Eva SDK For Android - v1.1

Introduction

Evature is a company that develops an Expert Virtual Agent (EVA), enabling free-text search for online travel.
Utilizing innovative algorithms to process search requests, EVA understands the users' inputs and converts them to structured search queries with very high precision.

EVA significantly improves conversion rates, revenues and user satisfaction.

Eva can be used in Web Sites and in Mobile Application. The Eva SDK for Android enables Android application to take advantage of the many benefits of Eva's search power.

The Eva SDK for Android is using the Eva APIS . Documentation for Eva API can be found at:
http://www.evature.com/docs/index.html

SDK Structure

The SDK comes with three directories:

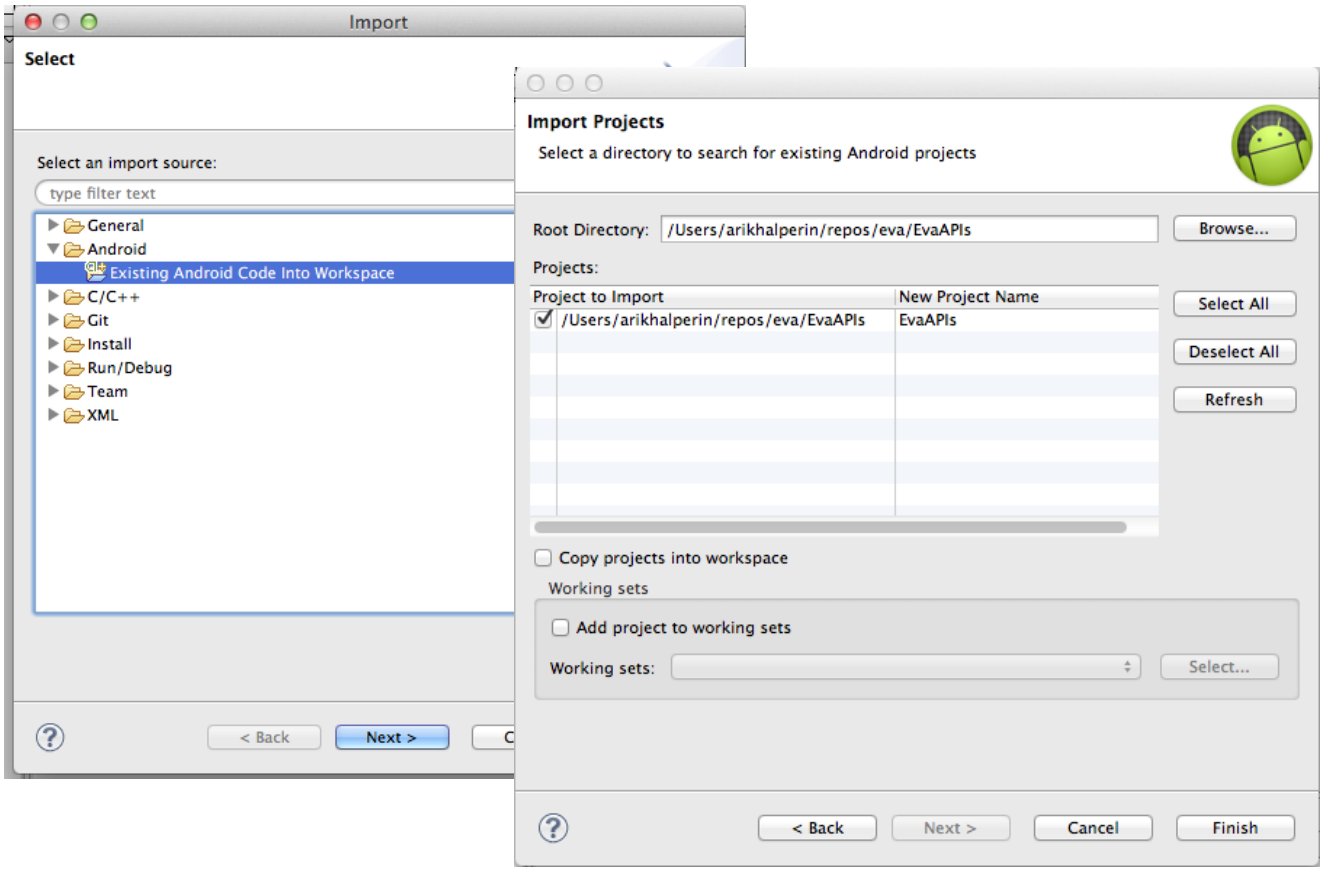
- **EvaAPIs**: The Eva SDK
- **android_demo**: A minimal demo that allows sending text to Eva and displays the response on the screen.
- **android_search**: A sample application which demonstrates Eva's usage for searching for hotels and flights. The application can search and book hotels using Expedia API and also search for a limited set of flights using Vayant FastSearch API. It has a Google map display, however to use it you must update the map key in the map view(hotel_map.xml)

Download or clone the SDK from:
https://github.com/evature/android/tree/v1.1

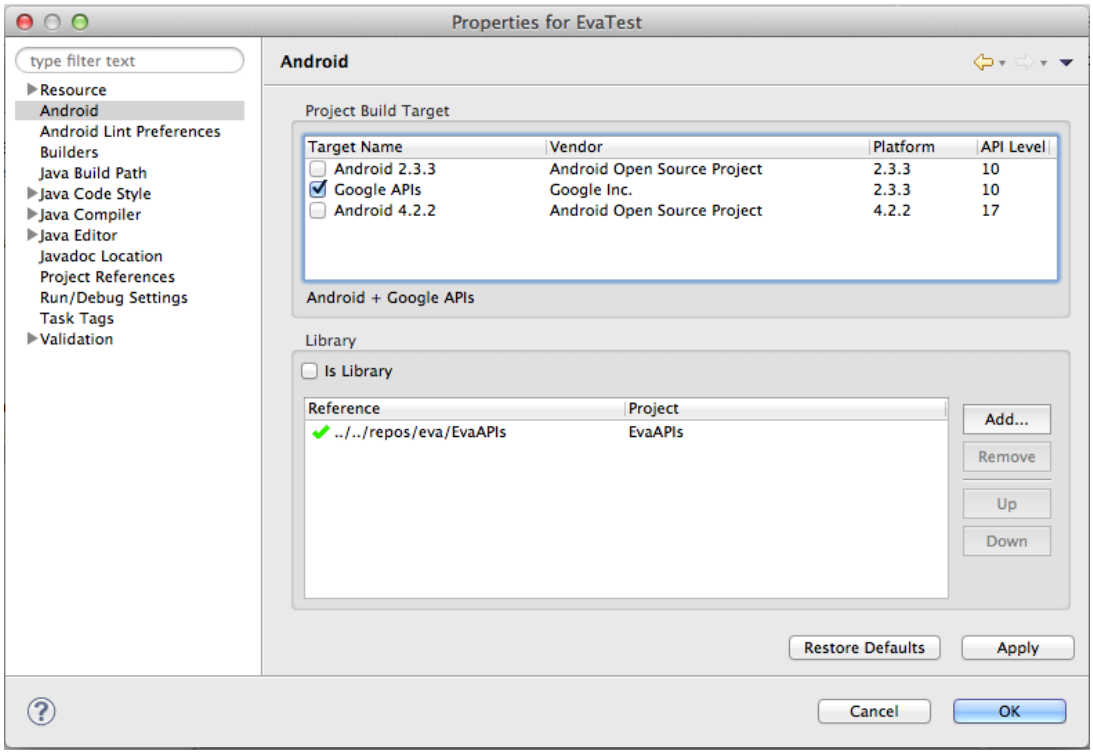
How to use the SDK

Creating a new project

After downloading the Evature SDK start Eclipse and import the SDK: select a workspace and import the EvaAPIs project to your workspace:



Start a new android project, and add EvaAPIs as a library:



Required permissions:

- android.permission.INTERNET
- android.permission.ACCESS_FINE_LOCATION
- android.permission.ACCESS_COARSE_LOCATION

If you want to use the voice based search you also need:

- android.permission.RECORD_AUDIO

Note: Eva SDK comes with Android Support Library v4. If you have it in your project and it causes problems, please delete it from the including project.

At this point you can start using Evature SDK.

Class Reference

EvaApiReply

Reply object for Eva, populated into java classes.
Documentation for it can be found at:
http://www.evature.com/docs/response.html#api-reply-in-depth

EvaBaseActivity

Base activity for activities that use the Eva SDK.

Note:
Many Android frameworks and libraries require you to extend your main activity from a base class (eg. RoboGuice, ActionbarSherlock, more…) . This is problematic since Java does not allow multiple inheritance and so only one such library can be used. Eva SDK is compatible with such libraries - you can use Eva without extending the base class but it requires more "wiring" and isn't recommended if not required. See the section below on using Eva without extending from a base class.

public methods:

void speak(String sayIt)

Uses TTS to say a string. Triggered a second *speak* while the first is still playing will stop the speech before playing the second one.

void searchWithVoice([Object cookie])

Asks Eva to start a voice dialog.

Note: To use *searchWithVoice* you need to add *com.evaapis.EvaSpeechRecognitionActivity* to your AndroidManifest.

If you wish to use Eva voice search, but do not want to present the default voice input dialog you can use EvaSpeechComponent (see below).

Parameters:

Object cookie: The cookie parameter is optional and isn't used - it will be returned with the search result callback.

void searchWithText(String text [, Object cookie])

Send a query to Eva with text instead of voice. The result will trigger a callback *onEvaReply* (see below).

Parameters:

String text: Text to query for

Object cookie: see above.

void replyToDialog(int replyIndex[, Object cookie])

When confronted with a multiple choice question in an Eva Reply, the answer chosen by the user should be sent using this function.

boolean isNewSession()

Returns true if the current state is of a new session.

void resetSession()

Start a new session.

Method Overrides:

public void onEvaReply(EvaApiReply reply, Object cookie)

Override this method to get the EvaReply from a voice or text search. Cookie is the object that was passed to the search request which triggered this reply.

public void newSessionStarted(boolean selfTriggered)

Called when a new session has started. If the parameter is true this is in response to a *resetSession* call, otherwise it means Eva's logic decided that the previous request started a new session.

EvaComponent

This class has setters and getters to configure Eva. By extending EvaBaseClass you get a protected member of this class called "eva".

Fields include:

apiKey, siteCode - the credentials to use Eva.

locale - see http://www.evature.com/docs/request.html#locale.

language - auto translate EvaReply, and also change the TTS language.

context - see http://www.evature.com/docs/request.html#scope

Advanced use cases

Using Eva without extending a base class

Many Android frameworks and libraries require you to extend your main activity from a base class (eg. RoboGuice, ActionbarSherlock, more…). This is problematic since Java does not allow multiple inheritance and so only one such library can be used.

To overcome this problem you can use Eva without extending a base class. To do so follow the following steps:

1. Create a com.evaapis.*EvaSearchReplyListener*. Simplest way would be to implement the interface in your Activity.
2. Create a field of type *EvaComponent* - the constructor requires two parameters: Activity and EvaSearchReplyListener.
3. In your activity's *onCreate, onResume, onPause, onActivityResult, onDestroy* methods call the matching methods on the *EvaComponent* field.
4. At this point you can use the methods described above for EvaBaseActivity - but activate them on the *EvaComponent* object.

Using Eva voice search without the default behaviour

When you activate *searchWithVoice* the default behaviour is to open a full screen activity with a GUI displaying the recording visualization and a big "stop" button. Successful replies trigger *onEvaReply* callbacks and erroneous requests (such as connection timeout) trigger a "Toast" notification with a user friendly message.

You may want to provide a different user experience in your application, such a playing a "beep" sound on recording start/end, and/or providing different visual feedback.

Triggering voice based search

If you wish to trigger voice search without the default behaviour you can use a lower level class: *com.evaapis.**EvaSpeechComponent.***

Public methods:

Constructor: Requires *EvaBaseActivity* or *EvaComponent*.

void start(SpeechRecognitionResultListener listener, Object cookie) -

Start the recording and uploading in a background thread. When silence is detected (after a period of non-silence) the recording is automatically stopped.
The cookie parameter is not used and passed back to the listener. EvaComponent implements this listener interface and provides the default behaviour described above.
The listener has two methods that only one of which will be called:

speechResultOK - this method is called when a valid response is returned. The first parameter is the response string that can be passed to EvaApiReply constructor.

speechResultError - called when there is an error (such as connection timeout) and the parameter is the error message. The default behaviour is to show the message in a Toast.

void stop() - stop the recording (before the silence detection kicks in) and process what was recorded.
void cancel() - stop the recording and ignore what was recorded.

Visual feedback of ongoing recording

While the recording is ongoing, you can poll Eva for data such as current volume or even a buffer of the previous samples. This data can be visualized on the screen, for example brighter background color on higher volume.

Data for visualization can be queried from the class *SpeechAudioStreamer* which can be retrieved from the *EvaSpeechComponent* using
Polling the recorder state and updating GUI can be done using *android.os.Handler* class.
For example:

```
speechSearch = new EvaSpeechComponent(eva);
updateLevel = new Handler()  {
   @Override
   public void handleMessage(Message msg) {
      SpeechAudioStreamer  speechAudioStreamer = speechSearch.getSpeechAudioStreamer();
      //   can read audio state from speechAudioStreamer here and update GUI based on
it
      sendEmptyMessageDelayed(0, 200); // poll the statistics every 200ms
      super.handleMessage(msg);
   }
};

// send initial message to start the polling
updateLevel.sendEmptyMessageDelayed(0, 100);

// start the recording
speechSearch.start(new SpeechRecognitionResultListener() {

   // stop the polling when the recording is complete
   @Override
```

```
   public void speechResultError(String message, Object cookie) {
      updateLevel.removeMessages(0); // stop the polling of statistics
      eva.speechResultError(message, cookie); // default handling of error - show a
Toast
   }

   @Override
   public void speechResultOK(String evaJson, Bundle debugData, Object cookie) {
      updateLevel.removeMessages(0); // stop the polling of statistics
      eva.speechResultOK(evaJson, debugData, cookie);
   }
}, "voice");
```

Eva provides a widget to visualize the wave-form of the recording, the class SoundView. To update it run:

```
soundView.setSoundData(  speechAudioStreamer.getSoundLevelBuffer(),
                         speechAudioStreamer.getBufferIndex(),
                         speechAudioStreamer.getPeakLevel(),
                         speechAudioStreamer.getMinSoundLevel()
                       );
```