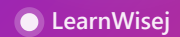


Theme-Bilder, URLs, SVG-Farbe und der Image Selector im Designer

Lektionsleitfaden · Lab- und Prüfungsleitfaden



Lektionsleitfaden

Bildquellen, die keine Dateien neben der Seite sind: benannte Theme-Bilder aus dem Theme-Cache des Clients, relative und absolute URLs, base64-Daten im Theme, der Image Selector in Visual Studio, der den Quell-String für Sie schreibt, und das Suffix ?color=, das ein geeignetes einfarbiges SVG mit dem Namen einer Theme-Farbe umfärbt. Was das für Entwicklerinnen und Entwickler bedeutet, die die Grundlagen von C# beherrschen und schon etwas mit den Steuerelementen und dem Designer von Wisej.NET gearbeitet haben – und wie Sie dieses Modul lesen.



Fokus: Produktivbetrieb

Bildquellen, die keine Dateien neben der Seite sind: benannte Theme-Bilder aus dem Theme-Cache des Clients, relative und absolute URLs, base64-Daten im Theme, der Image Selector in Visual Studio, der den Quell-String für Sie schreibt, und das Suffix ?color=, das ein geeignetes einfarbiges SVG mit dem Namen einer Theme-Farbe umfärbt.

1. Das Problem im Produktivbetrieb

IconDesk hat inzwischen ein Befehls-Panel und ein Bildlabor, und jedes Bild darin ist eine ausgelieferte Datei plus ein von Hand getippter Pfad. Das trägt, bis die Anwendung ein neues Theme bekommt, unter einem anderen virtuellen Verzeichnis ausgeliefert wird oder dasselbe Symbol auf einem Bildschirm in einer Warnfarbe und auf einem anderen neutral zeigen soll. Dieses Modul verlegt den Ursprung eines Bildes aus dem Projektordner in das Theme, in eine URL oder in den Designer, damit der String, den ein Steuerelement trägt, auch nach einem Wechsel von Grafik, Theme oder Server noch etwas bedeutet.

2. Zentrale Konzepte

Konzept	Warum es wichtig ist
benanntes Theme-Bild	Ein semantischer Name wie <code>icon-print</code> , den das aktive Theme auf eine Grafik abbildet, sodass ein anderes Theme ohne C#-Änderung ein anderes Bild liefern kann.
Theme-Cache des Clients	Theme-Bilder reisen mit dem Theme und liegen bereits im Browser, wenn ein Steuerelement eines anfordert – das macht ein Theme zum richtigen Ort für einen kleinen Satz von Befehlssymbolen und zum falschen für Tausende.
Quelle mit relativer URL	Ein Bildquellen-String, der gegen die Anwendung aufgelöst wird: die einfachste Form für ein Logo, das mit der Anwendung ausgeliefert wird.
Quelle mit absoluter URL	Eine vollständige externe Adresse, die das Bild zu einer Abhängigkeit von einem anderen System macht – mit Folgen für Verfügbarkeit, Datenschutz, CORS und Content Security Policy.
eingebettetes base64-Bild	Bildbytes, die in die Theme-Definition selbst geschrieben werden, sodass die Grafik keine zweite Anfrage braucht – zum Preis eines größeren Themes.
Image Selector	Der Auswahldialog zur Entwurfszeit in Visual Studio, der Projektbilder, Theme-Bilder und referenzierte Icon-Pack-Ressourcen durchsucht und den gewählten Quell-String dann in <code>Designer.cs</code> serialisiert.
Suffix <code>?color=</code>	Ein Suffix an einer Bildquelle, das Wisej.NET bittet, ein geeignetes einfarbiges SVG mit einer CSS-Farbe oder dem Namen einer Theme-Farbe umzufärben.

3. Der rote Faden der Lektion

Vier Formen, die ein Bild in einem Theme annehmen kann

Ein Wisej.NET-Theme besteht nicht nur aus Farben und Rändern; es trägt auch Bilddefinitionen, die das Framework und Ihre Anwendung gleichermaßen nutzen können. Die Dokumentation zum Theme Builder beschreibt vier nützliche Formen für diese Definitionen. Eine **relative URL** wird gegen die Anwendung aufgelöst. Eine **absolute URL** benennt eine vollständige externe Adresse. **Eingebettete base64**-Daten legen die Bildbytes in die Theme-Definition, sodass nichts weiter geholt werden muss. Ein **benanntes Bild** zeigt auf eine Bilddefinition und ergänzt damit eine Indirektionsebene: Viele Appearances können denselben Namen referenzieren, und eine einmalige Änderung der Definition ändert jede Verwendung.

Die Indirektion ist für IconDesk der interessante Teil. Wenn die Galerie nach einem Namen statt nach einer Datei fragt, gehört die Entscheidung, welche Grafik dieser Name bedeutet, dem Theme – und das Theme lässt sich beim Deployment oder pro Kunde austauschen. Fragt die Seite nach einer URL, gehört die Entscheidung dem, der die URL getippt hat, und ein Theme-Wechsel ändert nichts. Das Lab bringt beide Verhalten auf einen Bildschirm, damit der Unterschied keine Glaubensfrage bleibt.

Absolute URLs verdienen eine bewusste Entscheidung statt einer Gewohnheit. Ein von einem anderen Host geholtes Bild ist eine Laufzeitabhängigkeit: Es kann nicht verfügbar sein, es kann eine Authentifizierung verlangen, die der Browser nicht hat, es kann von einer Content Security Policy blockiert werden, und seine

Cache-Policy gehört nicht Ihnen. Ein Icon, das immer da sein muss, wird besser mit der Anwendung ausgeliefert oder als Ressource verpackt – und genau dorthin geht Modul 5.

Der Name eines Theme-Bilds ist ein Schlüssel, kein Pfad

Ein Theme-Bild zuzuweisen ist die kürzeste Zeile des Kurses: Setzen Sie eine Eigenschaft `ImageSource` auf den Namen, und Wisej.NET lädt es aus dem clientseitigen Theme-Cache. Auf dem Server öffnet nichts eine Datei und es entsteht kein `System.Drawing.Image` – das ist also die zweite Pipeline aus Modul 1, nicht die erste. Der String ist ein Theme-Schlüssel. Bildet ein anderes Theme diesen Schlüssel auf eine andere Grafik ab, ändert die Anwendung ihr Aussehen ohne eine einzige Änderung in C# – deshalb sind semantische Namen wie `icon-print`, `icon-save` oder `icon-warning` einem eingefügten Pfad überlegen.

Dieses Vorladen ist zugleich eine Regel für die Größe. Theme-Bilder kommen mit dem Theme, ein Theme eignet sich also für einen kleinen, stabilen Satz von Befehlssymbolen auf Framework-Ebene. Icon-Pack-Ressourcen verhalten sich umgekehrt: Jedes Icon wird einzeln angefordert und danach vom Browser gecacht, was bis zu sehr großen Katalogen skaliert, aber die erste Verwendung jedes Icons eine Anfrage kosten lässt. Zu wissen, welches von beiden Sie verwenden, sagt Ihnen, wie das Netzwerk-Panel aussehen sollte – und genau diesen Beleg sollen Sie in Modul 4 sammeln.

```
// IconGallery: drei Quell-Strings, drei verschiedene Auflösungswege.
buttonPrint.ImageSource = "icon-print";           // Theme-Schlüssel
logo.ImageSource        = "Images/company-logo.svg"; // relative URL
avatar.ImageSource      = "https://cdn.example.com/users/42.jpg"; // absolut

// Entscheidungen zur Laufzeit bleiben semantisch, nie aus Benutzertext zusammengesetzt.
buttonStatus.ImageSource = isDanger
    ? "icon-warning?color=error"
    : "icon-info?color=activeText";
```

Lassen Sie den Image Selector den Quell-String schreiben

Der Wisej.NET-Designer in Visual Studio bietet für Quell-Eigenschaften eine Bildauswahl an. Er durchsucht die Bilder im Projekt, die Bilder, die das aktuelle Theme definiert, und – sofern Icon-Pack-Assemblys referenziert sind – die automatisch gefundenen Packs. Der Icon-Pack-Explorer zeigt eine Liste oder ein Raster mit Vorschau, und ein einfarbiges Icon lässt sich vor der Übernahme in einer gewählten Farbe vorschauen. Wenn Sie eines auswählen, serialisiert der Designer eine konkrete Quelle wie eine `resource.wx/...`-URL in `Designer.cs`.

Das ist mehr als Tipphilfe. Die erzeugte Zeile zu lesen vermittelt die Darstellung zur Laufzeit: Ein Pack-Icon zu wählen kopiert das SVG **nicht** in das Formular. Das Formular speichert einen Quell-String, der auf die Pack-Assembly zurückverweist – deshalb liefern fünf Formulare mit demselben Icon weiterhin nur eine einzige Kopie der Grafik aus. Genau aus diesem Grund sollen Sie im Lab `Designer.cs` öffnen und diese Strings lesen.

- Projektbilder: mit dem Selector wählen, damit kein Pfad zweimal getippt wird.
- Theme-Bilder: die Theme-Liste mit Vorschau nutzen, damit Sie die Grafik des aktuellen Themes sehen.
- Icon-Pack-Ressourcen: suchen und vorschauen, statt Dateinamen auswendig zu lernen.

Ein einfarbiges SVG mit ?color= umfärben

Wisej.NET kann ein geeignetes einfarbiges SVG umfärben, indem Sie `?color=` an die Bildquelle anhängen. Der Wert kann eine CSS-artige Farbe wie `red` oder `#3078d4` sein oder der Name einer Theme-Farbe wie `error` oder `activeText`. Eine Theme-Farbe ist die stärkere Wahl, weil sie auch nach einem Theme-Wechsel sinnvoll bleibt: Ein fest verdrahtetes Dunkelgrau kann vor einem dunklen Theme verschwinden, während eine semantische Theme-Farbe darauf ausgelegt ist, sich mitzubewegen. Dasselbe Suffix funktioniert bei einem Theme-Bildnamen ebenso wie bei einer `resource.wx`-Ressourcenquelle, weil es auf den Quell-String angewendet wird und nicht auf einen bestimmten Speichermechanismus.

Umfärben funktioniert nicht überall, und das Scheitern ist leise. Die Dokumentation warnt, dass es fehlschlagen kann, wenn das SVG selbst Werte für `fill` oder `stroke` fest verdrahtet, denn diese Werte gewinnen. Bei einem Icon, das Wisej.NET einfärben soll, entfernen Sie die widersprüchlichen festen Farben, wo das angemessen ist; bei einem mehrfarbigen Logo lassen Sie es unangetastet. Die Galerie von IconDesk sollte je ein Beispiel enthalten, damit die Lab-Notiz sagen kann, welche Grafik das Suffix angenommen und welche es ignoriert hat. Halten Sie Quell-Strings zur Laufzeit semantisch und bauen Sie niemals einen Ressourcenpfad aus nicht vertrauenswürdigen Benutzertext zusammen.



Zuerst den Mechanismus benennen

Bevor Sie ein Bild zuweisen, sagen Sie, welchen Mechanismus Sie verwenden: ein `System.Drawing.Image`-Objekt, das der Server als PNG rendert, einen Bildquellen-String, den der Browser auflöst, ein benanntes Theme-Bild, eine über `resource.wx` eingebettete Ressource oder einen Icon-Font, der als HTML gezeichnet wird. Weisen Sie pro Steuerelement genau eine Bildeigenschaft zu, damit die anderen sie nicht still zurücksetzen, halten Sie den Quell-String aussagekräftig statt als eingefügten magischen Pfad, bevorzugen Sie SVG für Oberflächen-Icons, damit sie in jeder Größe und Skalierung scharf bleiben, und geben Sie jedem bedeutungstragenden Icon eine Textalternative.



Lernziele

Am Ende dieses Moduls können Sie:

- ✓ Relative URLs, absolute URLs, eingebettete base64-Daten und benannte Bilder in einem Wisej.NET-Theme verwenden.
- ✓ Namen von Theme-Bildern direkt an `ImageSource` zuweisen.
- ✓ Den Image Selector zur Entwurfszeit in Visual Studio nutzen, um lokale Assets, Theme-Assets und Icon-Pack-Assets auszuwählen.
- ✓ Geeignete SVG-Icons mit der `?color=`-Syntax von Wisej.NET und Theme-Farbnamen umfärben.

Lab- und Prüfungsleitfaden

Was Sie im praktischen Lab bauen, das vorgeschlagene Vorgehen und die geforderten Ergebnisse.



Praktisches Lab

Füllen Sie in IconDesk eine Seite IconGallery, indem Sie den Image Selector des Designers verwenden, statt Quell-Strings von Hand zu tippen. Weisen Sie fünf Steuerelementen Bilder zu: ein benanntes Theme-Bild, ein Projekt-SVG über eine relative URL, eine absolute URL und mindestens zwei Assets aus einem Icon-Pack, falls eines installiert ist. Färben Sie ein einfarbiges SVG mit dem Suffix `?color=` und dem Namen einer Theme-Farbe um, öffnen Sie die erzeugte Designer.cs und lesen Sie die Bildquellen-Strings, die der Selector geschrieben hat; schalten Sie die Anwendung dann zwischen zwei Wisej-Themes um und halten Sie fest, welche Assets sich geändert haben und welche exakt gleich geblieben sind.

Ausgangspunkt

Öffnen Sie die Projektmappe IconDesk in dem Zustand, in dem Modul 2 sie hinterlassen hat: das Befehls-Panel aus Modul 1 und die Seite ImageLab mit ihrem Size-Mode-Raster, ihrer `LoadAsync`-PictureBox und ihrem scharfen SVG. Fügen Sie eine neue Page namens `IconGallery` mit fünf leeren Steuerelementen und einer Theme-Auswahl hinzu, damit jedes Bild darauf in diesem Lab zugewiesen wird und nichts aus einem früheren Modul übernommen ist.

Vorgeschlagenes Vorgehen

- ✓ Platzieren Sie zuerst die fünf Galerie-Steuerelemente und geben Sie jedem ein Label, das den vorgesehenen Mechanismus benennt, bevor ein Bild zugewiesen ist.
- ✓ Weisen Sie jedes Bild über den Image Selector des Designers zu: die Theme-Liste für das benannte Theme-Bild, den Projektbaum für das relative SVG, das Adressfeld für die absolute URL und den Icon-Pack-Explorer für die Pack-Assets.
- ✓ Hängen Sie an ein einfarbiges SVG das Suffix `?color=` mit einem Theme-Farbnamen an und probieren Sie es auch an einem mehrfarbigen Asset, damit Sie sehen, wie das Suffix ignoriert wird.
- ✓ Öffnen Sie `Designer.cs`, kopieren Sie die erzeugten Bildquellen-Zeilen in die Lab-Notiz und markieren Sie jede als Theme-Schlüssel, relative URL, absolute URL oder Ressourcenquelle.
- ✓ Schalten Sie die Anwendung zwischen zwei Wisej-Themes um, halten Sie die Galerie in beiden fest und notieren Sie, welche Steuerelemente dem Theme gefolgt sind und welche nicht.

Geforderte Ergebnisse

- ✓ Seite IconGallery mit fünf Steuerelementen, deren Bilder über den Image Selector des Designers zugewiesen wurden
- ✓ Darunter ein benanntes Theme-Bild, ein relatives Projekt-SVG, eine absolute URL und mindestens zwei Icon-Pack-Assets
- ✓ Einfarbiges SVG, das über das Suffix `?color=` mit dem Namen einer Theme-Farbe umgefärbt wurde

- ✓ Auszug aus Designer.cs mit den Bildquellen-Strings, die der Selector erzeugt hat
- ✓ Notiz zum Theme-Wechsel, die festhält, welche Assets dem Theme gefolgt sind und welche unverändert blieben

Wiederholungsfragen

- ✓ Die Galerie erscheint auf Ihrem Rechner korrekt, zeigt aber auf dem einer Kollegin oder eines Kollegen nichts für die absolute URL. Welcher der fünf Mechanismen könnte diesen Unterschied erzeugen, und was würden Sie zuerst prüfen?
- ✓ Zwei Ihrer Icons ändern ihre Farbe nicht, obwohl das Suffix angehängt ist. Was steckt in diesen Dateien, und wann ist es die falsche Antwort, sie zu bearbeiten?
- ✓ Eine Designerin verlangt dasselbe Symbol in der Galerie, in einer Toolbar und in einer Grid-Zelle. Was kostet später am wenigsten Aufwand beim Ändern – der Theme-Schlüssel, die relative URL oder die Pack-Ressource – und warum?



Zuerst den Mechanismus benennen

Bevor Sie ein Bild zuweisen, sagen Sie, welchen Mechanismus Sie verwenden: ein `System.Drawing.Image`-Objekt, das der Server als PNG rendert, einen Bildquellen-String, den der Browser auflöst, ein benanntes Theme-Bild, eine über `resource.wx` eingebettete Ressource oder einen Icon-Font, der als HTML gezeichnet wird. Weisen Sie pro Steuerelement genau eine Bildeigenschaft zu, damit die anderen sie nicht still zurücksetzen, halten Sie den Quell-String aussagekräftig statt als eingefügten magischen Pfad, bevorzugen Sie SVG für Oberflächen-Icons, damit sie in jeder Größe und Skalierung scharf bleiben, und geben Sie jedem bedeutungstragenden Icon eine Textalternative.