

Primi passi con Wisej.NET

Da un Visual Studio vuoto a un'app in esecuzione — e le abitudini che mantengono pulito il codice.

● LearnWisej



Obiettivo della lezione

Crea un primo progetto Wisej.NET, aprilo nel Designer, esegui in locale, ispeziona la struttura della soluzione e scopri dove si trovano i file di avvio e di configurazione.

1. Parti dal template di progetto

In Visual Studio, crea un nuovo progetto a partire dai template Wisej.NET installati. L'elenco esatto cambia a seconda della versione, ma il flusso è sempre lo stesso: **crea un nuovo progetto** → **cerca "Wisej.NET"** → **scegli un template di applicazione web Wisej.NET** → **dagli un nome** → **crea**.

Nome consigliato per il progetto di esercitazione

WisejTrainingApp



Prima di iniziare

Assicurati che l'estensione Wisej.NET per Visual Studio e i template siano installati. Se i template non compaiono, ricontrolla l'installazione o riavvia Visual Studio.

2. Apri il Designer di Wisej.NET

Il Designer è il posto in cui posizioni i controlli in modo visuale. Apri la pagina o la finestra principale e passa alla vista di progettazione, poi trascina i controlli dalla Toolbox, imposta le proprietà e fai doppio clic su un controllo per creare un gestore di eventi.

Attività nel Designer	Che cosa fai
Aggiungere un controllo	Trascina sulla superficie un <code>Button</code> , una <code>Label</code> , una <code>TextBox</code> , un <code>Panel</code> o una <code>DataGridView</code> .
Impostare il testo	Seleziona il controllo, apri Properties e cambia la proprietà <code>Text</code> .
Rinominare un controllo	Cambia <code>Name</code> in qualcosa di chiaro come <code>btnRun</code> , <code>lblStatus</code> , <code>txtCustomerName</code> .
Creare il codice del click	Fai doppio clic su un pulsante (o usa il pannello Events) per creare un gestore <code>Click</code> .
Regolare il layout	Usa <code>Dock</code> , <code>Anchor</code> , <code>Size</code> , <code>Location</code> , <code>Margin</code> , <code>Padding</code> .

3. Esegui in locale

Eseguire in locale significa provare sul tuo computer prima di pubblicare. Usa il normale pulsante Start / Debug — il progetto viene compilato, si avvia un server locale e l'app si apre nel browser.

1. Fai clic su **Start / Debug** in Visual Studio.
2. Attendi che il progetto venga compilato.
3. Verifica che il browser apra l'applicazione Wisej.NET.
4. Prova il click di un pulsante per verificare che il tuo codice venga eseguito.
5. Usa i breakpoint per ispezionare gestori di eventi e variabili.



Abitudine di diagnostica

Se l'app non si apre, leggi prima le finestre **Error List** e **Output** di Visual Studio. La maggior parte dei problemi di chi inizia sono pacchetti mancanti, errori di compilazione, il progetto di avvio sbagliato o un file rinominato i cui riferimenti non sono stati aggiornati.

4. Ispeziona la struttura della soluzione

Il Solution Explorer non è un insieme di file a caso — mostra come è organizzata l'app.

File o cartella	A che cosa serve
<code>Program.cs</code>	Codice di avvio che fa partire l'applicazione e spesso mostra la prima Page o Window.
<code>Page1.cs</code> / <code>Window1.cs</code>	Code-behind: dove scrivi i gestori di eventi e il comportamento della schermata.
<code>*.Designer.cs</code>	Layout e inizializzazione dei controlli generati dal Designer. Non modificarlo a mano.
<code>*.resx</code>	File di risorse per il Designer e per i dati localizzati.
File di configurazione	<code>Web.config</code> più la configurazione JSON specifica di Wisej.NET.
<code>Models/</code>	Semplici classi di dati come <code>Ticket.cs</code> o <code>Customer.cs</code> .
<code>Services/</code>	Logica di business come <code>TicketService.cs</code> .

5. Buone abitudini da sviluppatore

Abitudine	Da evitare	Da preferire
Dare nomi chiari ai controlli	<code>button1</code> , <code>label2</code>	<code>btnSaveTicket</code> , <code>lblStatus</code>
Mantenere leggibili gli eventi	200 righe in <code>btnSave_Click</code>	Chiamare <code>ValidateInput()</code> e <code>ticketService.Save(ticket)</code>
Separare la logica di business	Tutta la logica di salvataggio nella pagina	Un <code>TicketService</code> che governa le regole
Organizzare i file	Tutto nella radice del progetto	Cartelle Models, Services, Pages, Controls

6. Esempio: un gestore di eventi leggibile

Un gestore di eventi dovrebbe leggersi come un breve racconto — prendi l'input, chiama la logica, aggiorna l'interfaccia:

```
private void btnSayHello_Click(object sender, EventArgs e)
{
    string name = txtName.Text.Trim();

    if (string.IsNullOrEmpty(name))
    {
        lblStatus.Text = "Please enter a name.";
        return;
    }

    lblStatus.Text = $"Hello, {name}!";
}
```

Man mano che l'app cresce, sposta regole e operazioni sui dati in classi di servizio:

```
private readonly TicketService ticketService = new TicketService();

private void btnSaveTicket_Click(object sender, EventArgs e)
{
    Ticket ticket = ReadTicketFromScreen();
    ticketService.Save(ticket);
    lblStatus.Text = "Ticket saved.";
}
```



Sei pronto per il lab

Prossimo passo: guarda la videoguia di costruzione, supera la verifica delle conoscenze, poi crea ed esegui la tua prima schermata Wisej.NET nel lab pratico.