Super Tag Trial SDK

Overview

The Super Tag Trial SDK gives developers a taste of the functionality offered by the full Super Tag SDK. With the trial SDK, developers can implement hashtag-based location sharing in their applications.

Limitations

The Super Tag Trial SDK only supports the sharing of locations. To access the ability to share all twelve types of information including products, movies, music, videos, and more, please license the full version of the SDK. Additionally, the trial SDK only supports the creation of intents that launch outside search applications (i.e. Google Maps). For native JSON support, the full version is required.

Platforms and Languages

Like the full version of the SDK, the trial SDK is developed for the Android platform. Written entirely in Java, the trial SDK has been compiled into a Java Archive (JAR) file.

For Android-specific questions, please consult the official Android documentation - http://developer.android.com/index.html.

What is a Super Tag?

A **Super Tag** is the combination of a **hashtag** - *an identifier for the type of information being shared* - followed by a **hashphrase** – *the information itself*. **Super Tags** allow end users to share media and information quickly and effortlessly through text communication.

Hashtags are categorized by different **functions** that determine how the information contained within a Super Tag is accessed.

For example:

> Message: "*Hey, meet me at* ***#location*** *The Statue of Liberty!*"
>
> The **#location** **hashtag** is associated with the **"location"** **function**. Therefore, the **hashphrase** - "The Statue of Liberty" is recognized as a location, and Google Maps is automatically invoked to search for it.
>
> Similarly, **#place**, **#address**, **#map**, **#maps**, **#at**, and **#to** are also **hashtags** that are associated with the **"location"** **function.**

Hashtags associated with a specific **function** all access information in the same way. Multiple **hashtags** are associated with every function simply for the sake of convenience and versatility of use.

For code-specific information regarding the **Super Tag** class (SuperTag.class), please consult the javadoc embedded within the "supertagtrial" java archive.

Super Tag Syntax

It is important to understand the proper syntax and conventions for **Super Tags** before integrating the SDK into your project.

A well-formed **Super Tag** begins with a **hashtag** denoting its function (Note: In the case of the full SDK, if the hashtag is not mapped to a supported **function**, it will default to a "generic search" **function**). Following the **hashtag** is a **hashphrase** that specifies the exact information being shared. Finally, the **Super Tag** ends with a recognized piece of ending punctuation or another **Super Tag**:

! . ? #another tag

Users may choose not to include ending punctuation if the **Super Tag** is at the end of a string or message.

Here are a few examples of well-formed **Super Tags** within messages (**hashtags** are **red and bolded**, and corresponding hashphrases are underlined**)**:

- *"Check out this **#product** Apple Macbook Pro"*
- *"Have you seen the **#video** nyan cat? It's really neat!"*
- *"You should watch the **#movie** American Hustle. It's great!"*

How are Super Tags created?

Super Tags are created via the **Super Text Analyzer**. This class analyzes strings of text and compiles lists of all the **Super Tags** they contain.

For example:

> Message: *"After the meeting today, we'll be holding a company dinner. Please meet **#at** Café Rouge!"*
>
> The **Super Text Analyzer** will recognize the **"#at" hashtag** and create a **Super Tag** object with the following attributes:
>
> - **Hashtag – #at**
> - **Function – location**
> - **Hashphrase –** "Café Rouge"

This process is automated by the **Super Linkifier** in order to create dynamic hyperlinks for all instances of **Super Tags** in a string (see page 6 – "Creating Super Tag Hyperlinks").

For code-specific information regarding the **Super Text Analyzer** class (SuperTextAnalyzer.class), please consult the javadoc embedded within the "supertagtrial" java archive.

What is a Super Intent?

A **Super Intent** is a custom Android intent created based on a **Super Tag.** The **Super Intent** is responsible for launching an appropriate application to access the information specified by a **Super Tag's hashphrase**.

For certain **functions**, only a single application is supported for accessing information. For example, Google Maps is the only application that can be launched via a **Super Intent** for a **Super Tag** with the **"location" function.**

In the case where multiple applications are supported, an Intent Chooser will be created, allowing the end user to choose between them.

For code-specific information regarding the **Super Intent** class (SuperIntent.class), please consult the javadoc embedded within the "supertagtrial" java archive.

How are Super Intents created?

Super Intents are created using the **Super Intent Creator** class. This object can be instantiated directly via the "getIntent" method within the **Super Tag** class (see javadoc). **Super Tags** can be passed into a **Super Intent Creator** in order to update them with **Super Intents** (also accessible via the "getIntent" method in the **Super Tag** class).

For code-specific information regarding the **Super Intent Creator** class (SuperIntentCreator.class), please consult the javadoc embedded within the "supertagtrial" java archive.

Creating Super Tag Hyperlinks

With the Super Tag SDK, creating dynamic hyperlinks is extremely straightforward. The **Super Linkifier** class provides methods for automatically hyperlinking any TextView based on the existence of **Super Tag** objects within a string.

If you are not familiar with Android's Linkify utility, please consult the official Android documentation.

Unless otherwise specified, the **Super Linkfier** creates hyperlinks with the preceding URL scheme **supertag://**. An activity must exist and be declared in your application's manifest in order to receive data passed prefaced with this scheme.

The built-in **Super Linkify Activity** (SuperLinkifyActivity.class) provides a simple mechanism for creating **Super Tags** and launching their corresponding **Super Intents** when hyperlinks are accessed. To use this activity, simply declare it in your application's AndroidManifest.xml as follows:

```
<activity
    android:name="com.ceazy.lib.STTrial.SuperLinkifyActivity"
    android:theme="@android:style/Theme.Translucent.NoTitleBar">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <data
            android:scheme="supertag" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

Alternatively, you may create your own activity to handle the launch process. The source for the **Super Linkify Activity** is available on GitHub.

For code-specific information regarding the **Super Linkifier** class (SuperLinkifier.class), please consult the javadoc embedded within the "supertagtrial" java archive.

Error Handling

For the purposes of the Super Tag Trial SDK, error handling is only necessary during the launch process of a **Super Intent**. When the "launch" method is called on a **Super Intent**, a Messenger object must be passed in for the purpose of monitoring the launch process for successful completion or the throwing of an exception.

In the case that the launch process completes successfully, a message with the attached object (msg.obj) *"Intent launched successfully"* will be sent through the specified messenger to its corresponding handler.

In the case that an error occurs, a special **Super Error** object is created containing an explanation of the error that occurred as well as a raw string version of the exception (exception.toString()). This **Super Error** object is sent to the handler via the following process:

```
SuperError error = new SuperError(explanation, errorString);
Bundle data = new Bundle();
data.putParcelable("error", error);
msg.setData(data);

try {
      messenger.send(msg);
      } catch (RemoteException e) {
            e.printStackTrace();
      }
```

It is recommended that the user be notified when the launch process has failed. For an example of proper error handling in the **Super Intent** launch process, please refer to SuperLinkifyActivity.class (available on GitHub).

For code-specific information regarding the **Super Error** class (SuperError.class), please consult the javadoc embedded within the "supertagtrial" java archive.

Final Notes

Thank you for trying the Super Tag SDK. Please consider purchasing the full SDK in order to use Super Tag for commercial use and gain access to its full feature set.

If you have any questions, comments, or concerns, please do not hesitate to contact us at supertagdev@gmail.com.