

Xcode Interface Builder

Dies ist eine XML-Datei im XIB-Format (Xcode Interface Builder), die verwendet wird, um die Benutzeroberfläche für eine macOS-Anwendung zu definieren, die in Cocoa geschrieben ist. Diese XIB-Datei beschreibt speziell ein Einstellungsfenster für eine Anwendung namens "Reveal-In-GitHub." Lassen Sie mich das für Sie Schritt für Schritt aufschlüsseln:

Übersicht

Diese Datei definiert ein Fenster mit dem Titel "Reveal-In-GitHub Settings" mit verschiedenen UI-Elementen wie Schaltflächen, Textfeldern und einer benutzerdefinierten Ansicht. Es wurde mit Interface Builder (ein Teil von Xcode) gestaltet und verwendet Auto Layout zur dynamischen Positionierung von Elementen. Das Fenster wird von einer benutzerdefinierten Klasse namens `RIGSettingWindowController` gesteuert.

Wichtige Abschnitte

1. Header-Informationen

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.Cocoa.XIB" version="3.0" toolsVersion="9059" systemVersion="14F27" useAutolayout="YES" />
```

- Dies gibt an, dass es sich um eine XIB-Datei für eine macOS Cocoa-App handelt, die Version 3.0 des XIB-Formats verwendet.
- `toolsVersion="9059"` und `systemVersion="14F27"` geben die Xcode- und macOS-Versionen an, die bei der Erstellung verwendet wurden.
- `useAutolayout="YES"` bedeutet, dass das Layout sich dynamisch basierend auf der Fenstergröße oder Bildschirmauflösung anpasst.

2. Abhängigkeiten

```
<dependencies>
  <plugin identifier="com.apple.InterfaceBuilder.CocoaPlugin" version="9059"/>
</dependencies>
```

- Dies verweist auf das Cocoa-Plugin für Interface Builder, das den Rahmen für macOS-UI-Elemente bereitstellt.

3. Objekte

Dieser Abschnitt definiert alle Objekte in der Benutzeroberfläche, einschließlich des Fensters, Ansichten, Schaltflächen und deren Verbindungen.

- **Datei-Besitzer** (`RIGSettingWindowController`)

```

<customObject id="-2" userLabel="File's Owner" customClass="RIGSettingWindowController">
  <connections>
    <outlet property="configsView" destination="IKd-Ev-B9V" id="z2z-MF-G88"/>
    <outlet property="mainView" destination="se5-gp-Tj0" id="pGn-EC-mzi"/>
    <outlet property="window" destination="F0z-JX-Cv5" id="gIp-Ho-8D9"/>
  </connections>
</customObject>

```

- Der File's Owner ist die Controller-Klasse (RIGSettingWindowController), die dieses Fenster verwaltet.
- Es hat Ausgänge (Verbindungen) zur Hauptansicht, einer Konfigurationsansicht und dem Fenster selbst, die UI-Elemente mit dem Code verknüpfen.

• Fenster

```

<window title="Reveal-In-GitHub Settings" ... id="F0z-JX-Cv5" userLabel="Settings">
  <windowStyleMask key="styleMask" titled="YES" closable="YES" miniaturizable="YES" resizable="YES">
  <rect key="contentRect" x="527" y="176" width="651" height="497"/>
  <rect key="screenRect" x="0.0" y="0.0" width="1440" height="877"/>

```

- Definiert ein Fenster mit einem Titel, Schließen/Minimieren/Vergrößern-Schaltflächen und einer Größe von 651x497 Pixeln.
- Das contentRect gibt seine Position und Größe auf dem Bildschirm an, während screenRect die volle Bildschirmauflösung (1440x877) beschreibt.

• Inhaltsansicht

```

<view key="contentView" id="se5-gp-Tj0">
  <rect key="frame" x="0.0" y="0.0" width="651" height="497"/>
  <subviews>

```

- Die Hauptinhaltsansicht füllt das Fenster und enthält alle UI-Elemente (Subviews).

4. UI-Elemente (Subviews) Hier sind einige wichtige Elemente innerhalb der Inhaltsansicht:

• Textfelder

```

<textField ... id="G1C-Td-n9Y">
  <rect key="frame" x="18" y="425" width="187" height="17"/>
  <textFieldCell ... title="Custom Menu Items" ...>

```

- Ein Label mit der Aufschrift "Custom Menu Items" an der Position (18, 425).
- Weitere Textfelder umfassen "Menu Title," " ^ ↑ ⌘ +" (Control+Shift+Command-Tastenkombination) und "URL Pattern."

• Schaltflächen

```

<button ... id="EuN-9g-Vcg">
  <rect key="frame" x="14" y="13" width="137" height="32"/>

```

```

<buttonCell ... title="Save" ...>
<connections>
    <action selector="saveButtonClicked:" target="-2" id="efd-QE-4aQ"/>
</connections>

```

- Eine "Save"-Schaltfläche an (14, 13), die die Methode `saveButtonClicked:` im Controller auslöst.
- Weitere Schaltflächen umfassen "Clear Default Repos" und "Reset Default Menus," die jeweils mit bestimmten Aktionen verbunden sind.

- **Benutzerdefinierte Ansicht**

```

<customView ... id="IKd-Ev-B9V" userLabel="Config View">
    <rect key="frame" x="20" y="54" width="611" height="330"/>

```

- Ein Platzhalter für eine benutzerdefinierte Ansicht (wahrscheinlich programmgesteuert befüllt), die den größten Teil des Fensters einnimmt.

5. Verbindungen

- Die Schaltflächen sind mit Methoden in der `RIGSettingWindowController` verdrahtet (z. B. `saveButtonClicked:`, `clearButtonClicked:`), sodass Benutzerinteraktionen Code auslösen können.

Was macht das?

Diese XIB-Datei erstellt ein Einstellungsfenster für eine macOS-App namens "Reveal-In-GitHub," wahrscheinlich ein Tool zur Integration von GitHub-Funktionalitäten in eine andere Anwendung. Das Fenster ermöglicht es den Benutzern: - Menüelemente anzupassen (mit Titeln und URL-Mustern). - Ihre Einstellungen zu speichern. - Standard-Repositorys zu löschen oder Menüs auf Standardwerte zurückzusetzen.

Das Layout ist fest (`fixedFrame="YES"`) anstatt vollständig Auto Layout-Beschränkungen zu nutzen, was bedeutet, dass Positionen in Pixeln hartcodiert sind. Dies könnte es weniger flexibel für verschiedene Bildschirmgrößen machen, aber einfacher zu gestalten.

Warum XIB verwenden?

XIB-Dateien ermöglichen Entwicklern, Benutzeroberflächen visuell in Xcode's Interface Builder zu gestalten und sie über Ausgänge und Aktionen mit dem Code zu verbinden. Diese Datei verbindet die Benutzeroberfläche (Schaltflächen, Text usw.) mit der `RIGSettingWindowController`-Klasse, die die Logik handelt.