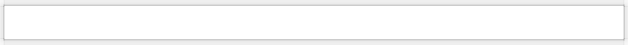
Dev Notes solutions for common problems

AUG
4

Android form validation - the right way

After quite a long pause, I'm getting back with new series of posts. And we'll start with Android EditText form validations.
At first lets define that Android form - it is set of EditText 's whose data should be processed. There are two main steps to make processing successful: data input and data validation.



Chuck Noris is always valid

Data input

In order to make user input easy and comfortable we need to configure EditText properly, here is some tips on it:

- put constraints by specifying the type of keyboard you want for your EditText with android:inputType attribute. For example, if you want the user to input only digits write android:inputType="number"
- besides specifying keyboard type using android:inputType attribute you can define other behaviors such as whether to capitalize all new words or use features like auto-complete and spelling suggestions, e. g. android:inputType="textCapWords|textNoSuggestions", note that you can put several markers using bitwise operator
- probably you'll want to disable fullscreen keyboard on landscape mode, you can do this using android:imeOptions="flagNoExtractUi" attribute
- pre-fill form data and provide auto complete if possible, for example in email registration form you can get user email from AccountManager, *example from Roman Nurik on stackoverflow*
- validate and submit form when user press enter on last EditText field, here code snippet I use:

```
    mTextView.setOnEditorActionListener(new TextView.OnEditorActionListener()
{
        @Override
        public boolean onEditorAction(TextView view, int actionId, KeyEvent event) {
            if (actionId == EditorInfo.IME_ACTION_DONE) {
                validateAndSubmit();
                return true;
            }
            return false;
        }});
```

Here is code snippet from layout file with described above attributes:

```
<EditText
        android:id="@+id/postal_address"
        ...
        android:inputType="textPostalAddress|textCapWords"
        android:autoText="false"
        android:editable="true"
        android:selectAllOnFocus="true"
        android:singleLine="true"
        android:imeOptions="flagNoExtractUi"
        android:hint="Enter postal address"
        ...
    />
```

Data validation

Actually question is not how to validate, but **how to show validation errors in a user friendly way**? Android API provide `public void setError (CharSequence error, Drawable icon)` method on EditText instance to display errors.
As form validation is quite commom task there are several third party libraries for that, e.g.:

- Android-Validator by Benjamin Besse
- Android Saripaar by Ragunath Jawahar

Both of this libraries use default API to display errors, personally I don't like it because when there are a lot of validation rules neglectful user can get screen like this:

Too much errors on one screen

So what is "*the right way*"? Main idea is to help user **solve errors one by one**. There are several things to do when validation error happens:

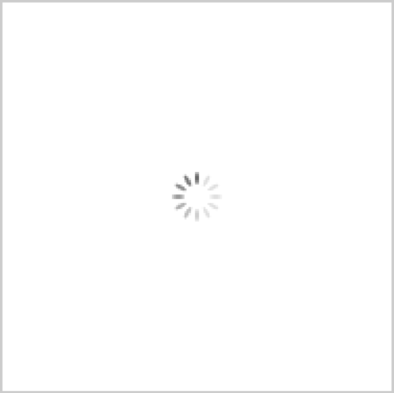
- find first text view where validation failed
- request focus for it: `mTextView.requestFocus()`
- show error message, I prefer using Crouton library: `Crouton.makeText(mActivity, message, Style.ALERT).show()`
- open keyboard so user can start typing instantly:

```
    InputMethodManager imm = (InputMethodManager) mContext.getSystemService(Context.INPUT_METHOD_SERVICE);
    if (imm != null) {
        // only will trigger it if no physical keyboard is open
        imm.showSoftInput(mTextView, 0);
    }
```

I've created small library with sample project where you can find all examples from this post, check it out on GitHub or install Demo app from Google Play. Sample code of creating validation form:

```
    mForm = new Form(mActivity);
    mForm.addField(Field.using(mName).validate(NotEmpty.build(mContext)));
    mForm.addField(Field.using(mEmail).validate(NotEmpty.build(mContext)).validate(IsEmail.build(mContext)));
    mForm.addField(Field.using(mAge).validate(InRange.build(mContext, 0, 120)));
```

```
    private void submit() {
        if (mForm.isValid()) {
            Toast.makeText(this, "Form is valid", Toast.LENGTH_SHORT).show();
        }
    }
```



Failed InRange validation

Useful links *(I'll update lists every time I find related topics)*

- Post about text input on d.android.com

Автор: Vitalik Zasadnyy, опубліковано 4th August 2013 тому

Мітки: android, best practices, edittext, form, validation

+1 Tweet Like

Переглянути коментарі