

# **Pepper CMS**

Implementing Pepper Apps

# Pepper CMS

- Information and Downloads:

<https://github.com/hanneseilers/PepperCMSBundle>

# Pepper CMS Bundle

- **demo-app**  
Demonstration app, see for app reference
- **demo-extionsion**  
Demonstration extension modul
- **doc**  
Documentation documents
- **pepper-cms**  
CMS-Core. Also repository for libraries.

# How to start?

1. Create a new Android app project
2. Robotify your project
3. Import *cms-lib* library module
4. Handle intent

# Android Robot Project

Create app project and robotify your project.

Follow instructions at:

[https://qisdk.softbankrobotics.com/sdk/doc/pepper-sdk/ch1\\_gettingstarted/starting\\_project.html](https://qisdk.softbankrobotics.com/sdk/doc/pepper-sdk/ch1_gettingstarted/starting_project.html)

# Import cms-lib

At the moment this step is not required!

Date: 2020/08/29

# Import cms-lib

- Download *cms-lib-release.aar* file from latest release:

<https://github.com/hanneseilers/PepperCMSBundle/releases>

# Import cms-lib

- Edit your projects build.gradle:

```
allprojects {  
    repositories {  
        google()  
        jcenter()  
        flatDir {  
            dirs 'libs'  
        }  
    }  
}
```



# Import cms-lib

1. Goto your projects directory.
2. Goto your modules (normally *app*) directory
3. Copy downloaded file to modules *libs* directory.

# Import cms-lib

- Edit your modules build.gradle:

Add:

```
implementation(name:'cms-lib-release', ext: 'aar')
```

# Handle intent

App is called via intent with extras.

Extras are simple JSON formatted `String` data.  
If needed you can parse the to `JSONObject` or any other library.

# Handle intent

App is called via intent with extras:

app

App information, including intent class and package information, as well as apps  
cms internal hashcode.

# Handle intent

App is called via intent with extras:

data

Apps stored data, as string. Normally use JSON formatted string for storing your applications data. CMS is storing a own configuration for every single use.

# Handle Intent

App is called via `android.content.Intent` with extras:

## user

Information about actual user. Also containing, name, birthday as long timestamp to create a `java.util.Date` object. Also internal username is added.

If extras string is empty, or not set, no user is registered.

# Handle Intent

- Tips:
  - Be aware, that even no extras data can be available. That means, the information is not available (for example not user found), or app was started without intent (without using pepper-cms). Implement a default behaviour for this case!
  - Parse json string to objects. Use `JSONObject` or `object` from *cms-lib* to work with them. For example `de.fhkiel.pepper.cms_lib.users.User` has a static function from `JSONObject` to parse `JSONObject` into `User` object.

# Handle Intent

- If you want to return, for example the user data. Use `android.app.Activity.setResult(Intent)` function to return an `android.content.Intent` with extras information as JSON formatted strings:

<code>app</code>	App data (as given with calling intent)
<code>user</code>	User data (as given with calling intent)
<code>data</code>	Data of actual user. Will be stored by the cms.