

TITOLO DEL DOCUMENTO

December 10, 2025

INDICE

1	Introduzione al Service Portal	5
1.1	Obiettivi del Modulo	5
1.2	Risultato Atteso	5
2	Cos'è un Portal?	5
2.1	Definizione	5
2.2	Funzionalità Principali	5
2.3	Branding e Tema	6
3	Anatomia di un Portal	6
3.1	Accesso e Struttura	6
3.2	Pagine e Contenuti	6
3.3	Containers, Rows e Columns	6
3.4	Widgets	6
4	Gestione della Personal Developer Instance (PDI)	6
4.1	Accesso e Panoramica	6
4.2	Attivazione Plugin	7
4.3	Gestione Email	7
4.4	Rilascio della PDI	7
4.5	Gestione Ruolo Utente	7
4.6	Rimozione Dati Demo	7
4.7	Gestione Password	7
4.8	Reset PDI	8
4.9	Aggiornamento PDI	8
5	Creazione di un Portal in ServiceNow	8
5.1	Accesso e Inizio	8
5.2	Campi Obbligatori	8
5.3	Configurazioni Principali	8
5.4	Aspetto e Branding	8
5.5	Liste Correlate	9
5.6	Suggerimento per Sviluppatori	9
6	Pagine del Portal	9
6.1	Struttura di una Pagina	9
6.2	Riutilizzo delle Pagine	9
6.3	Pagine Baseline	9
7	Creazione di una Pagina	9
7.1	Accesso al Designer	9
7.2	Configurazione della Nuova Pagina	10
8	Layout della Pagina	10
8.1	Struttura di Base	10
8.2	Aggiunta di Containers e Rows	10
8.3	Configurazione delle Colonne	10
8.4	Esempio di Layout	10
8.5	Nota	10
9	Configurazione delle Proprietà del Container	11

1 Introduzione al Service Portal

1.1 Obiettivi del Modulo

- Creare e configurare un portal in ServiceNow (nome, URL, impostazioni base).
- Utilizzare il Designer per creare e organizzare le pagine del portal.
- Aggiungere containers e layout alle pagine per definire la struttura responsive.
- Aggiungere widget alle pagine e posizionarli nei layout.
- Configurare le opzioni dei widget per personalizzare comportamento e contenuto.
- Eseguire la preview di una pagina durante lo sviluppo e testare diverse dimensioni dispositivo (desktop, tablet, mobile).
- Creare branding e un tema per il portal (colori, logo, font, stili).

1.2 Risultato Atteso

Al termine del modulo si dovrà essere in grado di:

- Progettare e pubblicare pagine di portal usando il Designer.
- Costruire layout responsivi con containers e colonne.
- Integrare e parametrizzare widget standard o personalizzati.
- Applicare un tema coerente e testare l'aspetto su vari dispositivi.

2 Cos'è un Portal?

2.1 Definizione

Un portal è un'interfaccia utente (UI) in ServiceNow costruita con il **Service Portal framework**, che offre un'esperienza alternativa rispetto alla UI standard. È progettato per essere intuitivo e accessibile da qualsiasi dispositivo: desktop, tablet o smartphone.

2.2 Funzionalità Principali

I portali permettono agli utenti di accedere a:

- Record selezionati da tabelle importanti (es. tutti i task assegnati all'utente)
- Metriche, report e analytics
- Service Catalog
- Knowledge Base
- Sondaggi
- Profilo utente
- Approvazioni
- Altri componenti della piattaforma

2.3 Branding e Tema

Il portal può essere facilmente personalizzato per riflettere il **branding** dell'organizzazione tramite temi, colori, logo e stili.

3 Anatomia di un Portal

3.1 Accesso e Struttura

Gli utenti accedono a un portal tramite un URL o un modulo, ad esempio:

`https://yourdomain.service-now.com/sp`

Un portal può essere configurato come **landing page** per l'utente al login. Ogni portal è composto da una o più pagine, che possono essere riutilizzate in più portal.

3.2 Pagine e Contenuti

Le pagine del portal hanno:

- Header
- Footer
- Theming
- Metadata aggiuntivi

3.3 Containers, Rows e Columns

Le pagine sono organizzate in **containers**, che definiscono il layout tramite **righe** e **colonne**:

- Un container contiene una o più righe
- Ogni riga ha un numero specifico di colonne
- Ogni colonna può contenere uno o più widget

3.4 Widgets

I widget definiscono il contenuto del portal. Ogni colonna può ospitare qualsiasi numero di widget, che visualizzano dati, form, report, o altre informazioni.

4 Gestione della Personal Developer Instance (PDI)

4.1 Accesso e Panoramica

Per gestire la PDI, accedi al **Developer Site** e seleziona *Manage my Instance*. Da qui puoi gestire impostazioni, plugin, dati demo e password.

4.2 Attivazione Plugin

1. Accedi al PDI tramite *Start Building*.
2. Naviga su *All > System Definition > Plugins*.
3. Cerca il plugin desiderato e clicca *Install/Update All*.
4. Se necessario, seleziona *Load demo data* e conferma con *Install*.

Nota: Alcune applicazioni del ServiceNow Store non sono installabili su PDI. L'installazione mostra lo stato *Activating Plugin* e invia una email al termine.

4.3 Gestione Email

- Vai su *Manage my Instance > Manage email properties*.
- Attiva/disattiva *Enable email sending and receiving*.
- Le PDIs inviano email solo a `guerrillamail.com` e possono ricevere da qualsiasi dominio eccetto lo stesso.

4.4 Rilascio della PDI

1. Se non serve più la PDI, seleziona *Actions > Release instance*.
2. Scegli un motivo e clicca *Release instance*.
3. La PDI viene reset e riassegnata ad un altro membro del Developer Program.

4.5 Gestione Ruolo Utente

1. Vai su *Actions > Change User Role*.
2. Seleziona *App Engine Studio Creator* o *Admin*.
3. Chiudi e riapri la PDI per vedere il cambio.

4.6 Rimozione Dati Demo

1. *Actions > Remove demo data*.
2. Conferma l'operazione. La rimozione richiede 20-30 minuti e invia email a completamento.

4.7 Gestione Password

- Visualizza, reimposta o invia via email le credenziali della PDI.
- Reimpostando la password, vengono aggiornati gli account *admin* e *aes.creator*.

4.8 Reset PDI

1. *Actions > Reset and wipe instance.*
2. Conferma l'operazione. Tutte le personalizzazioni andranno perse.
3. La PDI sarà pronta dopo alcune ore e una email confermerà la disponibilità.

4.9 Aggiornamento PDI

1. La PDI può essere aggiornata manualmente a nuove patch o release tramite *Upgrade release*.
2. L'aggiornamento può richiedere alcune ore e non può essere annullato.
3. Le patch mensili vengono applicate automaticamente.

5 Creazione di un Portal in ServiceNow

5.1 Accesso e Inizio

Per creare un nuovo portal, cliccare sulla tile **New Portal** nella pagina *Service Portal Configuration*.

5.2 Campi Obbligatori

- **Title:** Nome del portal, appare nell'header del browser e nei segnalibri.
- **URL suffix:** Suffix dell'URL per accedere al portal, es. /my_portal.

5.3 Configurazioni Principali

- **Homepage:** Pagina principale visibile dopo il login.
- **KB home page:** Pagina principale della Knowledge Base.
- **Login page:** Pagina di autenticazione utenti.
- **404 page:** Pagina predefinita se una pagina non può essere caricata.
- **Catalog home page:** Pagina del Service Catalog.
- **Main menu:** Menu nell'header del portal.

5.4 Aspetto e Branding

- **Theme:** Definisce lo stile e il branding, modificabile tramite *Branding Editor*.
- **Logo:** Logo nell'header.
- **Icon:** Icona nella barra dell'indirizzo.
- **Default:** Se selezionato, il portal diventa principale.

- **Quick start config:** Impostazioni generate dal Branding Editor.
- **CSS variables:** Variabili Sass specifiche del portal per sovrascrivere lo stile.

5.5 Liste Correlate

- **Search Sources:** Record da includere nella ricerca del portal.
- **Knowledge Bases:** Knowledge Base associate al portal.
- **Catalogs:** Service Catalog associati al portal.

5.6 Suggerimento per Sviluppatori

Specificare sempre quali Knowledge Base e Catalog includere. Se non viene specificato nulla, il portal userà tutte le Knowledge Base e tutti i Catalog.

6 Pagine del Portal

6.1 Struttura di una Pagina

Una portal page contiene **container**, **righe** e **colonne**, che a loro volta contengono **widget**. Gli sviluppatori progettano il layout della pagina e dei widget per costruire l'esperienza utente desiderata.

6.2 Riutilizzo delle Pagine

Le pagine possono essere riutilizzate tra diversi portal. Si possono usare le pagine fornite da Service Portal come **template** o punto di partenza per creare nuove pagine.

6.3 Pagine Baseline

Service Portal include pagine di base, come:

- **Landing page:** pagina di atterraggio predefinita del portal.
- **404 page:** pagina predefinita "Page not found", contenente un widget testuale e un widget di gioco *Breakout*.

7 Creazione di una Pagina

7.1 Accesso al Designer

Per creare una nuova portal page:

1. Cliccare sulla tile **Designer** nella pagina *Service Portal Configuration*. La freccia sulla tile indica che Designer si apre in una nuova scheda del browser.
2. In *Service Portal Designer*, cliccare sul link **Add a new Page**.

7.2 Configurazione della Nuova Pagina

- **Page title:** Titolo della pagina, appare come *<Portal Title> - <Page Title>* nei tab e nei segnalibri.
- **Page ID:** Identificatore unico della pagina. Non usare spazi o caratteri speciali diversi dagli underscore. ServiceNow suggerisce un ID basato sul titolo, modificabile dall'utente.

8 Layout della Pagina

8.1 Struttura di Base

Una portal page è composta da:

- **Containers:** dividono la pagina in sezioni.
- **Rows:** righe all'interno dei container.
- **Columns:** colonne all'interno delle righe, che utilizzano il **Bootstrap Grid System** a 12 colonne.

8.2 Aggiunta di Containers e Rows

1. Per aggiungere un container, trascinarlo dal *Layouts* alla pagina.
2. Per aggiungere una row a un container, trascinare la row dal *Layouts* nel container desiderato.

8.3 Configurazione delle Colonne

- Ogni row specifica quante colonne contiene e la larghezza di ciascuna.
- La larghezza di una colonna è indicata dal numero di colonne della griglia di Bootstrap che occupa.

8.4 Esempio di Layout

- Primo container: due rows
 - Una row con due colonne
 - Una row con tre colonne
- Secondo container: ancora senza rows

8.5 Nota

Le colonne delle row possono avere numeri diversi di colonne. Nell'interfaccia del Page Designer, gli indicatori delle colonne sono grigi con linee più sottili.

9 Configurazione delle Proprietà del Container

9.1 Selezione del Container

Per configurare un container:

1. Selezionare il container nella pagina. Un rettangolo blu e una sfumatura grigio scuro indicano che il container è selezionato.
2. Cliccare sul pulsante **Edit Container Properties** per aprire la finestra di configurazione.

9.2 Proprietà del Container

- **Name:** Nome del container, formato *<page name> - <container number>*.
- **Order:** Ordine di visualizzazione del container nella pagina.
- **Page:** Page ID della pagina del container.
- **Screen reader title:** Titolo alternativo letto dai screen reader per accessibilità.
- **Width:** Larghezza del container, *fixed* o *fluid*. Un layout fluido si adatta alla larghezza del browser.
- **Parent class:** Classe CSS genitore da cui ereditare lo stile.
- **CSS class:** Classe CSS specifica del container.
- **Background color:** Colore di sfondo del container, selezionabile o in esadecimale (es. #000000 per nero).
- **Background image:** Immagine di sfondo. Testo e altri elementi vengono renderizzati sopra.
- **Background style:** Modalità di visualizzazione dell'immagine di sfondo.
- **Bootstrap alternative:** Rimuove lo styling Bootstrap dal container.
- **Move to header:** Fissa il container nell'header, in modo che non scorra quando l'utente scorre la pagina.

10 Widgets

10.1 Definizione

I **widgets** sono componenti riutilizzabili che costituiscono la funzionalità di una portal page. Ogni widget include:

- HTML markup
- CSS
- Controller JavaScript
- Codice lato client e lato server

I widget definiscono le azioni del portal e le informazioni visualizzate agli utenti.

10.2 Widgets di Base

ServiceNow fornisce numerosi widget di base, ad esempio:

- Approvals
- Knowledge Base
- My Requests
- Carousel
- Catalog content
- Popular questions
- Search

10.3 Widgets Personalizzati

Gli sviluppatori possono creare widget personalizzati. Per approfondire, consultare il modulo formativo *Creating Custom Widgets*.

11 Aggiunta di Widget a una Pagina

11.1 Trascinamento dei Widget

Dopo aver definito il layout della pagina con containers, rows e columns:

1. Selezionare un widget dalla lista **Widgets**.
2. Trascinarlo nella colonna desiderata sulla pagina. Una evidenziazione grigia indica una posizione valida per il widget.

11.2 Gestione dei Widget nelle Colonne

- Le colonne possono contenere uno o più widget, in base al design della pagina.
- Esempio:
 - Colonna 1: 1 widget
 - Colonna 2: 1 widget
 - Colonna 3: 2 widget

11.3 Nota

Una colonna può contenere più widget senza limiti fissi, secondo le necessità del layout.

12 Opzioni dei Widget

12.1 Configurazione delle Opzioni

Molti widget hanno opzioni configurabili dall'utente. I campi e i valori possibili dipendono dal tipo di widget. Una lista di widget di base e delle loro opzioni è disponibile nella documentazione ServiceNow.

12.2 Istanza Unica del Widget

Quando un widget viene aggiunto a una pagina, si crea un'istanza unica del widget:

- Modificando le opzioni di quell'istanza, si cambiano solo le opzioni di quella specifica istanza.
- Esempio: modificare le opzioni del widget *Homepage Search* cambia solo quell'istanza, non tutte le istanze dello stesso widget.

12.3 Selezione e Modifica del Widget

1. Cliccare sul widget per selezionarlo.
2. Verificare la selezione tramite:
 - Breadcrumb nel banner
 - Nome del widget in alto a sinistra
3. Cliccare su **Edit Widget Properties** per aprire la finestra di configurazione.

12.4 Differenze tra Widget Default e Personalizzati

- Widget default e personalizzati possono avere **apparenza diversa** ma **stesso comportamento**.
- Cambiare le opzioni di un'istanza non influenza altre istanze dello stesso widget.

12.5 Anteprima e Modifica

- Dopo la configurazione, usare la tab **Preview** per vedere la pagina come un utente.
- Per continuare la modifica, tornare alla tab **Edit**.

13 Pagine Responsive

13.1 Adattamento a Dispositivi e Schermi

I layout delle pagine del Service Portal sono **responsive**:

- La larghezza delle colonne si adatta dinamicamente.
- Le colonne possono impilarsi se lo spazio disponibile è limitato.

13.2 Esempi di Visualizzazione

- **Desktop full screen:** tutte le colonne della pagina sono affiancate, avatar e nome utente visibili.
- **Desktop a larghezza parziale:** alcune colonne si impilano; il nome utente può non apparire.
- **Smartphone:** tutte le colonne sono impilate in una sola colonna; informazioni utente ridotte.

13.3 Concetto di Viewport

Le pagine devono essere considerate in termini di **viewport**:

- **Extra Small (XS)**
- **Small (S)**
- **Medium (M)**
- **Large (L)**

La dimensione del viewport è determinata dalla larghezza della pagina in pixel. Quando il viewport cambia, gli elementi si ridimensionano per rimanere accessibili senza scroll non necessari.

14 Container Fixed vs. Fluid

14.1 Container Fixed

- Gli elementi mantengono la stessa dimensione quando la larghezza della pagina cambia.
- Lo spazio tra elementi e bordo della pagina può variare.
- Esempio: un campo di ricerca rimane della stessa larghezza anche se la pagina si allarga o si restringe.

14.2 Container Fluid

- Gli elementi hanno dimensioni proporzionali alla larghezza del container.
- Quando la pagina cambia larghezza, anche gli elementi si ridimensionano.
- Esempio: un campo di ricerca rimane una percentuale della larghezza del container, adattandosi al cambiamento di larghezza.

14.3 Configurazione

Usare il campo **Width** nelle opzioni del container per specificare se il container deve essere *Fixed* o *Fluid*.

15 Mostrare e Nascondere i Containers

15.1 Motivazione

Alcuni container e widget diventano inutilizzabili o occupano troppo spazio su viewport piccoli o extra small. In questi casi, è utile nasconderli.

15.2 Esempi di Comportamento

- Nel Portal Menu Homepage, il widget **Homepage Search** e il suo container scompaiono quando non possono essere renderizzati in modo utilizzabile.
- Su viewport piccoli, il widget **Typeahead Search** può sostituire il widget Homepage Search.
- Su dispositivi mobili, tutte le colonne dei container vengono impilate.

15.3 Configurazione tramite Parent Class

- Usare il campo **Parent class** nelle opzioni del container per specificare quando un container deve apparire o scomparire.
- Esempi:
 - ‘hidden-xs’ → nasconde il container su viewport extra small
 - ‘visible-xs’ → mostra il container su viewport extra small
- Si possono usare le classi helper di Bootstrap per la visibilità responsive, come ‘hidden-md’, ‘visible-lg’, ecc.

16 Page Editor

16.1 Accesso al Page Editor

Per aprire il Page Editor:

1. Cliccare sulla tile **Page Editor** nella pagina *Service Portal Configuration*.
2. Selezionare una pagina dalla lista a tendina.

16.2 Vista ad Albero della Pagina

- Page Editor mostra una **vista gerarchica ad albero** di tutti gli elementi della pagina.
- Cliccando su un elemento del tree, è possibile esaminare opzioni, informazioni e HTML dell'elemento.
- Le informazioni disponibili dipendono dal tipo di elemento selezionato.

16.3 Esempio

Esaminando un'istanza del widget **Homepage Search**, il tree consente di visualizzare e modificare le proprietà specifiche di quell'istanza.

17 Branding di un Service Portal

17.1 Accesso al Branding Editor

Per applicare branding e temi a un portal:

1. Cliccare sulla tile **Branding Editor** nella pagina *Service Portal Configuration*.
2. Selezionare il portal da modificare dalla lista a tendina.

17.2 Quick Setup

- Permette di impostare rapidamente:
 - **Portal Title**: titolo del portal
 - **Logo**: immagine del logo
 - **Logo Padding**: spaziatura intorno al logo
- I campi sono già popolati al momento della creazione del portal e possono essere modificati anche dalle opzioni del portal stesso.

17.3 Theme Colors

- Impostare i colori per:
 - Navbar
 - Brand
 - Testo
- I valori di default derivano da Twitter Bootstrap.
- È possibile usare il **color picker** o specificare colori in esadecimale.
- Le modifiche si applicano solo al portal selezionato.

17.4 Theme Preview e Reset

- **Theme Preview**: aggiorna in tempo reale per verificare se i colori combinano bene.
- **Reset Changes**: ripristina i colori precedenti; disponibile solo se sono state effettuate modifiche.

18 Recap del Modulo di Introduzione a Service Portal

18.1 Concetti Chiave

- **Service Portal**: framework di ServiceNow per creare portali con un'interfaccia utente alternativa.
- **Service Portal Configuration**: accesso rapido a:
 - Branding Editor
 - Designer
 - Page Editor
 - Widget Editor
 - New Portal
 - Get Help
- **Page Designer**: crea le pagine del portal.
- **Struttura delle pagine**:

- Containers
 - Rows e Columns
 - Widgets
- **Opzioni dei widget:** dipendono dal tipo di widget.
 - **Branding Editor:** modifica loghi e temi del portal.
 - **Responsività:** gestisci la visualizzazione delle pagine su diversi dispositivi.

19 Notifications Objectives

19.1 Obiettivi del Modulo

- Creare notifiche email automatiche.
- Inviare notifiche in risposta a modifiche di record e a eventi di sistema.
- Effettuare il debug delle notifiche.
- Creare e utilizzare script per Email Notification.
- Utilizzare le System Mailboxes per analizzare messaggi inviati, ricevuti e scartati.
- Gestire le email in ingresso (inbound email).
- Usare template di tipo mailto per precompilare risposte email.

19.2 Creare Email Notifications

- Definire destinatari, condizioni di trigger e contenuto.
- Scegliere se la notifica è basata su record (record-based) o su evento.

19.3 Invio in Risposta a Cambiamenti di Record

- Trigger su modifiche di campo, stato o assegnazione.
- Configurare condizioni e script condizionali se necessario.

19.4 Invio in Risposta a Eventi

- Collegare notifiche ad eventi di sistema (es. `incident.closed`).
- Usare eventi per decoupling e riusabilità.

19.5 Debug delle Notifiche

- Controllare log delle email e system mailboxes.
- Verificare se le notifiche sono state inviate, ricevute o skipgate.
- Usare strumenti di tracing per isolare problemi.

19.6 Email Notification Scripts

- Utilizzare JavaScript per generare contenuti dinamici.
- Manipolare destinatari, soggetto e corpo dell'email programmaticamente.

19.7 System Mailboxes

- Analizzare messaggi **sent**, **received** e **skipped**.
- Usare le mailbox di sistema per il troubleshooting e la verifica.

19.8 Rispondere alle Email in Ingresso

- Configurare inbound actions per creare/aggiornare record.
- Gestire parsing, riconoscimento mittente e sicurezza.

19.9 Template mailto per Risposte Precompilate

- Creare template che precompilano campi nelle risposte email.
- Utili per standardizzare le reply degli utenti e includere dati del record.

20 About This Learning Module

20.1 Versione del Modulo

Il contenuto del modulo è stato aggiornato per la release San Diego. La release Utah può presentare differenze nell'interfaccia o nelle funzionalità rispetto a quanto mostrato nei materiali.

20.2 Applicazioni Utilizzate nel Modulo

- L'applicazione *Employee Special Days* è utilizzata per dimostrazioni teoriche.
- L'applicazione *NeedIt* è quella che verrà effettivamente sviluppata negli esercizi.

20.3 Source Control

I file necessari dell'applicazione NeedIt sono forniti tramite source control per iniziare lo sviluppo con una base preconfigurata.

20.4 Struttura degli Esercizi

Gli esercizi sono indicati tramite:

- Icona dedicata nel pannello di navigazione.
- Icona e parola *Exercise* nella parte superiore della pagina.
- Presenza delle parole *Exercise* o *Challenge* nel titolo della pagina.

21 Outbound Email

21.1 Panoramica

Le notifiche inviano email in uscita verso uno o più destinatari in risposta ad attività specifiche all'interno di ServiceNow.

21.2 Trigger delle Notifiche

Le email possono essere inviate quando:

- Un record viene inserito o aggiornato.
- Viene generato un evento.
- L'attività *Notification* viene eseguita in un workflow.
- L'azione *Send Email* viene eseguita in un flow.

22 Creating Notifications

22.1 Creazione di un File di Notifica

La procedura per aggiungere un file in Studio è identica per qualsiasi tipologia:

1. Selezionare il link *Create Application File*.
2. Scegliere il tipo *Notification*.
3. Configurare il nuovo file.

22.2 Campi Principali della Notifica

- **Name:** Nome della notifica.
- **Table:** Tabella che genera il trigger.
- **Category:** CATEGORIA per raggruppare notifiche correlate.
- **Application:** Applicazione di appartenenza.
- **Active:** Abilitazione della notifica nell'ambiente runtime.
- **Allow Digest:** Abilita la sezione per la configurazione del digest.

22.3 Default View e Advanced View

La *Default view* mostra solo una parte dei campi disponibili. Per accedere a tutti i campi, utilizzare il Related Link *Advanced view*.

22.4 Sezioni della Notifica

Una notifica è configurata tramite tre sezioni principali:

- **When to send:** Condizioni di invio.
- **Who will receive:** Destinatari dell'email.
- **What it will contain:** Contenuto del messaggio.

23 When to Send

23.1 Panoramica

La sezione **When to Send** definisce le condizioni necessarie per inviare una notifica. I campi disponibili dipendono dal tipo di trigger: record inserito/aggiornato, evento generato o notifiche triggerate manualmente.

23.2 Trigger Disponibili

- **Record inserted or updated:** la notifica scatta quando un record viene creato o aggiornato.
- **Event is fired:** la notifica scatta quando viene generato un evento specifico.
- **Triggered:** la notifica viene richiamata manualmente da script.

23.3 Opzioni per Record Inserted or Updated

- **Inserted:** invia email quando un nuovo record è aggiunto al database.
- **Updated:** invia email quando un record esistente viene aggiornato.
- **Conditions:** condizioni semplici per inviare la notifica.
- **Advanced Condition:** script server-side per condizioni complesse. Variabili disponibili:
 - **current:** il record target
 - **answer:** true/false per decidere se inviare la notifica

23.4 Opzioni per Event is Fired

- **Event name:** evento che triggerà la notifica.
- **Advanced Condition:** script con accesso a **current** e **event**.

23.5 Weight

- Determina quale notifica inviare se più notifiche sono triggerate per lo stesso record e destinatario.
- **Weight = 0:** notifiche sempre inviate.
- **Weight > 0:** viene inviata solo la notifica con il valore più alto.
- Se più notifiche hanno lo stesso Weight massimo, tutte vengono inviate.

23.6 Conditions vs Advanced Condition

Entrambi i campi devono restituire `true` affinché la notifica sia inviata.

- **Conditions:** builder grafico.
- **Advanced Condition:** script server-side per logiche più complesse (es. controllare date, anniversari, ecc.).

24 User Notification Settings

24.1 Panoramica

Aggiungere un utente nella sezione *Who will receive* non garantisce che l'utente riceverà la notifica. La ricezione dipende dalle impostazioni nel record utente.

24.2 Impostazioni del Record Utente

- **Email:** indirizzo email dell'utente.
- **Calendar integration:** nessuno o Outlook; con Outlook l'utente riceve notifiche di appuntamenti.

24.3 Gestione delle Notifiche da Parte dell'Utente

Gli utenti possono usare:

- **Show Notifications (campanella)** per aprire le impostazioni.
- **Settings (ingranaggio)** per configurare:
 - Abilitare/disabilitare notifiche
 - Aggiungere nuovi canali (email, mobile push, SMS, voice)
 - Disabilitare specifiche notifiche
 - Iscriversi a notifiche *subscribable*
 - Applicare schedule alle notifiche

24.4 Force Delivery per Sviluppatori

Modificando la form della notifica per includere il campo *Force delivery*, le notifiche possono essere inviate anche se l'utente ha disattivato la ricezione.

24.5 Creazione e Configurazione dei Canali

- **Name:** nome del device
- **Type:** Email, Instant Message, SMS, Voice
- **Email address / Phone number / Service provider:** a seconda del tipo di canale

24.6 Gestione Personale delle Notifiche

Gli utenti possono:

- Aggiungere o rimuovere se stessi da una notifica
- Applicare schedule o filtri a un canale
- Creare *Personal Notification* per iscriversi a notifiche non incluse nella loro lista *Who will receive*

25 Who Will Receive

25.1 Panoramica

La sezione **Who Will Receive** definisce i destinatari di una notifica. I destinatari possono essere statici (sempre gli stessi) o dinamici (determinati dai campi del record).

25.2 Destinatari Statici

- **Users:** selezionare utenti di ServiceNow, campi di riferimento utente, o email valide.
- **Groups:** tutti i membri del gruppo ricevono la notifica.

25.3 Destinatari Dinamici

- **Users/Groups in fields:** campi del record determinano dinamicamente i destinatari.

25.4 Opzioni Avanzate

- **Exclude delegates:** invia solo al destinatario principale.
- **Event parm1 / parm2:** lista dei destinatari passata come parametro durante un evento.
- **Send to event creator:** invia anche all'utente che ha scatenato l'evento.
- **Subscribable:** permette a utenti non destinatari di iscriversi alla notifica.

25.5 Consigli per Sviluppatori

Quando si inviano notifiche contenenti dati sensibili, limitare la lista dei destinatari ai soli utenti e gruppi autorizzati. Evita di abilitare l'opzione *Subscribable* per notifiche riservate.

26 What It Will Contain

26.1 Panoramica

Questa sezione definisce il contenuto della notifica e le informazioni dell'email, inclusi soggetto, testo e allegati. Il contenuto può essere statico o dinamico e può includere script lato server.

26.2 Opzioni Principali

- **Content type:** HTML e testo, solo HTML, solo testo.
- **Include attachments:** invia anche gli allegati del record.
- **Omit watermark:** rimuove il watermark (non raccomandato).
- **Message text / Message HTML:** testo della notifica; HTML supporta script server-side.
- **Push Message Only:** invio solo su mobile.
- **Importance:** Low, Normal, High.
- **From / Reply to:** indirizzi email del mittente e risposta.
- **Email template:** selezione template predefinito.
- **SMS alternate:** testo per SMS (max 140 caratteri).

26.3 Contenuto Dinamico

- Utilizzo di \$field_name per campi del record trigger.
- Dot-walking per campi correlati: \$related_field.field_name.
- Eventi: \$event.parm1, \$event.parm2.
- **Select variables:** aiuta a generare la sintassi dinamica.

26.4 Link e URI

- **\$URI:** link al record corrente, testo "LINK".
- **\$URI_REF:** link al record corrente, testo valore display.
- <reference_field>.URI / .URI_REF: link tramite dot-walking.
- **NOTIF_UNSUB:** link per disiscrizione dalla notifica.
- **NOTIF_PREFS:** link alle preferenze utente.
- Agent Workspace: link personalizzati tramite Notification Email Script.

26.5 Consigli per Sviluppatori

La dimensione massima della notifica, inclusi allegati, è 26 MB. Il MIME encoding aumenta la dimensione degli allegati di circa 37%.

27 Previewing Notifications

27.1 Panoramica

Il **Preview Notification** permette di testare una notifica senza inviarla realmente. Mostra come appariranno soggetto, corpo del messaggio e variabili dinamiche risolte.

27.2 Procedura

1. Cliccare il pulsante *Preview Notification*.
2. Selezionare l'**Event Creator** desiderato per la simulazione.
3. Selezionare il **Preview Record** su cui testare la notifica.

27.3 Dettagli Importanti

- Utenti sbarrati nella lista non riceveranno la notifica (notifiche disattivate o Send to event creator disabilitato).
- Il preview mostra il soggetto e il corpo della notifica con tutte le variabili dinamiche risolte.
- Tutti i link inclusi nella notifica sono attivi nel preview.

27.4 Consiglio per Sviluppatori

Usare il preview per verificare testo, HTML, link e variabili prima di inviare notifiche reali agli utenti.

28 Outbound Email Configuration

28.1 Panoramica

La configurazione dell'**outbound email** permette l'invio di email dalle notifiche. Tipicamente è responsabilità dell'amministratore di sistema, non dello sviluppatore.

28.2 Passaggi per Abilitare l'Email

1. In un Personal Developer Instance, l'invio email è disabilitato di default.
2. Aprire *All menu → System Diagnostics → Email Diagnostics* per verificare lo stato.
3. Abilitare l'invio email solo nel Global scope:
 - *All menu → System Properties → Email Properties*
 - Selezionare **Email sending enabled**.
4. Testare le email:
 - In un'istanza non-prod, inviare tutte le email a un indirizzo di test.
 - Per singole notifiche, aggiungere un indirizzo di test nella notifica.

- Rimuovere gli indirizzi di test al termine del test.
5. Salvare le modifiche e aprire *Email Diagnostics* per confermare che l'invio è operativo.
 6. Tornare allo scope dell'applicazione per continuare lo sviluppo.

28.3 Consiglio per Sviluppatori

L'invio email deve essere abilitato nel **Global scope** affinché tutte le notifiche funzionino correttamente.

29 Testing Notifications

29.1 Panoramica

Il testing delle notifiche prevede sia la **preview** sia la verifica dell'invio reale delle email. Queste fasi assicurano che le email vengano generate correttamente e raggiungano i destinatari previsti.

29.2 Preview Notification

- Verifica che i destinatari siano corretti e non barrati.
- Controlla che il **Subject** e il **Message HTML** risolvano correttamente le variabili dinamiche.

29.3 Generazione dell'Email Reale

1. Creare la condizione che triggerà la notifica (es. Scheduled Script Execution).
2. Verificare le email generate in *All menu → System Logs → Emails*.
3. Attendere fino a 60 secondi per l'invio effettivo.

29.4 Verifica Outbound Emails

- Aprire *All menu → System Mailboxes → Outbound*.
- Controllare i moduli:
 - **Outbox**: messaggi in attesa di invio.
 - **Sent**: messaggi inviati.
 - **Skipped**: messaggi non inviati a causa del campo **Weight**.
 - **Failed**: messaggi non inviati da ServiceNow.

29.5 Test Email e Mittente

- Assicurarsi che gli indirizzi di test ricevano l'email.
- Per cambiare il mittente, modificare il campo **Email user label** in *System Mailboxes → Administration → Email Accounts*.
- Tipicamente questa operazione è riservata all'amministratore.

30 Notification Email Scripts

30.1 Panoramica

I **Notification Email Scripts** permettono di generare dinamicamente il contenuto delle notifiche email tramite codice server-side. Questi script sono file applicativi creati tramite il menu principale di ServiceNow, non in Studio.

30.2 Creazione dello Script

1. Navigare in *System Notification* → *Email* → *Notification Email Scripts*.
2. Cliccare **New** per creare un nuovo script.
3. Configurare i campi principali:
 - **Name:** nome dello script (senza spazi o caratteri speciali eccetto underscore).
 - **Newlines to HTML:** gestione dei ritorni a capo.

30.3 Oggetti e Metodi Disponibili

Gli script possono accedere a:

- **current:** record associato alla notifica (GlideRecord API)
- **email:** gestione invio email, CC, BCC, subject (GlideEmailOutbound API)
- **template:** accesso al template (TemplatePrinter API)
- **event:** disponibile solo per notifiche basate su eventi

30.4 Override dei Campi Notifica

- Valori impostati via script sovrascrivono quelli della notifica standard.
- Per aggiungere CC o BCC usare `email.addAddress()`.

30.5 Inclusione nello HTML della Notifica

- Inserire lo script in **Message HTML** con la sintassi: `${mail_script:scriptName}`.
- Esempi di uso pratico: aggiungere manager come CC, conteggio dinamico di record.

30.6 Consigli per Sviluppatori

- Nei PDI, i campi Manager dei record utente potrebbero essere vuoti; aggiungere manualmente valori per testare gli script.
- Utilizzare Notification Email Scripts per logica dinamica complessa non gestibile direttamente dai campi della notifica.

31 Inbound Email

31.1 Panoramica

Le **Inbound Email** sono email ricevute da qualsiasi utente o indirizzo in ServiceNow. Quando un'email viene ricevuta, il sistema deve determinare se intraprendere azioni in risposta.

31.2 Determinazioni del Sistema

Per ogni email ricevuta, ServiceNow valuta:

- Se il messaggio è una **risposta**, un **inoltro** o un **nuovo messaggio**.
- Se esiste un **record esistente** associato al messaggio.
- Se il mittente è un **utente noto**.

31.3 Azione su Email Inbound

ServiceNow può:

- Aggiornare record esistenti.
- Creare nuovi record.
- Ignorare messaggi non rilevanti o da mittenti sconosciuti.

32 Identifying the Sender

32.1 Panoramica

Quando ServiceNow riceve un'email inbound, il mittente viene parsato nella proprietà `email.from` e il sistema tenta di associare l'indirizzo a un record nella tabella **User [sys_user]**.

32.2 Azioni in caso di corrispondenza

Se viene trovato un record corrispondente:

- ServiceNow impersona l'utente per eseguire le azioni relative all'email.
- Gli script hanno accesso alle informazioni dell'utente tramite i metodi `GlideSystem`, come ad esempio `gs.getUserName()`.

32.3 Azioni in assenza di corrispondenza

Se non viene trovato alcun record:

- Se la creazione automatica degli utenti (*automatic User creation*) è abilitata, ServiceNow crea automaticamente l'utente e lo impersona.
- Se la creazione automatica degli utenti è disabilitata, ServiceNow impersona l'utente **Guest**.

33 Inbound Email Actions

33.1 Panoramica

Le **Inbound Email Actions** definiscono quali azioni ServiceNow esegue in risposta a un'email in ingresso. Queste azioni determinano:

- Se il messaggio è una risposta (*reply*) o un inoltro (*forward*).
- Se il messaggio corrisponde a un record esistente nel database.
- Se il messaggio soddisfa le condizioni definite.

33.2 Determinare il tipo di messaggio

Il primo passo è identificare se l'email è una risposta, un inoltro o un nuovo messaggio. ServiceNow analizza la riga dell'oggetto dell'email in arrivo per prefissi come **Re:** o **Fwd:**. I prefissi possono essere configurati in **System Mailboxes > Administration > Email Properties > Inbound Email Configuration** come elenco separato da virgole e case-insensitive.

33.3 Corrispondenza con un record

Se il messaggio è riconosciuto come risposta o inoltro, ServiceNow tenta di abbinare l'email a un record esistente cercando:

- Un **watermark** unico presente nell'oggetto o nel corpo del messaggio (di solito l'ultima riga dell'email inviata).
- Se non viene trovato alcun watermark, il sistema cerca il numero del record nel soggetto o nel corpo del messaggio.

33.4 Esecuzione delle azioni inbound

Per determinare quale azione inbound eseguire, ServiceNow verifica che:

- Il tipo di email in ingresso corrisponda al tipo di azione inbound.
- Il watermark o il numero del record corrisponda a un record sulla tabella target.
- Le condizioni dell'inbound action siano valutate come **true**.

34 Creating Inbound Email Actions

34.1 Panoramica

Le **Inbound Email Actions** permettono di creare o aggiornare record nel database o inviare risposte automatiche in seguito alla ricezione di un'email. Il processo di creazione è simile a quello di altri file applicativi in Studio:

1. Cliccare su *Create Application File*.
2. Scegliere il tipo **Inbound Email Actions**.
3. Configurare il nuovo file.

34.2 Configurazione dell’Inbound Email Action

I principali campi di configurazione includono:

- **Name:** Nome dell’Inbound Action.
- **Target table:** La tabella in cui i record saranno creati o aggiornati.
- **Action type:** *Record Action* per modificare un record, *Reply Email* per inviare un'email.
- **Active:** Se selezionato, abilita l’Inbound Action.
- **Stop processing:** Interrompe l’esecuzione delle azioni successive dopo questa inbound action.

34.3 Sezione *When to Run*

Questa sezione definisce i criteri per l’esecuzione dell’Inbound Action:

- **Type:** *New, Reply, Forward.*
- **Execution Order:** Ordine di esecuzione tra le inbound actions. I numeri più bassi vengono eseguiti per primi.
- **Required roles:** Ruoli richiesti al mittente per attivare l’azione.
- **From:** Specifica l’utente richiesto. Lasciare vuoto per tutti gli utenti.
- **Conditions:** Condizioni definite tramite Condition Builder o script lato server.

Nota: Se *Conditions* o *Condition* non hanno valore, la condizione restituisce `true`. Entrambe devono restituire `true` affinché l’azione venga eseguita.

34.4 Sezione *Actions*

Definisce le azioni da eseguire sui record creati o aggiornati:

- **Field actions:** Aggiornare i campi con valori hardcoded o derivati dall'email.
- **Script:** Script lato server con accesso a:
 - `current`: record corrente della tabella target.
 - `event`: contiene `parm1` (record number) e `parm2` (username).
 - `email`: oggetto email parsato (`email.subject`, `email.from`, ecc.).
 - `logger`: utilizzare `logger.log()` per scrivere nel system log.

Ricordarsi di chiamare `current.update()` per salvare le modifiche nel database.

34.5 Sezione *Description*

Inserire una descrizione dettagliata della finalità dell’Inbound Action. **Esempio:** “Replies to birthday greetings are unexpected. This Inbound Action creates an Incident record to follow up with any replies received from a birthday greeting.”

35 Auto Create Reply Text

35.1 Panoramica

La funzionalità **Auto Create Reply Text** consente di aggiungere link cliccabili nelle notifiche email che pre-compilano automaticamente il soggetto della risposta. Questo permette di semplificare le risposte standardizzate degli utenti e automatizzare processi come la correzione di dati o la conferma di informazioni.

35.2 Flusso Operativo

Il flusso tipico prevede i seguenti passaggi:

1. Creare un **Email Template** di tipo mailto:

- Nome del template: `mailto.<nome>` (senza spazi, minuscolo)
- Layout dell'email
- Tabella di riferimento
- Soggetto precompilato
- Corpo del messaggio HTML

2. Includere il template nella notifica usando la sintassi:

```
 ${mailto:<emailtemplatename>}
```

3. Quando l'utente clicca sul link nell'email:

- Il soggetto della risposta è automaticamente popolato
- L'utente può aggiungere testo nel corpo del messaggio

4. La risposta inviata può attivare una **Inbound Email Action** per eseguire azioni automatiche, come creare un incidente per correggere dati errati.

35.3 Vantaggi

- Migliora l'esperienza utente riducendo errori manuali
- Standardizza le risposte
- Automatizza processi ripetitivi senza intervento manuale

36 Notifications Module Recap

36.1 Concetti Chiave

- Le **Email Notifications** inviano email in uscita.
- Possono rispondere a **inserimenti o aggiornamenti di record**.
- Possono rispondere a **eventi**.
- I destinatari possono essere **dinamici** o **hard coded**.
- Il contenuto del messaggio può essere **dinamico** o **hard coded**.
- ServiceNow analizza i messaggi in entrata nell'oggetto **email**.
- Il mittente viene identificato confrontando la proprietà **email.from** con il campo **email** della tabella **Users [sys_user]**.

36.2 Scripting e Reuse

- Usare i metodi `GlideSystem` come `gs.getUserName()` per ottenere informazioni sull'utente.
- I **Notification Email Scripts** sono script riutilizzabili lato server che hanno accesso a:
 - `current`
 - `email`
 - `template`
 - `event`
- Per includere uno script in un messaggio, usare la sintassi:
 `${mail_script:scriptName}`

36.3 Inbound Email Actions

- Elaborano le email in entrata.
- Possono rispondere a risposte (*replies*) e inoltri (*forwards*).
- Possono anche rispondere a messaggi nuovi.
- Gli script delle Inbound Email Actions hanno accesso a:
 - `current`
 - `event`
 - `email`
 - `logger`

36.4 Strumenti e Template

- Usare i moduli **System Mailboxes** per verificare se le email sono inviate, ricevute o saltate.
- Creare **mailto Email Templates** per pre-compilare le risposte.

37 About This Learning Module

37.1 Versione del Contenuto

Il contenuto in questo modulo è aggiornato all'**San Diego ServiceNow release**. Se si utilizza la release Utah, alcune funzionalità potrebbero differire leggermente.

37.2 Esercizi Pratici

Gli esercizi nel modulo sono indicati in tre modi:

- Icona “Exercise” nella barra di navigazione.
- Icona “Exercise” e parola “Exercise” in cima alla pagina.
- Parole “Exercise” o “Challenge” nel titolo della pagina.

Non è necessario ricreare tutti gli esempi teorici presenti nel modulo.

37.3 Applicazione NeedIt

L'applicazione **NeedIt** permette agli utenti di richiedere servizi da diversi dipartimenti. Per gli esercizi hands-on:

- Inizierai con tutti i file necessari disponibili tramite **source control**.
- Questo permette di concentrarsi sul coding e sul testing senza creare i file da zero.