**1. Ajouter la variable globale du module dans votre fichier principal (TrinityAdmin.lua)**

Dans votre fichier **TrinityAdmin.lua**, là où vous créez déjà les modules existants, ajoutez une ligne pour créer le module ServerAdmin :

lua

Copier

local ServerAdminModule = TrinityAdmin:NewModule("ServerAdmin")

Ensuite, ajoutez la fonction globale qui sera appelée depuis le XML. Par exemple, juste après les autres fonctions globales :

lua

Copier

function TrinityAdmin\_ShowServerAdminPanel()

ServerAdminModule:ShowServerAdminPanel()

End

Ajouter les lignes pour cacher les boutons et montrer les boutons du menu principale

------------------------------------------------------------

-- Gestion du "menu principal" (les 4 boutons désormais)

------------------------------------------------------------

function TrinityAdmin:ShowMainMenu()

if TeleportModule.panel then TeleportModule.panel:Hide() end

if GMModule.panel then GMModule.panel:Hide() end

if NPCModule.panel then NPCModule.panel:Hide() end

if AccountModule.panel then AccountModule.panel:Hide() end

if GMFunctionsModule.panel then GMFunctionsModule.panel:Hide() end

if DatabaseModule.panel then DatabaseModule.panel:Hide() end

if ServerAdminModule.panel then ServerAdminModule.panel:Hide() end

TrinityAdminMainFrameTeleportButton:Show()

TrinityAdminMainFrameGMButton:Show()

TrinityAdminMainFrameNPCButton:Show()

TrinityAdminMainFrameGMFunctionsButton:Show()

TrinityAdminMainFrameAccountButton:Show()

TrinityAdminMainFrameDatabaseButton:Show()

TrinityAdminMainFrameServerAdminButton:Show()

TrinityAdminMainFrame:Show()

end

------------------------------------------------------------

-- Fonction pour cacher le menu principale quand on est dans les sous menus

------------------------------------------------------------

function TrinityAdmin:HideMainMenu()

TrinityAdminMainFrameTeleportButton:Hide()

TrinityAdminMainFrameGMButton:Hide()

TrinityAdminMainFrameNPCButton:Hide()

TrinityAdminMainFrameGMFunctionsButton:Hide()

TrinityAdminMainFrameAccountButton:Hide()

TrinityAdminMainFrameDatabaseButton:Hide()

TrinityAdminMainFrameServerAdminButton:Hide()

TrinityAdminMainFrame:Show()

end

**2. Créer le fichier du module (ServerAdmin.lua)**

Créez un nouveau fichier nommé **ServerAdmin.lua** dans votre dossier d'addon. Ce fichier contiendra le code spécifique à votre module ServerAdmin. Voici un exemple de module vide :

local ServerAdmin = TrinityAdmin:GetModule("ServerAdmin")

-- Fonction pour afficher le panneau ServerAdmin

function ServerAdmin:ShowServerAdminPanel()

TrinityAdmin:HideMainMenu()

if not self.panel then

self:CreateServerAdminPanel()

end

self.panel:Show()

end

-- Fonction pour créer le panneau ServerAdmin

function ServerAdmin:CreateServerAdminPanel()

local panel = CreateFrame("Frame", "TrinityAdminServerAdminPanel", TrinityAdminMainFrame)

panel:ClearAllPoints()

panel:SetPoint("TOPLEFT", TrinityAdminMainFrame, "TOPLEFT", 10, -50)

panel:SetPoint("BOTTOMRIGHT", TrinityAdminMainFrame, "BOTTOMRIGHT", -10, 10)

local bg = panel:CreateTexture(nil, "BACKGROUND")

bg:SetAllPoints(true)

bg:SetColorTexture(0.2, 0.2, 0.5, 0.7) -- Fond sombre, modifiez selon vos besoins

panel.title = panel:CreateFontString(nil, "OVERLAY", "GameFontNormalLarge")

panel.title:SetPoint("TOPLEFT", 10, -10)

panel.title:SetText("Server Admin Panel") -- Vous pouvez utiliser TrinityAdmin\_Translations si nécessaire

-- Ici, vous pouvez ajouter d'autres éléments (boutons, champs de saisie, etc.) pour la gestion du serveur

local btnBack = CreateFrame("Button", "TrinityAdminTeleportBackButton", panel, "UIPanelButtonTemplate")

btnBack:SetPoint("BOTTOM", 0, 10)

btnBack:SetText(TrinityAdmin\_Translations["Back"])

btnBack:SetHeight(22)

btnBack:SetWidth(btnBack:GetTextWidth() + 20)

btnBack:SetScript("OnClick", function()

panel:Hide()

TrinityAdmin:ShowMainMenu()

end)

self.panel = panel

end

Ce module est vide à l'exception de la création d'un panneau avec un fond sombre et un titre. Vous pourrez ensuite y ajouter vos fonctionnalités spécifiques pour la gestion du serveur.

**3. Ajouter le bouton dans le fichier XML**

Ouvrez votre fichier **TrinityAdmin.xml** et ajoutez un nouveau bouton pour lancer le panneau ServerAdmin. Par exemple, vous pouvez l’ajouter à côté des autres boutons déjà présents :

xml

<!-- Bouton Server Admin -->

<Button name="$parentServerAdminButton" inherits="UIPanelButtonTemplate" text="Server Admin">

<Size>

<AbsDimension x="100" y="22"/>

</Size>

<Anchors>

<Anchor point="LEFT" relativeTo="$parentDatabaseButton" relativePoint="RIGHT">

<Offset>

<AbsDimension x="10" y="0"/>

</Offset>

</Anchor>

</Anchors>

<Scripts>

<OnClick>

TrinityAdmin\_ShowServerAdminPanel()

</OnClick>

</Scripts>

</Button>

**4. Mettre à jour le fichier .toc**

N'oubliez pas d'ajouter le nouveau fichier **ServerAdmin.lua** dans votre fichier .toc pour qu'il soit chargé :

plaintext

Copier

# Code principal

TrinityAdmin.lua

TeleportPanel.lua

GMPanel.lua

NPCPanel.lua

AccountPanel.lua

GMFunctionsPAnel.lua

DatabasePanel.lua

ServerAdmin.lua

**5. Tester votre module**

Lancez World of Warcraft (votre serveur privé) et vérifiez que le nouveau bouton "Server Admin" apparaît dans le menu principal. Cliquez dessus pour ouvrir le panneau. Vous devriez voir un panneau avec le titre "Server Admin Panel".

**Résumé**

* **Dans TrinityAdmin.lua :**
  + Ajoutez local ServerAdminModule = TrinityAdmin:NewModule("ServerAdmin")
  + Ajoutez la fonction globale TrinityAdmin\_ShowServerAdminPanel()
* **Créez ServerAdmin.lua :**
  + Définissez le module avec des fonctions : ShowServerAdminPanel() et CreateServerAdminPanel()
* **Dans le XML :**
  + Ajoutez un bouton qui appelle TrinityAdmin\_ShowServerAdminPanel()
* **Dans le .toc :**
  + Listez ServerAdmin.lua pour le chargement.

Avec ces étapes, vous disposerez d’un module vide ServerAdmin que vous pourrez compléter par la suite.