Spelling Checker Framework

In This Document

See also

- Spell Checker Service sample app
- Spell Checker Client sample app

The Android platform offers a spelling checker framework that lets you implement and access spell checking in your application. The framework is one of the Text Service APIs offered by the Android platform.

To use the framework in your app, you create a special type of Android service that generates a spelling checker **session** object. Based on text you provide, the session object returns spelling suggestions generated by the spelling checker.

Spell Checker Lifecycle

The following diagram shows the lifecycle of the spelling checker service:

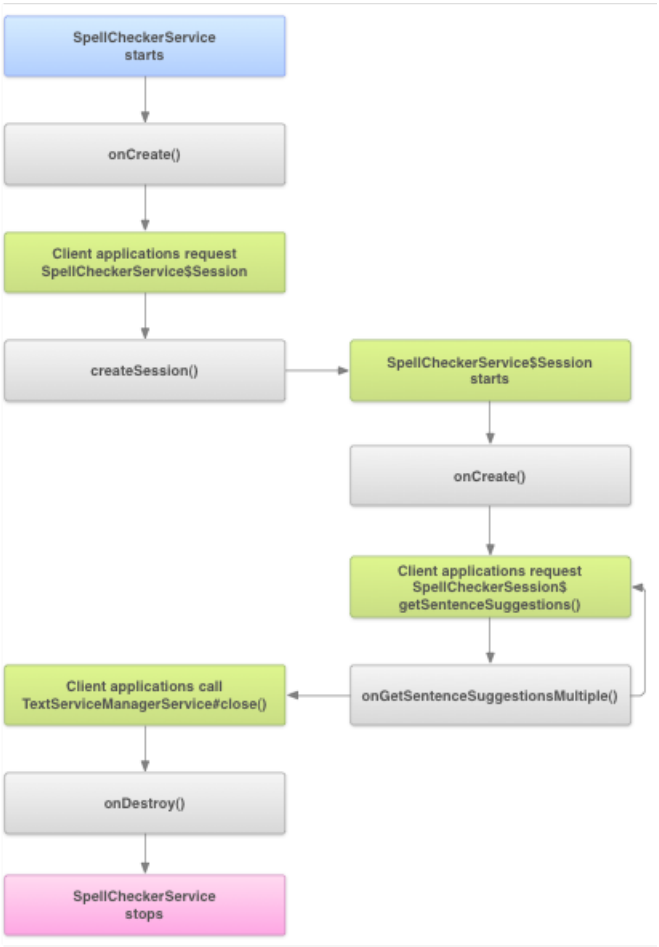

Figure 1. The spelling checker service lifecycle.

To initiate spell checking, your app starts its implementation of the spelling checker service. Clients in your app, such as activities or individual UI elements, request a spelling checker session from the service, then use the session to get suggestions for text. As a client terminates its operation, it closes its spelling checker session. If necessary, your app can shut down the spelling checker service at any time.

Implementing a Spell Checker Service

To use the spelling checker framework in your app, add a spelling checker service component including the session object definition. You can also add to your app an optional activity that controls settings. You must also add an XML metadata file that describes the spelling checker service, and add the appropriate elements to your manifest file.

Spell checker classes

Define the service and session object with the following classes:

A subclass of `SpellCheckerService`

> The `SpellCheckerService` implements both the `Service` class and the spelling checker framework interface. Within your subclass, you must implement the following method:
>
> `createSession()`
>
> > A factory method that returns a `SpellCheckerService.Session` object to a client that wants to do spell checking.
>
> See the Spell Checker Service sample app to learn more about implementing this class.

An implementation of `SpellCheckerService.Session`

> An object that the spelling checker service provides to clients, to let them pass text to the spelling checker and receive suggestions. Within this class, you must implement the following methods:
>
> `onCreate()`
>
> > Called by the system in response to `createSession()`. In this method, you can initialize the `SpellCheckerService.Session` object based on the current locale and so forth.
>
> `onGetSentenceSuggestionsMultiple()`
>
> > Does the actual spell checking. This method returns an array of `SentenceSuggestionsInfo` containing suggestions for the sentences passed to it.
>
> Optionally, you can implement `onCancel()`, which handles requests to cancel spell checking, `onGetSuggestions()`, which handles a word suggestion request, or `onGetSuggestionsMultiple()`, which handles batches of word suggestion requests.
>
> See the Spell Checker Client sample app to learn more about implementing this class.

> **Note:** You must implement all aspects of spell checking as asynchronous and thread-safe. A spelling checker may be called simultaneously by different threads running on different cores. The `SpellCheckerService` and `SpellCheckerService.Session` take care of this automatically.

Spell checker manifest and metadata

In addition to code, you need to provide the appropriate manifest file and a metadata file for the spelling checker.

The manifest file defines the application, the service, and the activity for controlling settings, as shown in the following snippet:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.samplespellcheckerservice" >
    <application
        android:label="@string/app_name" >
        <service
            android:label="@string/app_name"
            android:name=".SampleSpellCheckerService"
            android:permission="android.permission.BIND_TEXT_SERVICE" >
            <intent-filter >
                <action android:name="android.service.textservice.SpellCheckerService" />
            </intent-filter>

            <meta-data
                android:name="android.view.textservice.scs"
                android:resource="@xml/spellchecker" />
        </service>

        <activity
            android:label="@string/sample_settings"
            android:name="SpellCheckerSettingsActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Notice that components that want to use the service must request the permission `BIND_TEXT_SERVICE` to ensure that only the system binds to the service. The service's definition also specifies the `spellchecker.xml` metadata file, which is described in the next section.

The metadata file `spellchecker.xml` contains the following XML:

```
<spell-checker xmlns:android="http://schemas.android.com/apk/res/android"
        android:label="@string/spellchecker_name"
        android:settingsActivity="com.example.SpellCheckerSettingsActivity">
    <subtype
            android:label="@string/subtype_generic"
            android:subtypeLocale="en”
    />
    <subtype
            android:label="@string/subtype_generic"
            android:subtypeLocale="fr”
    />
</spell-checker>
```

The metadata specifies the activity that the spelling checker uses for controlling settings. It also defines subtypes for the spelling checker; in this case, the subtypes define locales that the spelling checker can handle.

Accessing the Spell Checker Service from a Client

Applications that use `TextView` views automatically benefit from spell checking, because `TextView` automatically uses a spelling checker. The following screenshots show this:

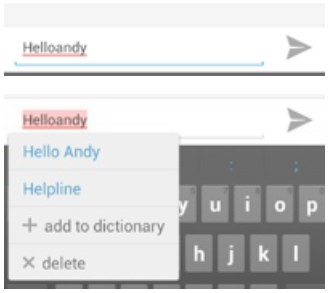

Figure 2. Spell checking in TextView.

However, you may want to interact directly with a spelling checker service in other cases as well. The following diagram shows the flow of

control for interacting with a spelling checker service:

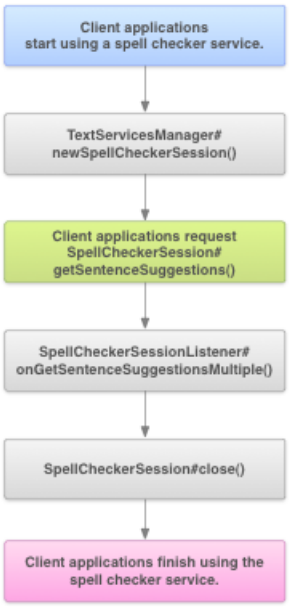

Figure 3. Interacting with a spelling checker service.

The Spell Checker Client sample app shows how to interact with a spelling checker service. The LatinIME input method editor in the Android Open Source Project also contains an example of spell checking.