**Module 4 – ServiceNow Scripting Fundamentals and Functions**

**Catalog Items & Record Producers:**

**Catalog Items:**

* Definition: Catalog items are offerings that users can request, such as hardware, software, or services, through the Service Catalog.
* Example: A "New Laptop Request" form where users can provide details (e.g., preferred laptop model, configuration, etc.). Once submitted, this request follows a workflow for approval and fulfillment.
* Use Case: End-users can select from various pre-configured items like laptops, software licenses, or access to specific services.

**Access Path:**

* Navigate to Service Catalog > Maintain Items to create or edit catalog items.
* End users access the items from Self-Service > Service Catalog.

**Record Producers:**

* Definition: Record producers allow users to create records in a specific ServiceNow table via simplified forms.
* Example: A "Create Incident" record producer might open an incident in the Incident Management module, triggered by a user reporting an IT issue.
* Use Case: Record producers streamline record creation by abstracting complex forms or workflows into easy-to-use interfaces.

**Access Path:**

* Go to Service Catalog > Record Producers to manage record producers.

**Client-Side and Server-Side Scripting:**

**Client-Side Scripting:**

Definition: Client-side scripts run in the user’s browser to control the behavior of forms, fields, and user input dynamically.

**Types of Scripts:**

* Client Scripts: Triggered by form events like load, submit, or field changes. They manage form validation, field behaviors, or data input in real-time.
* UI Policies: Control form field visibility, mandatory status, or read-only properties based on conditions.
* UI Actions: Add buttons or links that execute specific functionality on the form.

Example: Hiding a form field when a specific option is selected from a dropdown list.

**Access Path:**

* System UI > UI Scripts for custom scripts; use UI Policies and UI Actions within Forms.

**Server-Side Scripting:**

Definition: Server-side scripts run on the server and perform operations like querying or manipulating the database.

**Types of Scripts:**

* Business Rules: Automate actions when a record is inserted, updated, deleted, or queried.
* Script Includes: Reusable scripts that can be called by other server-side or client-side scripts.
* ACL (Access Control List): Control access to data by applying rules at the table or field level based on conditions.

Example: A business rule might query the user’s department and auto-populate a form field based on that department.

**Access Path:**

* System Definition > Business Rules for server-side logic; System Security > Access Control for ACLs.

**ACL (Access Control List) Scripting:**

* Definition: ACLs (Access Control Lists) regulate who can see, create, update, or delete records or fields in a table based on roles and conditions.
* Server-Side Scripting: ACLs are written on the server because they enforce data security by controlling access to records and fields stored in the database.
* Example: An ACL script might restrict access to the "Salary" field to only users with the "HR Admin" role.

**Access Path:**

* System Security > Access Control (ACL) to define table- or field-level security.

**Identifying Client-Side vs. Server-Side:**

**Client-Side:**

* Involves form validation, field visibility, or user interaction with forms.
* Example: Auto-filling a form field based on a dropdown selection before data is sent to the server.

**Server-Side:**

* Involves database operations like querying records or applying business logic.
* Example: Calculating a field value based on related records or applying complex validations after the user submits a form.

**Client-Side Modules:**

**Common Client-Side Modules:**

* Client Scripts: Triggered by events on forms (onLoad, onSubmit, onChange).
* UI Policies: Manage field behaviors (e.g., hiding, showing, making fields mandatory) dynamically.
* UI Actions: Add buttons or links to forms to trigger client-side functionality.
* Data Policies: Enforce data rules to maintain integrity across the platform.

**Access Path:**

* Navigate to System UI > UI Policies/UI Actions.

**Server-Side Modules:**

**Common Server-Side Modules:**

* Business Rules: Trigger on insert, update, delete, or query events in the database.
* Script Includes: Reusable server-side functions that can be called by other scripts.
* Background Scripts: Run one-time scripts to manipulate or query data without affecting user sessions.
* Fixed Scripts: Version-controlled scripts that can be transferred between environments.
* Email Scripts: Customize the content of emails based on specific conditions or records.

**Access Path:**

* System Definition > Business Rules for business logic, System Update Sets > Local Update Sets for fixed scripts.

**Scripting in Integrations:**

* Inbound & Outbound Integrations: Scripts transform or manipulate data when integrating with external systems via REST or SOAP APIs.
* Authentication: OAuth, API keys, or Basic Auth are often required to authenticate between systems.

**Handling Attachments:**

* Base64: Used for binary data like images or documents.
* Multipart/Form-Data: Supports sending multiple types of data (text and files) in one request.
* Scripted REST API: Custom REST endpoints within ServiceNow can process inbound data requests from external systems.

**Access Path:**

* System Web Services > Scripted REST APIs for building integrations.

**GlideAjax & Script Includes:**

* GlideAjax: Allows client-side scripts to asynchronously communicate with the server to fetch or update data without refreshing the page.
* Script Includes: Server-side scripts invoked by GlideAjax to fetch data or perform server-side logic.
* Example: A form might need to fetch a user’s location from the database based on a user ID entered in a field, triggered via GlideAjax.

**Access Path:**

* System Definition > Script Includes for server-side logic; System UI > Client Scripts for GlideAjax calls.

**Fixed Script:**

* Fixed Script: A server-side script that is intended to be run once for specific tasks, such as data migration or bulk updates.
* Key Differences from Background Script: Fixed scripts are versioned and can be moved across instances via update sets, while background scripts are typically ad-hoc and stay within the current instance.

**Access Path:**

* System Update Sets > Local Update Sets to version and manage fixed scripts.

**Handling Attachments (Base64 & Multipart/Form-Data)**

* Attachments play a significant role in ServiceNow integrations, especially when working with REST APIs or other external systems.
* Two common methods for handling attachments are Base64 encoding and Multipart/Form-Data.

**Base64 Encoding**

* Definition: Base64 is a method of converting binary data (like images, PDFs, or other file types) into text. This encoded format is useful for sending attachments over channels that only support text, such as certain API requests.
* Use Case**:**Sending/Receiving Binary Data: If you need to send or receive an attachment (e.g., a PDF or image file) through an API, the binary data is encoded into Base64 before transmission and decoded back on the receiving side.
* Example: A user uploads a PDF document through a ServiceNow form. Before sending this document to an external system via a REST API, it is Base64-encoded into a text string.

**Multipart/Form-Data**

* Definition: Multipart/Form-Data is used to send both text fields and file attachments in a single request. It allows the transmission of various content types (such as text and binary) in a single form submission.
* Use Case:Form Submissions: When a form includes both text fields and file uploads (such as uploading a resume and filling out application information), multipart/form-data ensures everything is sent together in a single request.
* Example: A user uploads an image and fills out a form with text fields like their name and email. Using multipart/form-data, the system can send both the image and form fields in one request to a REST endpoint.

**Access Path for Attachments:**

**View Attachments in Tables:**

Path: Go to any table with attachments enabled, such as Incident or Task, and find the paperclip icon next to the record to view or manage attachments.

**Enable Attachments:**

Path: In the Table Configuration, enable attachments by navigating to System Definition > Tables. Locate your table, then enable the Attachments option.

**Service Portal Scripting**

Definition: The Service Portal is a user-friendly web interface that allows users to interact with ServiceNow in a more intuitive way. It enables the creation of custom widgets, which are reusable components that display data or allow users to interact with forms or services.

**Client-Side Scripting in Service Portal:**

* AngularJS: Service Portal uses AngularJS for client-side scripting, allowing dynamic content updates without needing to refresh the page.
* Client-Side Controllers: Handle user input and UI logic, such as displaying information dynamically or validating form fields.

**Server-Side Scripting in Service Portal:**

* Server Scripts: Responsible for retrieving data from ServiceNow tables and processing it before sending it to the client-side.
* Data Providers: These scripts fetch the necessary data for widgets, typically from ServiceNow tables, and pass it to the client-side for display.

Example: A user accesses a custom widget on the Service Portal to view their pending approvals. The widget retrieves the data from the Task table using a server-side script and displays it using AngularJS on the client side.

**Access Path:**

**Service Portal Configuration:**

* Path: Service Portal > Service Portal Configuration
* Navigation: In the Application Navigator, type "Service Portal Configuration" to access and customize portal settings.

**Custom Widgets:**

* Path: Service Portal > Widgets
* Navigation: In the Application Navigator, type "Widgets" under Service Portal to create or edit widgets.

**Integrations**

* ServiceNow provides robust integration capabilities using REST and SOAP APIs, allowing it to exchange data with other platforms. Integrations typically involve both inbound and outbound data flows.

**Inbound Integrations**

* Definition: Inbound integrations bring data into ServiceNow from external systems. These integrations typically use the ServiceNow REST API or SOAP API to accept data and create or update records in ServiceNow.
* Use Case: Receiving incident updates from an external monitoring tool and creating or updating incidents in ServiceNow automatically.
* Example: ServiceNow receives an update from Jira via a REST API when a task is completed, which triggers the closure of the corresponding ServiceNow task.

**Outbound Integrations**

* Definition: Outbound integrations send data from ServiceNow to other systems. These may involve sending incident reports, status updates, or any other records via the REST or SOAP APIs.
* Use Case: Sending incident data from ServiceNow to an external third-party system for monitoring or reporting purposes.
* Example: Automatically sending incident status updates from ServiceNow to a Slack channel whenever the status of a critical incident changes.

**REST API & SOAP API**

**Key Differences:**

* REST API: Representational State Transfer; simpler and more flexible, typically using JSON or XML for data transfer. Preferred for modern integrations due to its lightweight nature.
* SOAP API: Simple Object Access Protocol; older and more complex, typically using XML. SOAP is used where strict security or legacy system compatibility is required.
* Data Transformation: Often, external systems use a different data structure than ServiceNow. To handle this, scripts are used to transform the incoming or outgoing data into the appropriate format.
* Example of Transformation: Transforming a JSON payload received from an external HR system into a format suitable for the User table in ServiceNow.

**Access Path:**

**REST API Explorer:**

* Path: System Web Services > REST > REST API Explorer
* Navigation: In the Application Navigator, type "REST API Explorer" to explore, test, and integrate with REST APIs.

**SOAP API Configuration:**

* Path: System Web Services > SOAP Message
* Navigation: In the Application Navigator, type "SOAP Message" to create or configure SOAP API integrations.

**What is ServiceNow?**

ServiceNow is a cloud-based platform that provides digital workflows for enterprise operations. It helps organizations automate various processes across different departments, such as IT, HR, security, and customer service. ServiceNow uses a single data model to deliver services efficiently by centralizing information and operations in one platform.

**Key features include:**

* IT Service Management (ITSM): Automating IT workflows like incident management, change management, and problem management.
* Customer Service Management (CSM): Managing customer interactions and requests.
* HR Service Delivery: Automating HR processes and workflows.
* Security Operations: Managing and responding to security threats and incidents.

**Services of ServiceNow**

ServiceNow offers several services, which can be categorized into:

* IT Service Management (ITSM): Automates IT workflows like incident management, service requests, and change management.
* IT Operations Management (ITOM): Optimizes service health by detecting issues and providing service visibility.
* IT Business Management (ITBM): Aligns work with business objectives, managing resources and project portfolios.
* HR Service Delivery: Streamlines HR workflows to improve employee experiences.
* Customer Service Management (CSM): Enables companies to resolve customer issues more efficiently.
* Security Operations: Offers a single platform for managing security incidents, threats, and vulnerabilities.

**How to Get Free ServiceNow Instances?**

ServiceNow provides free developer instances to anyone interested in exploring and learning the platform.

**Steps to get a free ServiceNow instance:**

* Go to the ServiceNow Developer Portal: developer.servicenow.com
* Sign up for a free account.
* Once logged in, click on Request Instance.
* Choose the desired version of ServiceNow and request a personal developer instance (PDI).
* These instances come with admin access, allowing you to practice and learn the platform features.

**Instance Activity**

* Active Users: Track the number of users currently logged into the instance.
* Instance Logs: Review system logs to understand performance or errors.
* Monitoring Performance: Use performance analytics to track how the instance behaves under load.

**How to Become a ServiceNow Developer?**

To become a ServiceNow developer,

* Learn JavaScript: ServiceNow scripting relies heavily on JavaScript.
* Explore the Developer Portal: Access free training, hands-on labs, and instances on the ServiceNow Developer site.
* Take Online Courses: Use platforms like Udemy, Coursera, or ServiceNow’s Learning Paths to learn modules like ITSM, CSM, and HR workflows.
* Practice with Developer Instances: Continuously work on your PDI, build custom apps, and develop hands-on experience.
* Get Certified: Start with Certified System Administrator (CSA) certification and work your way up to more advanced certifications.

**Career & Growth in ServiceNow**

ServiceNow offers significant career opportunities, especially as demand for professionals with expertise in enterprise workflow automation grows. Key roles include:

* ServiceNow Developer
* ServiceNow Administrator
* ServiceNow Architect
* ITSM Manager
* ServiceNow Business Analyst

**Career growth often includes:**

* Starting with ServiceNow Administrator and Developer roles.
* Moving into ServiceNow Architect or Lead Developer roles.
* High demand in industries such as finance, healthcare, and technology, with excellent salary prospects.

**ServiceNow Certification Training**

Certification in ServiceNow can enhance your career prospects. Common certifications include:

* Certified System Administrator (CSA): Foundation-level certification.
* Certified Application Developer (CAD): Advanced developer skills.
* Certified Implementation Specialist (CIS): Specializing in different areas like ITSM, CSM, HR, or SecOps.

**Training can be found on:**

* ServiceNow’s official website
* ServiceNow Developer Portal (for free courses)

**Voucher Code for ServiceNow CSA Exam**

Voucher codes are sometimes available for discounts on certification exams. These are usually distributed during promotions, events, or through ServiceNow partners. Check the ServiceNow Learning Portal or official announcements for active voucher codes.

**ServiceNow UI Overview**

This section likely covers the user interface (UI) components in ServiceNow, focusing on how to navigate the platform.

**Key areas in the UI:**

* Application Navigator: Lists all available modules.
* Content Frame: Displays the selected module or form.
* Banner Frame: Contains user settings, instance search, and branding.
* Forms & Lists: The UI components that display data from ServiceNow tables.

**ServiceNow Components**

The components of ServiceNow include:

* Tables: Store all data in the instance.
* Forms: Display individual records from a table.
* Lists: Display multiple records from a table.
* Workflows: Automate business processes.
* Business Rules: Execute server-side logic during database transactions.
* Client Scripts: Execute browser-side logic for form interactions.

Modules:

Modules are distinct areas within ServiceNow that provide access to different parts of the system. Some examples include:

* Incident Management
* Problem Management
* Change Management
* Service Catalog
* Asset Management

Each module has unique features to manage specific business functions.

**Incident Module**

The Incident Management module helps IT teams manage and resolve service interruptions or issues reported by users.

**Key elements include:**

* Incident Creation: Users can log incidents via the Service Portal or directly in the module.
* Incident Lifecycle: Assign incidents, resolve them, and track the status through various stages.
* SLA Management: Track Service Level Agreements (SLAs) for incident resolution times.

**Problem Module**

The Problem Management module focuses on identifying the root causes of incidents and preventing future issues.

**Key features:**

* Problem Creation: Problems can be created manually or from incidents.
* Root Cause Analysis: Investigate and document the underlying issues.
* Known Errors: Track and link problems to workarounds or solutions.

**Change Module**

The Change Management module manages changes to the IT environment while minimizing disruption to services.

**Key aspects:**

* Change Requests: Users request changes that need approval and assessment before implementation.
* Risk Management: Assess the potential risks of changes.
* Change Approval Process: Involves automatic or manual approval workflows.

**Lists**

Lists in ServiceNow represent sets of records from a table. They display data in a tabular format, allowing users to interact with multiple records simultaneously.

**Features include:**

* Filtering: Narrow down the list of records by specific criteria.
* Bulk Actions: Perform actions on multiple records at once (e.g., deleting or assigning).
* Customizable Columns: Adjust the columns displayed in a list to show relevant data.

**Forms**

Forms in ServiceNow display a single record from a table. They allow users to input, edit, or view data. Each form has fields that correspond to the columns in the table.

**Form Features:**

* UI Policies: Dynamically show/hide fields based on conditions.
* Client Scripts: Add interactivity, such as auto-filling data based on user selections.
* Related Lists: Show related records, like tasks linked to an incident.

Example: A user accesses the incident form to view or update the status, priority, and description of an incident record.