they start and stop the service.
- `onServiceStart()` and `onServiceStop()`
Called by the service when the service has been started or stopped. The method `onServiceStart()` writes the server's URL to a `TextView`. In this sample project `onServiceStop()` does nothing,

The BuildOwnPawServerService Class

This is the service that starts the server.

Of special interest is the `init()` method which sets the server parameters.

The following parameters are available:

<code>TAG</code>	Tag name for message logging.
<code>startOnBoot</code>	Indicates if service has been started on boot.
<code>isRuntime</code>	If set to true this will only allow local connections.
<code>serverConfig</code>	Path to server configuration directory.
<code>pawHome</code>	PAW installation directory.
<code>useWakeLock</code>	Switch <code>WakeLock</code> on or off.
<code>hideNotificationIcon</code>	Set to true if no notifications should be shown.
<code>execAutostartScripts</code>	Set to true if scripts inside the auto start directory should be executed on startup.
<code>showUrlInNotification</code>	Set to true if URL should be shown in notification.
<code>notificationTitle</code>	The notification title.
<code>notificationMessage</code>	The notification message.
<code>appName</code>	The Application name
<code>activityClass</code>	Activity class name
<code>notificationDrawableId</code>	ID of the notification icon to display.

Configuration Tips

The server configuration file is located inside the `conf/server.xml` file inside the `content.zip` file.

Port Number

To change the port number, alter the `<port>` element.

Another option is to let the server choose a random port (starting from 1024) to make sure an available port is selected. You can achieve this by defining the port as random:

```
<port>random</port>
```

File Upload Size

To change the upload size change the `<maxPost>` element. The size is specified in bytes. The default size is 2MB:

```
<maxPost>209715200</maxPost>
```