

# VRoid SDK Manual

## About the included package

VRoid SDK includes UniVRM. UniVRM is imported automatically when VRoid SDK is imported.

## Use CharacterModelExample



By using VRoidHubController in

`Assets/VRoidSDK/Examples/CharacterModelExample/Prefabs`, you can import a new scene with the same process.

- Create a new application from the [application management page](#)
- Download `credential.json.bytes` from the created application's page

### Actions

Show application page

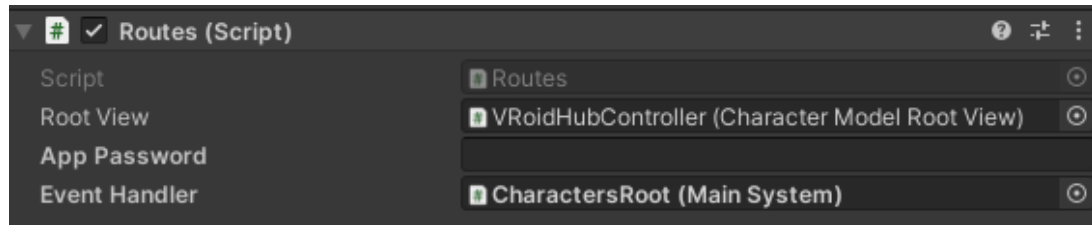
Edit

Destroy

Create credential file

- Place the downloaded `credential.json.bytes` in the following `Assets/Resources` location
- Open Example of `Assets/VRoidSDK/Examples/CharacterModelExample/Scenes`
- Set arbitrary password to `AppPassword` in `VRoidHubController/Routes` within Hierarchy

- The password set here will be used for model encryption



- When a scene is played, Hub connection flow can be checked

## Method without using Example or Prefab

### Login

```
class Login : MonoBehaviour
{
    void Start()
    {
        // ThreadContext used for communication
        var context = SynchronizationContext.Current;
        // Load the downloaded credential.json.bytes file
        var credential = Resources.Load<TextAsset>("credential.json");
        // App information used for configuration
        var credentialJson = credential.text;
        // Create a Config to be used for authorization
        var config = OAuthProvider.CreateSdkConfig(credentialJson);
        // Create a Client to handle OAuth authorization
        var oauthClient = OAuthProvider.CreateOAuthClient(config, context);
        // Create a Browser for login
        var browser = BrowserProvider.Create(oauthClient, config);
        // Account files are saved locally and have not expired
        var isLoggedIn = oauthClient.IsAccountFileExist() && !oauthClient.IsAccessTokenExpired();
        // Login
        if (!isLoggedIn)
        {
            // If already authorized but expired, it gets reauthorized.
            // Otherwise opens a browser and starts the authorization flow.
            oauthClient.Login(
                browser,
                (account) => { /*if login is successful*/ },
                (error) => { /*if login fails*/ }
            );
        }
    }
}
```

# Loading models

```
var defaultApi = new DefaultApi(oauthClient);
oauthClient.Login(
    browser,
    (account) => {
        // Initialize the ModelLoader used for model loading
        ModelLoader.Initialize(
            config,                // Config created from Credentials
            defaultApi,            // Authorized API
            "PASSWORD_FOR_YOUR_APP", // The model's encryption password
            10                     // Maximum number of caches for a model
        );

        defaultApi.GetAccountCharacterModels(10, (models) => {
            // Start model loading
            ModelLoader.LoadVrm(
                models[0], // Model to load
                (gameObject) => {
                    // Callback after loading is complete
                    gameObject.transform.parent = this.transform;
                },
                (progress) => {
                    // Loading progress callback
                },
                (error) => {
                    // Callback when an error occurs
                }
            );
        }, (error) => { });
    },
    (error) => { /*if login fails*/ }
);
```