Contents

Version history

Version	Date	Description	Resp.
A	16/12-2016	First release	PB
B	20/01-2017	New template	JB

1 Overview

An application can use the number search capabilities of Calldorado. This is achieved by the application, by implementing a phone number field and a search action in the context of the application. The application can either control the Search Waiting UI, which is suggested in order to have a more native app experience (e.g. spinner while searching) or if not considered in the app, there will be a default handling in Calldorado. When search is finished, Calldorado will show the search result and inform the hosting app about the result.

2 Usage

For the best user experience, make sure the user have accepted the opt-in and have the Calldorado permissions, prior to executing the search function. If they have not, they will be granted/handled upon the search call.

The call take 2 parameters or optional a 3rd parameter

Calldorado.search(<ACTIVITY_CONTEXT>, <CDO_PHONE_NUMBER>, OPTIONAL <CALLBACK>)

If the callback is used, the application is expected to handle the UI to be shown to the end user during search, e.g show a waiting spinner and remove it when the search is finished. If the callback

is not used CDO will show a default spinner using the current theme while searching and remove it when the search is finished.

Examples of how the call could look:

```
Example 1:
Calldorado.search(activityContext, new CDOPhoneNumber("1111"));
```

```
Example 2:
Calldorado.search(activityContext, new CDOPhoneNumber("1111"), new
CDOSearchProcessListener() {
    @Override
    public void onSearchSent() {

    }

    @Override
    public void onSearchSuccess() {

    }

    @Override
    public void onSearchFailed(String errorMessage) {

    }
});
```

In Example 1, the search method is provided an activity context along with a new instance of a CDOPhoneNumber object where the construction parameter “1111” is the phonenumber as a string.

If possible, try to provide a country code to the input string. As an example let’s put in the country code 45(country code for Denmark), it would look like this “+451111” or “00451111”.

It will work with the phone number only, but the search will be done locally depending on the where the phone is placed. No spaces in the string are needed.

Example 2, is like example 1, but it has an interface with callbacks about the search process. This could be useful for providing your own search UI (e.g. a progressbar or other) and then show/hide it using the callbacks. If no CDOSearchProcessListener is provided (first line of code), we will show our own search UI.

3 UI/UX

If a CDOSearchProcessListener has been specified in the Calldorado.search() call, you are responsible for providing your own UI while the search is communicating with server. If none is specified, a default search screen will be shown. A successful search will automatically start an activity with the result.

4 Pitfalls

Please note that there is a limit of 4 searches a minute, if you exceed this, you will get an onSearchFailed callback with the error code: ERROR_SEARCH_LIMIT_PER_MINUTE_REACHED
If a phone number search can't find anything on the server, the onSearchFailed will be called with the code: ERROR_SERVER_NO_RESULT but an activity will still be shown to the user with the option of providing a name to this unknown number. So don't treat this as an error.

5 Class doc

5.1 CDOPhoneNumber

Is a wrapper class containing the phonenumber as a string

```
public class CDOPhoneNumber {
    private String phoneNumber;

    /**
     * Make an instance of this class and give a valid phone number as a string
parameter.
     * @param phoneNumber Valid phone number
     */
    public CDOPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }
}
```

5.2 CDOSearchProcessListener

Is an interface containing callbacks from the search

```
/**
 * Use this interface to get callback messages about the search process.
 */
public interface CDOSearchProcessListener extends Serializable {
    void onSearchSent();

    void onSearchSuccess();

    void onSearchFailed(String errorMessage);
}
```

5.3 ErrorCodes

A class containing the possible errors that get passed to the onSearchFailed callback

```
public class ErrorCodes {
    public final static String ERROR_OPT_IN_NOT_ACCTEPTED =
"ERROR_OPT_IN_NOT_ACCTEPTED";
    public final static String ERROR_NETWORK= "ERROR_NETWORK";
    public final static String ERROR_SERVER_NO_RESULT =
"ERROR_SERVER_NO_RESULT";
    public final static String ERROR_SERVER_COMM_FAILED =
"ERROR_SERVER_COMM_FAILED";
    public final static String ERROR_SERVER_FAILED_TO_PROCESS_RESPONSE =
"ERROR_SERVER_FAILED_TO_PROCESS_RESPONSE";
    public final static String ERROR_NEED_PERMISSION= "ERROR_NEED_PERMISSION";
    public final static String ERROR_NEED_PERMISSION_CANT_ASK_AGAIN=
"ERROR_NEED_PERMISSION_CANT_ASK_AGAIN";
    public static final String ERROR_PHONENUMBER_INCORRECT_LENGTH =
"ERROR_PHONENUMBER_INCORRECT_LENGTH";
    public static final String ERROR_PHONENUMBER_CONTAINS_ILLEGAL_CHARS =
"ERROR_PHONENUMBER_CONTAINS_ILLEGAL_CHARS";
    public static final String ERROR_PHONENUMBER_NULL =
"ERROR_PHONENUMBER_NULL";
    public static final String ERROR_SEARCH_LIMIT_PER_MINUTE_REACHED =
"ERROR_SEARCH_LIMIT_PER_MINUTE_REACHED";
}
```