

文件管理器书签视图插件

1. 继承ViewInterface，遵守Qt的插件规范实现你的书签视图插件接口

● 插件头文件

如果你安装了libdde-file-manager-dev你可以在如下路径找到**viewinterface.h**

- 在/etc/apt/source.list中添加如下源：

```
deb-src http://pools.corp.deepin.com/deepin unstable main non-free
```
- 升级安装libdde-file-manager-dev

```
sudo apt-get update
sudo apt-get install libdde-file-manager-dev
```
- 就可以找到/usr/include/dde-file-manager/viewinterface.h

同样可以在 git clone <https://github.com/linuxdeepin/dde-file-manager> 下载源码到dde-file-manager，切到develop2.0分支 可以找到dde-file-manager/dde-file-manager-plugins/plugininterfaces/view/viewinterface.h

● 接口说明:

- virtual QString bookMarkText(); 侧边栏书签对应文本
- virtual QIcon bookMarkNormalIcon(); 侧边栏书签normal状态图标
- virtual QIcon bookMarkHoverIcon(); 侧边栏书签hover状态图标
- virtual QIcon bookMarkPressedIcon(); 侧边栏书签pressed状态图标
- +virtual QIcon bookMarkCheckedIcon(); 侧边栏书签checked状态图标
- virtual QString crumbText(); 顶部面包屑文本
- virtual QIcon crumbNormalIcon(); 顶部面包屑normal状态图标
- virtual QIcon crumbHoverIcon(); 顶部面包屑hover状态图标
- virtual QIcon crumbPressedIcon(); 顶部面包屑pressed状态图标
- virtual QIcon crumbCheckedIcon(); 顶部面包屑checked状态图标
- virtual bool isAddSeparator(); 侧边栏书签是否添加分隔符
- virtual QString scheme(); 视图遵循的协议例如computer:// 、trash://、usershare:// 中的computer、trash、usershare等字符串
- virtual QWidget\* createView(); 返回视图协议对应的QWidget对象指针

● 插件默认安装目录

- x86架
  - /usr/lib/x86\_64-linux-gnu/dde-file-manager/plugins/view/ 64位
  - /usr/lib/x86\_32-linux-gnu/dde-file-manager/plugins/view/32位
- 龙芯架构
  - /usr/lib/mipsel-linux-gnu//dde-file-manager/plugins/view/
- 神威架构
  - /usr/lib/sw\_64-linux-gnu/dde-file-manager/plugins/view/

● 源码:

```
#ifndef VIEWINTERFACE_H
#define VIEWINTERFACE_H

#include <QString>
#include <QIcon>
#include <QWidget>
/*
 * The view interface of dde-file-manager
 * 文件管理器的视图插件接口
 */
class ViewInterface
{
public:
    virtual ~ViewInterface() {}

    /*!
     * \brief bookMarkText
     * \return QString: the text of leftbar bookmark which should be 16X16 size
     */
    /*!
     * \brief bookMarkText
     * \return QString: 侧边栏书签文本内容
     */
    virtual QString bookMarkText(){
        return "";
    }

    /*!
     * \brief bookMarkNormalIcon
     * \return QIcon: the release icon of leftbar bookmark which should be 16X16 size
     */
    /*!
     * \brief bookMarkNormalIcon
     * \return QIcon: release状态-侧边栏书签Icon
     */
    virtual QIcon bookMarkNormalIcon(){
        return QIcon();
    }

    /*!
     * \brief bookMarkHoverIcon
     * \return QIcon: the hover icon of leftbar bookmark which should be 16X16 size
     */
    /*!
     * \brief bookMarkHoverIcon
     * \return QIcon: hover状态-侧边栏书签Icon
     */
    virtual QIcon bookMarkHoverIcon(){
        return QIcon();
    }

    /*!
     * \brief bookMarkPressedIcon
     */
}
```

```
* \return QIcon: the pressed hover icon of leftbar bookmark which should be 16X16 size
*/
/*!
 * \brief bookMarkPressedIcon
 * \return QIcon: pressed状态-侧边栏书签Icon
 */
virtual QIcon bookMarkPressedIcon(){
    return QIcon();
}

/*!
 * \brief bookMarkCheckedIcon
 * \return QIcon: the checked icon of leftbar bookmark which should be 16X16 size
 */
/*!
 * \brief bookMarkCheckedIcon
 * \return QIcon: checked状态-侧边栏书签Icon
 */
virtual QIcon bookMarkCheckedIcon(){
    return QIcon();
}

/*!
 * \brief crumbText
 * \return QString: text of first crumbButton in titlebar
 */
/*!
 * \brief crumbText
 * \return QString: 第一个面包屑的显示文本
 */
virtual QString crumbText(){
    return "";
}

/*!
 * \brief crumbNormalIcon
 * \return QIcon: the normal icon of first crumbButton in titlebar
 */
/*!
 * \brief crumbNormalIcon
 * \return QIcon: 第一个面包屑的normal状态图标
 */
virtual QIcon crumbNormalIcon(){
    return QIcon();
}

/*!
 * \brief crumbHoverIcon
 * \return QIcon: the hover icon of first crumbButton in titlebar
 */
/*!
 * \brief crumbHoverIcon
 * \return QIcon: 第一个面包屑的normal状态图标
 */
virtual QIcon crumbHoverIcon(){
    return QIcon();
}

/*!
 * \brief crumbPressedIcon
 * \return QIcon: the pressed icon of first crumbButton in titlebar
 */
/*!
 * \brief crumbHoverIcon
 * \return QIcon: 第一个面包屑的pressed状态图标
 */
virtual QIcon crumbPressedIcon(){
    return QIcon();
}

/*!
 * \brief crumbCheckedIcon
 * \return QIcon: the checked icon of first crumbButton in titlebar
 */
/*!
 * \brief crumbHoverIcon
 * \return QIcon: 第一个面包屑的checked状态图标
 */
virtual QIcon crumbCheckedIcon(){
    return QIcon();
}

}

/*!
 * \brief isAddSeparator
 * \return bool: addSeparator whether or not
 */
/*!
 * \brief isAddSeparator
 * \return bool: 是否添加书签分隔符
 */
virtual bool isAddSeparator(){
    return false;
}

}

/*!
 * \brief scheme
 * \return QString: scheme of view which should by like computer 、trash、 usershare and so on
 */
/*!
 * \brief scheme
 * \return QString: 视图view对应的协议头, 譬如computer 、trash、 usershare等
 */
virtual QString scheme(){
    return "";
}

}

/*!
 * \brief createView
 * \return QWidget* : the view will be registered into dde-file-manager view
 */
/*!
 * \brief createView
 * \return QWidget*: 注册到文件管理器视图区域的视图控件
 */
virtual QWidget* createView(){
    return NULL;
}

}

};

#define ViewInterface_iid "com.deepin.dde-file-manager.ViewInterface"
Q_DECLARE_INTERFACE(ViewInterface,ViewInterface_iid)
```

```
#endif // VIEWINTERFACE_H
```

#### ● 示例

```
#include "viewplugin.h"
#include <QLabel>

ViewPlugin::ViewPlugin(QObject *parent) :
    QObject(parent)
{
    Q_INIT_RESOURCE(pluginview);
}

QString ViewPlugin::bookMarkText()
{
    return "view";
}

QIcon ViewPlugin::bookMarkNormalIcon()
{
    return QIcon(":/images/release.png");
}

QIcon ViewPlugin::bookMarkHoverIcon()
{
    return QIcon(":/images/hover.png");
}

QIcon ViewPlugin::bookMarkPressedIcon()
{
    return QIcon(":/images/pressed.svg");
}

QIcon ViewPlugin::bookMarkCheckedIcon()
{
    return QIcon(":/images/checked.svg");
}

QString ViewPlugin::crumbText()
{
    return "view";
}

QIcon ViewPlugin::crumbNormalIcon()
{
    return QIcon(":/images/release.png");
}

QIcon ViewPlugin::crumbHoverIcon()
{
    return QIcon(":/images/hover.png");
}

QIcon ViewPlugin::crumbPressedIcon()
{
    return QIcon(":/images/hover.png");
}

QIcon ViewPlugin::crumbCheckedIcon()
{
    return QIcon(":/images/checked.svg");
}

bool ViewPlugin::isAddSeparator()
{
    return true;
}

QString ViewPlugin::scheme()
{
    return "view";
}

QWidget *ViewPlugin::createView()
{
    QLabel* label = new QLabel;
    label->setText("view");
    label->setAlignment(Qt::AlignCenter);
    // label->setStyleSheet("background-color: green; margins:0");
    return label;
}
```

#### ● 预览效果

