

Jellyfin

- [Plugins](#)
 - [Custom Tabs](#)
 - [Plugin Template](#)
 - [!Catalogue](#)
 - [JS Injector](#)

Plugins

Custom Tabs

Repository	https://www.iamparadox.dev/jellyfin/plugins/manifest.json
ver	0.2.6.0
	Link

XML Config

```
<?xml version="1.0" encoding="utf-8"?>
<PluginConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Tabs>
    <TabConfig>
      <ContentHtml>&lt;style&gt;
#customTab_0 {
  :root {
    --save-gut: max(env(safe-area-inset-left), 3.3%);
  }
  @media only screen and (max-width: 990px) {
    .appsIframe{
      height: calc(100vh - 105.15px)!important;
      margin-top: -6.15px!important;
    }
  }
}
```


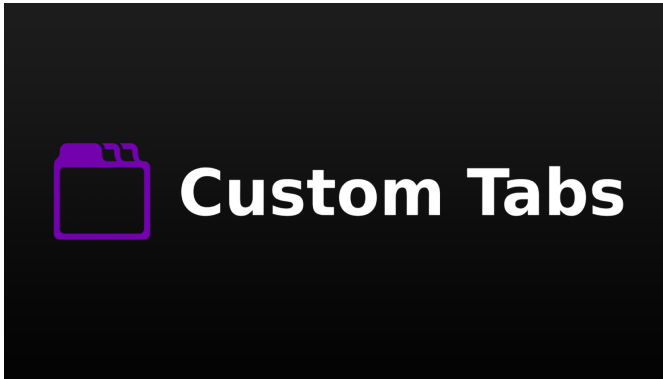
```

    .appsIframe {
        width: 100%;
        height: calc(100vh - 50.716px);
        position: absolute;
        border: 0;
        margin-top: -20.716px;
    } }
</style>
<div class="sections">
    <iframe class="appsIframe" src="https://apps.netflix.pknw1.co.uk"></iframe>
</div></ContentHtml>
    <Title>User Apps</Title>
</TabConfig>
<TabConfig>
    <ContentHtml><style>
#customTab_1 {
    :root {
        --save-gut: max(env(safe-area-inset-left), 3.3%);
    }
    @media only screen and (max-width: 990px) {
        .appsIframe{
            height: calc(100vh - 105.15px)!important;
            margin-top: -6.15px!important;
        }
    }
    .appsIframe {
        width: 100%;
        height: calc(100vh - 50.716px);
        position: absolute;
        border: 0;
        margin-top: -20.716px;
    } }
</style>
<div class="sections">
    <iframe class="appsIframe" src="https://gas.netflix.pknw1.co.uk"></iframe>
</div></ContentHtml>
    <Title>*NEW* Console Games</Title>
</TabConfig>
</Tabs>

```

```
</PluginConfiguration>
```

Plugin Template

	
Developer	
Repository	
ver	
Config Link	Link

XML Config

```
/home/apps/frontend/jellyfin-10.10.7/config/plugins/configurations
```

```
conf
```

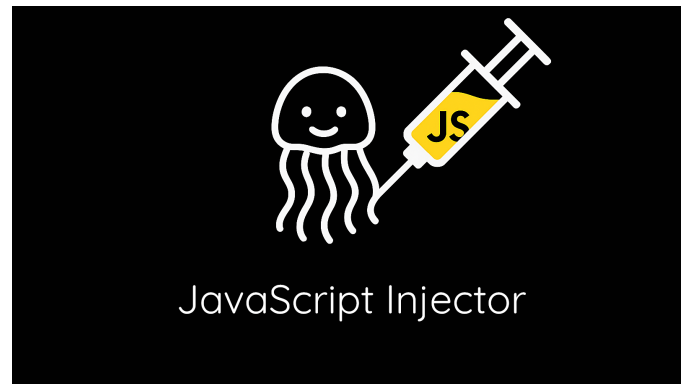

!Catalogue

<https://notflix.pknw1.co.uk/web/#/dashboard/plugins/repositories>

- <https://raw.githubusercontent.com/9p4/jellyfin-plugin-ss0/manifest-release/manifest.json>
- <https://github.com/lachlandcp/jellyfin-editors-choice-plugin/raw/main/manifest.json>
- <https://raw.githubusercontent.com/crobibero/jellyfin-plugin-tmdb-trailers/master/manifest.json>
- <https://www.iamparadox.dev/jellyfin/plugins/manifest.json>
- <https://raw.githubusercontent.com/9p4/jellyfin-plugin-ss0/manifest-release/manifest.json>
- <https://raw.githubusercontent.com/bvolv/CustomLogoPlugin/main/manifest.json>
- <https://raw.githubusercontent.com/pknw1-development/jelly-static-assets/refs/heads/main/repository/manifest.json>
- <https://raw.githubusercontent.com/Eeeeelias/playlist-generator/refs/heads/main/manifest.json>
- <https://raw.githubusercontent.com/0belous/universal-plugin-repo/refs/heads/main/manifest.json>
- <https://raw.githubusercontent.com/n00bcd/00r/jellyfin-enhanced/main/manifest.json>
- <https://raw.githubusercontent.com/ankenyr/jellyfin-plugin-repo/master/manifest.json>
- <https://raw.githubusercontent.com/GrandguyJS/media-upload-plugin/main/manifest.json>
- <https://raw.githubusercontent.com/jyourstone/jellyfin-plugin-manifest/refs/heads/main/manifest.json>
- <https://raw.githubusercontent.com/n00bcd/00r/jellyfin-plugins/main/10.10/manifest.json>
- <https://raw.githubusercontent.com/n00bcd/00r/Jellyfin-JavaScript-Injector/refs/heads/main/manifest.json>

JS Injector

JS Injector



Auto-Whitelist Cookies

- on index.html
- checks for `notflixuid` cookie
- if cookie DOES NOT exist
- loads URL `/resume/gen/<username>`
- URL is mapped to custom python api.notflix.pknw1.co.uk via nginx proxy manager
- reads the payload
- creates 1 year JWT
- returns JWT to store as cookie

```
(function () {  
  function getCookie(name) {  
    return document.cookie  
      .split('; ')  
      .find(row => row.startsWith(name + '='))  
      ?.split('=')[1];  
  }  
  
  function setCookie(name, value, seconds) {
```

```
document.cookie = `${name}=${encodeURIComponent(value)}; path=/; max-age=${seconds};
secure; samesite=lax`;
}

async function getSession(username) {
  const res = await fetch('/resume/gen/'+username);
  const data = await res.text(); // or res.text() if not JSON
  return data;
}

const existing = getCookie('notflixuid');

if (!existing) {
  setTimeout(() => {
    const userButton = document.querySelector(".headerUserButton");
    const uid = userButton?.title?.toLowerCase();
    if (uid) {
      getSession(uid).then(sess => {
        setCookie('notflixuid', sess, 34560000); // 7 days
        console.log('Cookie set:', sess);
      });
    }

  }, 5000);
} else {
  console.log('Cookie exists:', decodeURIComponent(existing));
}
})();
```