

SNPDG - Sprint 1 Tech Specs

Details & product documentation links for each story

Scoped app: An application scope is a protective barrier that prevents data from seeping out into other applications and secures your custom application from outside data, configurations, or customizations. 3 ways to create a scoped app (AES, studio, globe icon)

https://docs.servicenow.com/bundle/washingtondc-application-development/page/build/applications/concept/c_CreatingApplications.html

<https://docs.servicenow.com/bundle/washingtondc-application-development/page/build/app-engine-studio/task/create-app-scratch.html>(App Engine)

https://docs.servicenow.com/bundle/washingtondc-application-development/page/build/applications/task/t_AccessStudio.html(Studio)

Created update set: Creating an update set and not using the default one is important to cluster and capture all of the changes being made for a single story table or globe icon.

<https://docs.servicenow.com/bundle/washingtondc-application-development/page/build/system-update-sets/task/create-select-update-set.html>

Creating tables: For this project you will be creating 3 tables from scratch. Tables store data. 3 ways highlighting studio + the platform way - warning about naming inconsistencies when using AES

https://docs.servicenow.com/bundle/washingtondc-platform-administration/page/administer/table-administration/task/t_CreateATable.html

Form configuration: Once your tables are created you have the ability to add columns which will be the fields on the form. I suggest creating your columns/fields from the table record then going to the form of the newly created table and using form layout or form builder. I suggest using form layout or form builder versus using form designer. Please do not create fields in the form designer to avoid inconsistencies.

<https://docs.servicenow.com/bundle/washingtondc-platform-administration/page/administer/form-administration/concept/basic-form-administration.html>

<https://docs.servicenow.com/bundle/washingtondc-platform-administration/page/administer/form-administration/concept/configure-form-layout.html>

Menus and modules: Application menus and modules help navigate and organize your custom application. In the application menus table under system definition you have the ability to create an application menu that will populate in the all menu. From that record you can create and organize the modules that fall into the application menu.

Import sets: When trying to load data from outside sources into an existing table on the platform using import sets is one way to do so. The import set table created will be the housing area for records imported from a data source. Your target table is the table that the imported records will be mapped too. Transform maps are used to make the connection between import sets and target tables. Once the transform map is created you can use auto map or mapping assist to ensure the fields and data in those fields correlate properly. Make sure to hit the transform button located in the related links to solidify the transform.

https://docs.servicenow.com/bundle/tokyo-platform-administration/page/administer/import-sets/concept/c_ImportSetsKeyConcepts.html

https://docs.servicenow.com/bundle/tokyo-platform-administration/page/script/server-scripting/task/t_CreateATransformMap.html

https://docs.servicenow.com/bundle/tokyo-platform-administration/page/administer/import-sets/task/t_RunImport.html

Related Lists: Related lists show records that are in another table that have a relationship with the current record being displayed. Related lists can be added at the bottom of the form.

<https://docs.servicenow.com/bundle/washingtondc-platform-administration/page/administer/form-administration/concept/configure-form-layout.html> *how to add related list is included in the product doc*

UI Policy: Is a rule that is applied to a form to automatically change information or the form itself. Once a UI policy is saved **UI policy actions are used to set fields on a form too mandatory/optional, hidden/visible, and read-only/editable.** UI policies run

on the client side, which is the web browser side. UI policies are for managing user experience and data integrity.

Use the link below to see what each field is for on the UI policy form

https://docs.servicenow.com/bundle/vancouver-platform-administration/page/administer/form-administration/task/t_CreateAUIPolicy.html

UI Action: Is a button, link, or context menu item that appears on forms and lists. We configure UI actions to make the UI more interactive, customized, and specific to user activities.

Use the link below to find out what each field on the UI action form is for

https://docs.servicenow.com/bundle/vancouver-platform-administration/page/administer/list-administration/task/t_EditingUIAction.html

Catalog Item: Providing a self-service request opportunity for all catalog items come in many types. Define individual catalog items using the maintain items module. Next you would assign the item to additional catalogs and categories, if required -> define **variables** for the item, if applicable -> add any configurations and customizations using flow, ui policies, ui actions, and catalog client scripts.

Catalog item form

<https://docs.servicenow.com/bundle/washingtondc-servicenow-platform/page/product/service-catalog-management/reference/catalog-item-form.html>

How to create a catalog item

https://docs.servicenow.com/bundle/washingtondc-servicenow-platform/page/product/service-catalog-management/task/t_DefineACatalogItem.html

Types of catalog items

https://docs.servicenow.com/bundle/washingtondc-servicenow-platform/page/product/service-catalog-management/reference/r_VariableTypes.html

How to create a catalog

https://docs.servicenow.com/bundle/washingtondc-servicenow-platform/page/product/service-catalog-management/task/t_CreateAVariableForACatalogItem.html