简述

- 学习短信发送中

发送流程

- 打开短信应用会进入ConversationListActivity，位于packages/apps/Messaging/src/com/android/messaging/ui/conversationlist/ConversationlistActivity.java
- 创建新的消息，进入ConversationActivity，位于ui/conversation/ConversationActivity.java
  - 在onCreate()中调用updateUiState()创建ConversationFragment对象
- 在ConversationFragment的onCreateView获得ComposeMessageView并bind
  - ComposeMessageView：This view contains the UI required to generate and send messages.
  - ```
    public View onCreateView(final LayoutInflater inflater, final ViewGroup container,final Bundle savedInstanceState) {
         mComposeMessageView = (ComposeMessageView)view.findViewById(R.id.message_compose_view_container);
         // Bind the compose message view to the DraftMessageData

    mComposeMessageView.bind(DataModel.get().createDraftMessageData(mBinding.getData().getConversationId()), this);
    }
    ```
- 在ComposeMessageView的onFinishInflate()中获取发送按钮并绑定点击事件
  - ```
    private ImageButton mSendButton;
    protected void onFinishInflate() {
        mSendButton = (ImageButton) findViewById(R.id.send_message_button);
        mSendButton.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(final View clickView) {
                sendMessageInternal(true /* checkMessageSize */);
            }
        });
    }
    ```
  - 在sendMessageInternal（）中设置短信的内容和接收者，检查短信格式，根据结果执行相对应的操作
    - ```
      //ConversationFragment实现了IComposeMessageViewHost
      private IComposeMessageViewHost mHost;
      private void sendMessageInternal(final boolean checkMessageSize) {
            LogUtil.i(LogUtil.BUGLE_TAG, "UI initiated message sending in conversation " +
                    mBinding.getData().getConversationId());
            // Check the host for pre-conditions about any action.
            if (mHost.isReadyForAction()) {
      ```
      ```

```
            mInputManager.showHideSimSelector(false /* show */, true
/* animate */);
            //mBinding.getData()获得DraftMessageData对象，设置短信的内容
            final String messageToSend =
mComposeEditText.getText().toString();
            mBinding.getData().setMessageText(messageToSend);
            //设置短信的接收者
            final String subject =
mComposeSubjectText.getText().toString();
            mBinding.getData().setMessageSubject(subject);
            // Asynchronously check the draft against various
requirements before sending.
            mBinding.getData().checkDraftForAction(checkMessageSize,
                    mHost.getConversationSelfSubId(), new
CheckDraftTaskCallback() {
                @Override
                public void onDraftChecked(DraftMessageData data, int
result) {
                    mBinding.ensureBound(data);
                    switch (result) {
                        case CheckDraftForSendTask.RESULT_PASSED:
                            // Continue sending after check succeeded.
                            //将DraftMessageData对象变为MessageData对象
                            final MessageData message =
mBinding.getData().prepareMessageForSending(mBinding);
                            if (message != null &&
message.hasContent()) {
                                playSentSound();
                                //使用ConversationFragment类的
sendMessage()方法开始发送Message。
                                mHost.sendMessage(message);
                                hideSubjectEditor();
                                if
(AccessibilityUtil.isTouchExplorationEnabled(getContext())) {

 AccessibilityUtil.announceForAccessibilityCompat(
                                            ComposeMessageView.this,
null,
                                            R.string.sending_message);
                                }
                            }
                            break;
                    }
                }
            }, mBinding);
        }
    }
```

```
/*首先创建DraftMessageData类的内部类对象CheckDraftForSendTask,它继承了SafeAsync Task;接着调用此对象的executeOnThreadPool方法触发重写父类的三个方法调用onPreExecute、dolnBackgroundTimed 和onPostExecute,这几个方法的处理逻辑是发送短信的前置条件判断，最终通过mCallback.onDraftChecked调用将判断结果发送给CheckDraftTaskCallback对象。*/
public void checkDraftForAction(final boolean checkMessageSize,
final int selfSubId,final CheckDraftTaskCallback callback,
final Binding<DraftMessageData> binding) {
     new CheckDraftForSendTask(checkMessageSize, selfSubId,
callback, binding).executeOnThreadPool((Void) null);
}
```

- 在ConversationFragment类的sendMessage()中

```
public void sendMessage(final MessageData message) {
    if (isReadyForAction()) {
        if (ensureKnownRecipients()) {
            // Merge the caption text from attachments into the text
body of the messages
            message.consolidateText();
            //获得ConversationData，调用它的sendMessage()方法
            mBinding.getData().sendMessage(mBinding, message);
            mComposeMessageView.resetMediaPickerState();
        }
    }
}
```