

jellyfin-mpv-shim

Cast media from Jellyfin Mobile and Web apps to MPV

runtime dependencies	mpv (Audio and video applications)
	python3 (Interpreter languages and scripting)
	tk (Graphic libraries and applications/Graphic libraries)
	python-jinja2 (Interpreter languages and scripting/External python modules)
	python-jellyfin-apiclient
	python-mpv
	python-mpv-jsonipc
	python-pywebview
make dependencies	gettext (Development tools)

```

jellyfin-mpv-shim
├─ mpv (Audio and video applications)
├─ python3 (Interpreter languages and scripting)
├─ tk (Graphic libraries and applications/Graphic libraries)
├─ python-jinja2 (Interpreter languages and scriptingExternal python
modules)
├─ gettext (Development tools)
├─ python-jellyfin-apiclient (Custom)
│   └─ python (Interpreter languages and scripting)
│   └─ python-certifi (Interpreter languages and scripting/External python
modules)
│   └─ python-requests (Interpreter languages and scripting/External python
modules)
│   └─ python-six (Interpreter languages and scripting/External python
modules)
│   └─ python-urllib3 (Interpreter languages and scripting/External python
modules)
│   └─ python-websocket-client (Interpreter languages and
scripting/External python modules)
├─ python-mpv (Custom)
│   └─ mpv (Audio and video applications)
│   └─ python (Interpreter languages and scripting)
│   └─ python-pillow (Interpreter languages and scripting/External python
modules)
├─ python-mpv-jsonipc (Custom)
│   └─ python (Interpreter languages and scripting)
├─ python-pywebview (Custom)
│   └─ python (Interpreter languages and scripting)
│   └─ python-proxy-tools (Custom)
│       └─ python (Interpreter languages and scripting)
└─ mpv-shim-default-shaders (Custom)

```

```

mkdir /data/buildroot/package/jellyfin-mpv-shim
echo "sha256
38dd5096babe97f87969db70c503be3cefd5af901f1055109a7817b6ee724bed
v2.0.2.tar.gz" > /data/buildroot/package/jellyfin-mpv-shim/v2.0.2.hash
cat > /data/buildroot/package/jellyfin-mpv-shim/Config.in<< 'EOF'
config BR2_PACKAGE_JELLYFIN_MPV_SHIM
    bool "jellyfin-mpv-shim"
    select BR2_PACKAGE_MPV
    select BR2_PACKAGE_TK
    select BR2_PACKAGE_GETTEXT_TINY
    select BR2_PACKAGE_PYTHON_SETUPTOOLS
    select BR2_PACKAGE_PYTHON_MPV
    select BR2_PACKAGE_PYTHON_MPV_JSONIPC
    select BR2_PACKAGE_PYTHON_JELLYFIN_APICLIENT
    help
        Cast media from Jellyfin Mobile and Web apps to MPV
EOF

cat > /data/buildroot/package/jellyfin-mpv-shim/jellyfin-mpv-shim.mk<< 'EOF'
#####
#
# jellyfin-mpv-shim
#
#####

JELLYFIN_MPV_SHIM_VERSION = 2.0.2
JELLYFIN_MPV_SHIM_SOURCE = v$(JELLYFIN_MPV_SHIM_VERSION).tar.gz
JELLYFIN_MPV_SHIM_SITE =
https://github.com/jellyfin/jellyfin-mpv-shim/archive/refs/tags

JELLYFIN_MPV_SHIM_DEPENDENCIES = mpv tk gettext python-setuptools python-mpv
python-mpv-jsonipc python-jellyfin-apiclient

define JELLYFIN_MPV_SHIM_BUILD_CMDS
    cd $(@D)/ && \
    find -iname '*.po' | while read -r _file; do $(HOST_DIR)/bin/msgfmt
"${$_file}" -o "${$_file%.*}.mo" ; done && \
    PYTHONPATH="$(ls -d $(TARGET_DIR)/usr/lib/*/ | grep python)/site-
packages" \
    $(HOST_DIR)/usr/bin/python setup.py build
endef

define JELLYFIN_MPV_SHIM_INSTALL_TARGET_CMD
    cd "$(TARGET_DIR)" && \
    _sitepkg="$(ls -d usr/lib/*/ | grep python)site-
packages/jellyfin_mpv_shim" && \
    mkdir -p "$(TARGET_DIR)/$_sitepkg" && \
    cp -r "$(TARGET_DIR)/usr/share/mpv-shim-default-shaders"
"$$(TARGET_DIR)/$_sitepkg/default_shader_pack" && \
    cd $(@D)/ && \
    PYTHONPATH="$(ls -d $(TARGET_DIR)/usr/lib/*/ | grep python)/site-

```

```
packages" \  
    $(HOST_DIR)/usr/bin/python setup.py install --prefix="$(TARGET_DIR)/usr"  
--optimize=1 --skip-build && \  
    install -Dm 644 "LICENSE.md" "$(TARGET_DIR)/usr/share/licenses/jellyfin-  
mpv-shim/LICENSE"  
endif  
  
$(eval $(generic-package))  
EOF  
  
sed -i '/menu "Custom packages"/a source "package/jellyfin-mpv-  
shim/Config.in"' /data/buildroot/package/Config.in
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=buildroot:jellyfin-mpv-shim>

Last update: **2025/02/19 10:59**

