A SS OCIATE-ANDROID-DEVELOPER Q&As

Google Developers Certification - Associate Android Developer (Kotlin and Java Exam)

Pass Google ASSOCIATE-ANDROID-DEVELOPER Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/associate-android-developer.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Google Official Exam Center

QUESTION 1

For example, we have a file in our assets folder app/src/main/assets/sample_teas.json. To get an InputStream for reading it, from out Context context, we can try doing this:

A. InputStream input = context.getResources().openRawResource (R.raw.sample_teas);

B. InputStream input = context.getAssets().open("sample_teas.json");

C. InputStream input = context.getResources().getAssets().open ("sample_teas.json");

Correct Answer: B

QUESTION 2

What is a correct part of an Implicit Intent for sharing data implementation?

A. val sendIntent = Intent(this, UploadService::class.java).apply { putExtra(Intent.EXTRA_TEXT, textMessage) ...

B. val sendIntent = Intent().apply { type = Intent.ACTION_SEND; ...

C. val sendIntent = Intent(this, UploadService::class.java).apply { data = Uri.parse(fileUrl) ...

D. val sendIntent = Intent().apply { action = Intent.ACTION_SEND ...

Correct Answer: D

Create the text message with a string

val sendIntent = Intent().apply {

action = Intent.ACTION_SEND

putExtra(Intent.EXTRA_TEXT, textMessage)

type = "text/plain"

}

Reference:

https://developer.android.com/guide/components/fundamentals

QUESTION 3

When scheduling unique work, you must tell WorkManager what action to take when there is a conflict. You do this by passing an enum when enquing the work. For one-time work, you provide an ExistingWorkPolicy, which supports some options for handling the conflict. (Choose four.)

A. REPLACE (existing work with the new work. This option cancels the existing work)

B. KEEP (existing work and ignore the new work)

C. APPEND (the new work to the end of the existing work. This policy will cause your new work to be chained to the existing work, running after the existing work finishes)

D. APPEND_OR_REPLACE (functions similarly to APPEND, except that it is not dependent on prerequisite work status. If the existing work is CANCELLED or FAILED, the new work still runs)

E. APPEND_OR_KEEP (functions similarly to APPEND, except that it is not dependent on prerequisite work status. If the existing work is CANCELLED or FAILED, the new work still not runs)

F. APPEND_AND_RUN (functions similarly to APPEND, except that it is not dependent on prerequisite work status. If the existing work is PAUSED, the new work still runs)

G. DESTROY (if any work exists, the new work will be ignored)

H. APPEND_OR_DESTROY (if no any work exists, the new work will be ignored)

Correct Answer: ABCD

Videos:

1.

Working with WorkManager, from the 2018 Android Dev Summit

2.

WorkManager: Beyond the basics, from the 2019 Android Dev Summit

Reference: https://developer.android.com/reference/androidx/work/WorkManager?hl=en

QUESTION 4

Custom duration in milliseconds as a parameter for the setDuration method is available when you are working with:

A. Toast

B. Snackbar

C. for none of them

D. for both of them

Correct Answer: B

Reference:

https://developer.android.com/guide/topics/ui/notifiers/toasts https://developer.android.com/training/snackbar/action

QUESTION 5

For example, our preferences.xmlfile was added by addPreferencesFromResource(R.xml.preferences). Our

preferences.xmlfile contains such item:

In our Fragment, we can dynamically get current notification preference value in this way:

A. boolean isNotificationOn = PreferenceManager.getDefaultSharedPreferences(getContext()).getBoolean(getContext().getString(R.string.pref_notification_key),getContext().getResources().getBoolean(R.bool.pref_notification_default_value)

);

B. boolean isNotificationOn = PreferenceManager.getSharedPreferences(getContext()).getBoolean(getContext().getString(R.string.pref_notification_default_value),getContext().getString(R.string.pref_notification_key)

);

C. boolean isNotificationOn = PreferenceManager.getSharedPreferences(getContext()).getBoolean(getContext().getResources().getBoolean(R.bool.pref_notification_default_value),getContext().getString(R.string.pref_notification_key)

);

Correct Answer: A