

Application Development Fundamentals – Answer Guide

Instructions:

The answers in this guide address the  **Reflection & Notes:** questions in the eBook. Use this Answer Guide to self-check your answers and verify they are correct.

Module 1 – Application Development Overview

No questions in this module.

Module 2 – Application Analysis and Design

No questions in this module.

Module 3 – Create Application and Modules

Lab 3.2 Create an Application

D — Extending a table is a good idea. We'll be tracking requests and the tasks to fulfill those requests; many of these fields and features are already well-known on the Task table.

F.3.a — All application scopes can read, create, and update this table's records. The 'Accessible from' field is set to 'All application scopes' by default.

F.3.b — Yes, the records are accessible by web services since the 'Allow access to this table via web services' box is checked by default.

F.3.c — Some of these columns, such as 'Created' and 'Created by' appear on every table. The 'Number' column is there from the 'Auto-number' option having been selected. The remainder have been inherited from the Task table.

F.3.d — The new role 'loaner_request_user' was created while in the Guided Application Creator. Furthermore, while creating the table, 'Create Access controls' was selected. When that is selected, a role is required to create the access control rules. We elected to create a new one, though an existing role could have been used.

F.3.e — While creating the table, 'Create Access controls' was selected.

F.3.f — The Application Menu and modules were created as part of the Guided Application Creator (GAC) process. When creating a table *outside* of the GAC process, you can choose

whether to create a '*List of records*' link type module for your new table, and whether to place it in a new or existing Application Menu.

Lab 3.3 Link Loaner Application to GitLab Repository

A.18.a. Two commits have been done.

A.18.b - There is one branch.

Lab 3.4 Create Application Menu and Files

A.0a – The Create New module opens a Loaner Request form, and the other modules open in List view with no records.

A.3.a – The loaner request user role (x_cdltd_loaner_req.loaner_request_user)

A.5.a – (Q: Is there a Filter or Argument set?)

B.1.a – Yes, the order reflects the previous step, placing the items in order from 100 – 600.

E.1 – It is a good practice to review the list of Changed Files to be committed.

Module 4 – Application Forms

Lab 4.1 Working with Fields & Views

A.1.a — Remove the following fields:

Parent

Priority

Active

Add the following fields:

Requested for

Start date

End date

Item type

Location to be used

Depot

A.23.a — Closed Complete has a low value because it came from the choice list values as set for the Task [task] table.

B.5.a – Self Service

B.5.b – No, ESS was the value entered

B.5.c – ESS is the name for Self Service (Title)

The screenshot shows a ServiceNow UI interface. At the top, there is a navigation bar with a menu icon, a filter icon, a chat icon, and the text 'UI Views'. Below this is a search bar with a dropdown menu set to 'Name' and a search input field. The main content area shows a list of records. The first record has the value 'ess' in the 'Name' field and 'Self Service' in the 'Title' field. There are also search filters for 'Name' and 'Title' with input fields.

Lab 4.2a Create a UI Policy

C.3.a – Yes, the *Other* field appears when the Depot field is set to 'Other'.

C.4.a – Yes, the *Other* field appears when the Depot field is set to 'Other', and disappears when there is any other value in the Depot field.

E.3.a - Yes, the *Other* field appears when the Depot field is set to Other, and disappears when there is any other value in the Depot field.

E.3.b – It disappears when the Depot field is set to any value except 'Other'.

E.4.a – Twice; No, we do not want the message to appear more than once. We need to edit the UI policy. We will do this in the next section.

Lab 4.2b Scripting

B.7.a – Here is what the script does: If the configuration item field contains the word *blackberry*, *iphone*, or *android*, the Item type field is set to 'Mobile Phone' and read-only. If the configuration item field does not contain the word *blackberry*, *iphone*, or *android*, then Item type is not set to read-only.

B.9.a - If the regular expression match method was not used, the terms would be case sensitive—i.e., 'iphone' would match, but 'iPhone' would not.

C.3.a – The 'location_to_be_used' field will populate using the values from the scratchpad.

C.3.b – var city = g_scratchpad.city;

var country = g_scratchpad.country;

D.11.a – The field should have the values you set for ‘City’ and ‘Country code’ in the ‘Location to be used’ field.

E.3.a – If the end date is before the start date an error message will appear. The setAbortAction prevents the insert or update action from happening.

Module 5 – Controlling Access

Lab 5.1 Configure Application Security

A.30 – Roles and/or groups assigned to a specific user is user data. Studio does not support editing data. Studio is for building and managing the application structure and artifacts. Therefore, we needed to make the changes to the groups and roles assigned to Fred and the Sys Admin in the main ServiceNow window.

Lab 5.2 Set Application Access [Optional]

C.5 – No records should have experienced any change to their state value.

D.7 – Yes. Because we selected Can Update on the Loaner Request table, the Malicious Script was able to change the State values on three records that were not ‘Closed Complete’ previously.

Module 6 – Automating Work

Lab 6.1 Flow Designer to Manage Deployment States

I.a - The Loaner Laptop flow is triggered when a loaner request record is created or updated and the Item type is laptop with the State of Reserved.

I.8 - Wait For Condition; Waiting for the loaner request record’s State to be Deployment Prep (15).

Loaner Request LNR002155

Number: LNR002155 State: Post Deployment

* Requested for: Aileen Mottern * Start date: 2022-09-27 20:06:07

Configuration Item: *ANNIE-IBM * End date: 2022-09-30 20:06:13

Assigned to: Andrew Och Item type: Laptop

Location to be used: Jackson, US Depot: Santa Clara

Short description: Description:

Loaner Tasks						
	Number	Priority	State	Assigned to	Short description	Task type
✖	TASK0020889	3 - Moderate	Closed Complete	(empty)	Prepare the laptop for deployment	Loaner Task
✖	TASK0020892	3 - Moderate	Open		Perform the post deployment tasks	Loaner Task

J.4.a – Yes. Since the business rule sets the state to Reserved, it will trigger the Loaner Laptop flow as long as the item type is laptop.

Lab 6.2 Creating Application Properties

D.2.a – Yes, all three properties appear.

D.2.b – Yes, the properties are in order according to the values in the order field.

D.4.a – Yes, the values were saved correctly.

D.4.b – Yes, the Separator module appears.

D.5.a – No, Beth should not be able to see the Properties module. Beth does not have the application's admin role.

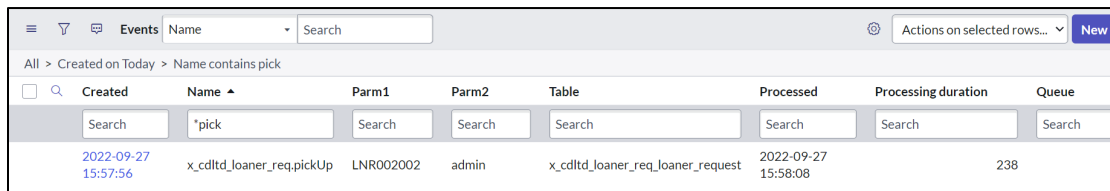
D.5.b – No, Beth should not be able to see the Separator module. Beth does not have the application's admin role.

Lab 6.3 Scheduled Script Execution and Email

B.7.a – Ready for Pickup (16)

B.7.b — The point of setting the `pick_up_reminder` variable in the while loop is to update each record that was returned/found by the query.

C.3.a — The GlideSystem server-side API `eventQueue()` method standard form is `gs.eventQueue("<event_name>", object, parm1, parm2);`



Created	Name	Parm1	Parm2	Table	Processed	Processing duration	Queue
2022-09-27 15:57:56	x_cdltd_loaner_req.pickUp	LNR002002	admin	x_cdltd_loaner_req_loaner_request	2022-09-27 15:58:08		238

Lab 6.4 Script Include and Commit

B.6.a — This line of script grabs the value of the `pickup.leadtime` property (created in Lab 6.2) the moment we invoke the Script Include, and stores it into a value called `this.p PickupLeadTime` which can then be used *throughout the entire Script Include's other functions* because it was done in the `initialize()` function.

B.6.b — As you can see in the code comments, this function gets items in State 'Ready for Pickup' that have not yet had a pick-up reminder sent.

B.6.c — This function fills in the Pick-up reminder field with the current date/time, and generates a pickUp event in the Event log to trigger the pick-up notification.

B.6.d — If you compare the original code in the Scheduled Script with the code in the two functions from the Script Include (`getNullPickupReminders()` and `sendPickupReminder()`), you will see they're very similar! The Script Include is more modular, though, which is a good practice from a standpoint of maintenance and reusability. Notice that the *new* code in the Scheduled Script uses `getNullPickupReminders()` to do what the first ten or so lines of the original Scheduled Script do (locating Loaner Requests that need to be picked up), and then `sendPickupReminder()` handles the remainder. The remainder of triggering an email notification for each such Loaner Request found by the previous function.

Module 7 – Importing and Integrating

Lab 7.2 Web Service Consumer

D.6.a — While we've provided the answer in the steps, if you'd like to know *why*, it's because all the other choices are *synchronous*, and we are making a web service call through the internet to another ServiceNow instance. That could take a moment and create a perceived slowness to the application. By using an *asynchronous* call, we're not asking the user to sit and wait for the other instance to get back to us.

Module 8 – Test with Automated Test Framework (ATF)

Lab 8.1 Automated Test Framework

F.a – The test was expected to fail since the end date is before the start date.

F.9.a – Yes, we expected the test to fail because the end date is before the start date.

G.a – The test failed at the Submit a Form due to the Set Field Values. The active business rule states the start date must be before the end date. It is the `setAbortAction()` in the business rule that prevents the Submit test step from being successful.

Module 9 – Additional Application Development Topics

Lab 9.2 [Optional] Outbound REST Message

C.2.a – Yes, the latitude was returned.

C.2.b – Yes, the longitude was returned.

C.2.c & d – When evaluating the accuracy of the data, consider the parameters. For example, the address string – how precise is it? Did you just provide a city, or a more focused street address?

D.7.a – We are performing a query where the Active field is true, and user record referenced by the Requested for field has a value in the City field.

D.12.a - XML. We are using the `.parseXML` method to decode the response body of the REST message.

D.12.b – `item.icon = "https://maps.google.com/mapfiles/ms/micons/yellow.png";`

F.3.a – Yes. Note - sometimes the publisher site (geocode.farm) is unreachable. If your map does not show markers, wait a few minutes and try again. The problem may not be with your code, but with the publisher you are attempting to connect to.

F.4.a - The loaner request record associated with that location, based on the user record location information (City and Country values) referenced by the Requested for field.