

DocPortalNC

NITROMVC - COREBASE - NOTIONCUBESOFTWARE GMBH

Table of Contents

Symbol Reference	1
Classes	1
NCDiagramApiController Class	1
NCDiagramImageView Class	2
NCDiagramImageView::NCDiagramImageView Constructor	3
NCDiagramImageView Methods	3
NCDiagramImageView::Paint Method	3
NCDiagramImageView::drawBorder Method	4
NCDiagramImageView::drawImage Method	5
NCDiagramImageView::drawRing Method	5
drawStatus Method	6
NCDiagramImageView::drawTitle Method	6
NCDiagramImageView::drawVal Method	7
NCDiagramImageView::drawValue Method	7
NCDiagramImageView::drawValues Method	8
NCDiagramImageView::val Method	9
NCDiagramImageView::valInt Method	9
NCFrontendController Class	9
NCFrontendController Methods	10
NCFrontendController::callScene Method	10
clearScene Method	10
NCFrontendController::loadScene Method	12
NCFrontendCoreView Class	12
NCFrontendCoreView::NCFrontendCoreView Constructor	13
NCFrontendCoreView::~NCFrontendCoreView Destructor	14
NCFrontendCoreView Methods	14
NCFrontendCoreView::Footer Method	14
NCFrontendCoreView::Layout Method	15
NCFrontendCoreView::Leftbar Method	16
NCFrontendCoreView::Navigation Method	16
NCFrontendCoreView::Script Method	17
NCFrontendCoreView::Style Method	17
NCFrontendCoreView::Toolbar Method	18
NCFrontendCoreView::callScene Method	19
NCFrontendCoreView::rowBrand Method	19
NCFrontendView Class	20
NCFrontendView::NCFrontendView Constructor	21
NCFrontendView::~NCFrontendView Destructor	22

NCFrontendViewFull Class	22
NCFrontendViewFull::NCFrontendViewFull Constructor	24
NCFrontendViewFull::~~NCFrontendViewFull Destructor	25
NCFrontendViewFull Methods	25
NCFrontendViewFull::Footer Method	25
NCFrontendViewFull::Layout Method	26
NCFrontendViewFull::Navigation Method	26
NCFrontendViewFull::Toolbar Method	27
NCHomeController Class	28
PortalNC Class	29
Files	29
NCDiagramAPIController.cpp	30
NCDiagramAPIController.h	35
NCDiagramImageView.cpp	36
NCDiagramImageView.h	40
NCFrontendController.cpp	41
NCFrontendController.h	42
NCFrontendCoreView.cpp	43
NCFrontendCoreView.h	46
NCFrontendView.cpp	47
NCFrontendView.h	48
NCHomeController.cpp	49
NCHomeController.h	53
Portal.cpp	54
PortalNC.cpp	54
PortalNC.h	55
main.cpp	55

Index









a

1 Symbol Reference

1.1 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	NCDiagramApiController (see page 1)	
	NCDiagramImageView (see page 2)	
	NCFrontendController (see page 9)	
	NCFrontendCoreView (see page 12)	
	NCFrontendView (see page 20)	
	NCFrontendViewFull (see page 22)	
	NHomeController (see page 28)	
	PortalNC (see page 29)	

Legend

	Class
---	-------

1.1.1 NCDiagramApiController Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.06.2020

C++

```
class NCDiagramApiController : public TemplateWebController;
```

Inheritance



Group

Classes ([see page 1](#))

Context

DefaultNamespace

Method

NCDiagramApiController

Class

NCDiagramApiController

1.1.2 NCDiagramImageView Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

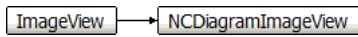
Date

09.06.2020

C++

```
class NCDiagramImageView final : public ImageView;
```

Inheritance



Group

Classes (see page 1)

Legend

	Constructo r
--	-----------------

Members

Methods

	Name	Description
	NCDiagramImageView (see page 3)	

NCDiagramImageView Methods

	Name	Description
	Paint (see page 3)	
	drawBorder (see page 4)	
	drawImage (see page 5)	
	drawRing (see page 5)	
	drawStatus() (see page 6)	
	drawTitle (see page 6)	
	drawVal (see page 7)	
	drawValue (see page 7)	
	drawValues (see page 8)	
	val (see page 9)	
	valInt (see page 9)	

Methods

	Name	Description
	NCDiagramImageView (see page 3)	

NCDiagramImageView Methods

	Name	Description
≡◆	Paint (↗ see page 3)	
≡◆	drawBorder (↗ see page 4)	
≡◆	drawImage (↗ see page 5)	
≡◆	drawRing (↗ see page 5)	
≡◆	drawStatus() (↗ see page 6)	
≡◆	drawTitle (↗ see page 6)	
≡◆	drawVal (↗ see page 7)	
≡◆	drawValue (↗ see page 7)	
≡◆	drawValues (↗ see page 8)	
≡◆	val (↗ see page 9)	
≡◆	vallnt (↗ see page 9)	

Context

DefaultNamespace

Method

ImagePNG

ClassNCHomeController ([↗](#) see page 28)**1.1.2.1 NCDiagramImageView::NCDiagramImageView Constructor****C++**

```
NCDiagramImageView(const QString& content, qint32 width, qint32 height, const QString& path);
```

Description

This is NCDiagramImageView, a member of class NCDiagramImageView.

Class

NCDiagramImageView Class ([↗](#) see page 2)

1.1.2.2 NCDiagramImageView Methods**1.1.2.2.1 NCDiagramImageView::Paint Method****Author**

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1: void NCDiagramImageView::Paint()
2: {
3:     O_STACK(Img, Paint);
```

```
4:     painter().setRenderHint(QPainter::Antialiasing);
5:
6:     drawImage();
7:     drawRing();
8:     drawBorder();
9:     drawValues();
10:    drawStatus();
11:    drawTitle();
12:    drawValue();
13:    drawImage(true);
14: }
```

C++

```
void Paint() override;
```

Class

NCDiagramImageView Class ([see page 2](#))

Context

DefaultNamespace

Method

Paint

Class

NCDiagramImageView ([see page 2](#))

1.1.2.2.2 NCDiagramImageView::drawBorder Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1: void NCDiagramImageView::drawBorder()
2: {
3:     painter().setPen(QPen(QColor(255, 255, 255), 8));
4:     painter().setBrush(Qt::NoBrush);
5:     painter().drawRect(QRect(8, 8, width() - 16, height() - 16));
6: }
```

C++

```
void drawBorder();
```

Class

NCDiagramImageView Class ([see page 2](#))

Context

DefaultNamespace

Method

drawBorder

Class

NCDiagramImageView ([see page 2](#))

1.1.2.2.3 NCDiagramImageView::drawImage Method

C++

```
void drawImage(bool overflow = false);
```

Description

This is drawImage, a member of class NCDiagramImageView.

Class

NCDiagramImageView Class ([see page 2](#))

1.1.2.2.4 NCDiagramImageView::drawRing Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1: void NCDiagramImageView::drawRing()
2: {
3:     const qint32 outer = valInt("RingOuter");
4:     const qint32 inner = valInt("RingInner");
5:
6:     const qint32 w2 = width() / 2;
7:     const qint32 val = w2 - w2 * outer / 100;
8:     const qint32 val2 = w2 - w2 * inner / 100;
9:
10:    if (outer || inner)
11:    {
12:        QPainterPath path;
13:        if (outer) path.addEllipse(val, val, width() - val - val, height() - val - val);
14:        if (inner) path.addEllipse(val2, val2, width() - val2 - val2, height() - val2 -
val2);
15:
16:        painter().setPen(QPen(QColor(255, 255, 255), 8));
17:        painter().setBrush(QBrush(QColor(128, 128, 128, 64)));
18:        painter().drawPath(path);
19:    }
20: }
```

C++

```
void drawRing();
```

Class

NCDiagramImageView Class ([see page 2](#))

Context

DefaultNamespace

Method

drawRing

Class

NCDiagramImageView ([see page 2](#))

1.1.2.2.5 drawStatus Method

1.1.2.2.5.1 NCDiagramImageView::drawStatus Method ()

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1:
2: void NCDiagramImageView::drawStatus()
3: {
4:     const qint32 blue = valInt("StatusBlue");
5:     const qint32 green = valInt("StatusGreen");
6:     const qint32 yellow = valInt("StatusYellow");
7:     const qint32 red = valInt("StatusRed");
8:
9:     if (blue || green || yellow || red)
10:    {
11:        const qint32 stepx = (width() - 16) / 10;
12:        for (qint32 pos = 0; pos < 10; pos++)
13:        {
14:            drawStatus(pos, stepx, blue, green, yellow, red);
15:        }
16:    }
17: }
18:
19:
```

C++

```
void drawStatus();
void drawStatus(qint32 pos, qint32 stepx, qint32 blue, qint32 green, qint32 yellow, qint32 red);
```

Class

NCDiagramImageView Class ([🔗](#) see page 2)

Context

DefaultNamespace

Method

drawStatus

Class

NCDiagramImageView ([🔗](#) see page 2)

1.1.2.2.6 NCDiagramImageView::drawTitle Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1: void NCDiagramImageView::drawTitle()
2: {
3:     const QString title = val("Title");
4:     if (!title.isEmpty())
5:     {
6:         painter().save();
7:         painter().setFont(QFont("Helvetica", 80));
8:         painter().drawText(QRectF(0, 20, width(), 128), Qt::AlignCenter, title);
9:         painter().restore();
10:    }
11: }
```

C++

```
void drawTitle();
```

ClassNCDiagramImageView Class ([see page 2](#))**Context**

DefaultNamespace

Method

drawTitle

ClassNCDiagramImageView ([see page 2](#))

1.1.2.2.7 NCDiagramImageView::drawVal Method

C++

```
void drawVal(qint32 val, qint32 i, qint32 stepx);
```

Description

This is drawVal, a member of class NCDiagramImageView.

ClassNCDiagramImageView Class ([see page 2](#))

1.1.2.2.8 NCDiagramImageView::drawValue Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1: void NCDiagramImageView::drawValue()
2: {
3:     const QString value = val("Value");
4:     if (!value.isEmpty())
5:     {
```

```
6:     painter().save();
7:     painter().setFont(QFont("Helvetica", 4 * 60, QFont::Bold));
8:     painter().drawText(QRectF(0, 128, width(), 128 + 4 * 60), Qt::AlignCenter,
value);
9:     painter().restore();
10:    }
11: }
```

C++

```
void drawValue();
```

Class

NCDiagramImageView Class ([see page 2](#))

Context

DefaultNamespace

Method

drawValue

Class

NCDiagramImageView ([see page 2](#))

1.1.2.2.9 NCDiagramImageView::drawValues Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

06.09.2020

Body Source

```
1: void NCDiagramImageView::drawValues()
2: {
3:     const qint32 stepx = (width() - 16) / 10;
4:
5:     drawVal(valInt("Val1"), 0, stepx);
6:     drawVal(valInt("Val2"), 1, stepx);
7:     drawVal(valInt("Val3"), 2, stepx);
8:     drawVal(valInt("Val4"), 3, stepx);
9:     drawVal(valInt("Val5"), 4, stepx);
10:    drawVal(valInt("Val6"), 5, stepx);
11:    drawVal(valInt("Val7"), 6, stepx);
12:    drawVal(valInt("Val8"), 7, stepx);
13:    drawVal(valInt("Val9"), 8, stepx);
14:    drawVal(valInt("Val10"), 9, stepx);
15: }
```

C++

```
void drawValues();
```

Class

NCDiagramImageView Class ([see page 2](#))

Context

DefaultNamespace

Method

drawValues

Class

[NCDiagramImageView](#) (see page 2)

1.1.2.2.10 NCDiagramImageView::val Method

Body Source

```
1: QString val(const QString& name)
2: {
3:     return m_Strings[name];
4: }
```

C++

```
QString val(const QString& name);
```

Description

This is val, a member of class NCDiagramImageView.

Class

[NCDiagramImageView Class](#) (see page 2)

1.1.2.2.11 NCDiagramImageView::valInt Method

Body Source

```
1: qint32 valInt(const QString& name)
2: {
3:     return m_Values[name];
4: }
```

C++

```
qint32 valInt(const QString& name);
```

Description

This is valInt, a member of class NCDiagramImageView.

Class

[NCDiagramImageView Class](#) (see page 2)

1.1.3 NCFrontendController Class

Author

Tobias Post

Company

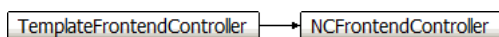
NOTIONCUBE Software GmbH

Date



12.05.2020

C++




```
class NCFrontendController : public TemplateFrontendController;
```

Inheritance




GroupClasses ([see page 1](#))**Legend**

	Method
	protected

Members**NCFrontendController Methods**

	Name	Description
	callScene (see page 10)	
	clearScene() (see page 10)	
	loadScene (see page 12)	

NCFrontendController Methods

	Name	Description
	callScene (see page 10)	
	clearScene() (see page 10)	
	loadScene (see page 12)	

Context

DefaultNamespace

Method

NCFrontendController

Class

NCFrontendController

1.1.3.1 NCFrontendController Methods

1.1.3.1.1 NCFrontendController::callScene Method

C++

```
QString callScene(const QString& url, qint32 delay);
```

Description

This is callScene, a member of class NCFrontendController.

Class

NCFrontendController Class ([see page 9](#))

1.1.3.1.2 clearScene Method

1.1.3.1.2.1 NCFrontendController::clearScene Method ()

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

18.05.2020

Body Source

```
1: QString NCFrontendController::clearScene() const
2: {
3:     return Core3DCanvas::clearScene();
4: }
```

C++

```
QString clearScene() const;
```

ClassNCFrontendController Class ([↗](#) see page 9)**Context**

DefaultNamespace

Method

clearScene

ClassNCFrontendController ([↗](#) see page 9)

1.1.3.1.2.2 NCFrontendController::clearScene Method (const QString&)

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

18.05.2020

Body Source

```
1: QString NCFrontendController::clearScene(const QString& name) const
2: {
3:     return Core3DCanvas::clearScene(name);
4: }
```

C++

```
QString clearScene(const QString& name) const;
```

ClassNCFrontendController Class ([↗](#) see page 9)**Context**

DefaultNamespace

Method

clearScene

ClassNCFrontendController ([↗](#) see page 9)

1.1.3.1.3 NCFrontendController::loadScene Method

C++

```
QString loadScene(const QString& name, QVector3D pos = QVector3D()) const;
```

Description

This is loadScene, a member of class NCFrontendController.

Class

NCFrontendController Class ([see page 9](#))

1.1.4 NCFrontendCoreView Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

C++

```
class NCFrontendCoreView : public TemplateFrontendView;
```

Inheritance



Group

Classes ([see page 1](#))

Legend

	Constructo r
	virtual


Members

Methods



	Name	Description
	NCFrontendCoreView (see page 13)	
	~NCFrontendCoreView (see page 14)	

NCFrontendCoreView Methods










	Name	Description
	Footer (see page 14)	
	Layout (see page 15)	
	Leftbar (see page 16)	
	Navigation (see page 16)	
	Script (see page 17)	
	Style (see page 17)	
	Toolbar (see page 18)	

	callScene (see page 19)	
	rowBrand (see page 19)	

Methods

	Name	Description
	NCFrontendCoreView (see page 13)	
	~NCFrontendCoreView (see page 14)	

NCFrontendCoreView Methods

	Name	Description
	Footer (see page 14)	
	Layout (see page 15)	
	Leftbar (see page 16)	
	Navigation (see page 16)	
	Script (see page 17)	
	Style (see page 17)	
	Toolbar (see page 18)	
	callScene (see page 19)	
	rowBrand (see page 19)	

Context

DefaultNamespace

MethodNCFrontendView ([see page 20](#))**Class**NCFrontendView ([see page 20](#))

1.1.4.1 NCFrontendCoreView::NCFrontendCoreView Constructor

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: NCFrontendCoreView::NCFrontendCoreView()
```

C++

```
NCFrontendCoreView();
```

ClassNCFrontendCoreView Class ([see page 12](#))**Context**

DefaultNamespace

Method

NCFrontendCoreView

Class

NCFrontendCoreView

1.1.4.2 NCFrontendCoreView::~~NCFrontendCoreView Destructor

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: NCFrontendCoreView::~~NCFrontendCoreView()=
```

C++

```
virtual ~NCFrontendCoreView();
```

ClassNCFrontendCoreView Class ([↗](#) see page 12)**Context**

DefaultNamespace

Method~NCFrontendView ([↗](#) see page 20)**Class**NCFrontendView ([↗](#) see page 20)

1.1.4.3 NCFrontendCoreView Methods

1.1.4.3.1 NCFrontendCoreView::~Footer Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::~Footer()
2: {
3:     //TemplateFrontendView::Footer();
4:     Open(div().Py(2).Class("organismfooterlinks"));
5:     Open(footer().Class("footer", "text-center"));
6:
7:     Open(container().Py(2).Class("text-center"));
8:     Open(span().Class("fa-stack", "text-primary"));
9:     i().Pad().Icon("fa-circle").Class("fa-stack-2x");
```

```
10:             i().Fad().Icon("fa-cube").Class("fa", "fa-inverse", "fa-stack-1x");
11:             Close();
12:
13:             raw(" #notioncube3d (" + QString(__TIMESTAMP__) + ")");
14:
15:             Close();
16:             Close();
17:             Close();
18: }
```

C++

```
void Footer() override;
```

Class

NCFrontendCoreView Class ([↗](#) see page 12)

Context

DefaultNamespace

Method

Footer

Class

NCFrontendView ([↗](#) see page 20)

1.1.4.3.2 NCFrontendCoreView::Layout Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::Layout()
2: {
3:     //LayoutFrontend();
4:
5:     LayoutHeader(true, true);
6:
7:     Open(containerfluid());
8:     Scope("Content");
9:     Close();
10:
11:     LayoutFooter(false);
12: }
```

C++

```
void Layout() override;
```

Class

NCFrontendCoreView Class ([↗](#) see page 12)

Context

DefaultNamespace

Method

Layout

Class

NCFrontendView ([↗](#) see page 20)

1.1.4.3.3 NCFrontendCoreView::Leftbar Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::Leftbar()  
2: {  
3:     TemplateFrontendView::Leftbar();  
4: }
```

C++

```
void Leftbar() override;
```

Class

NCFrontendCoreView Class ([↗](#) see page 12)

Context

DefaultNamespace

Method

Leftbar

Class

NCFrontendCoreView ([↗](#) see page 12)

1.1.4.3.4 NCFrontendCoreView::Navigation Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::Navigation()  
2: {  
3:     FullNavigation("#notioncube3d", "fa-cube");  
4: }
```

C++

```
void Navigation() override;
```

Class

NCFrontendCoreView Class ([↗](#) see page 12)

Context

DefaultNamespace

Method

Navigation

Class

NCFrontendView (🔗 see page 20)

1.1.4.3.5 NCFrontendCoreView::Script Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::Script()  
2: {  
3:     raw(R"(<script src="https://code.jquery.com/jquery-3.5.1.min.js"  
integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin =  
"anonymous"></script>");  
4:     raw(R"(<script  
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"  
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"  
crossorigin="anonymous"></script>");  
5:     Core3DScript();  
6: }
```

C++

```
void Script() override;
```

Class

NCFrontendCoreView Class (🔗 see page 12)

Context

DefaultNamespace

Method

Script

Class

NCFrontendCoreView (🔗 see page 12)

1.1.4.3.6 NCFrontendCoreView::Style Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::Style()
2: {
3:     raw("<title>");raw(bag("Title"));raw("</title>");
4:     raw(R("<link rel='stylesheet'
href='https://pro.fontawesome.com/releases/v5.13.0/css/all.css'
integrity='sha384-IIED/eyOkM6ihtOiQsX2zizxFBphgnv1zbe1bKA+njdFzkr6cDNy16jfIKWu4FNH'
crossorigin='anonymous'>"));
5:     raw(R("<link rel='stylesheet'
href='https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css'
integrity='sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T'
crossorigin='anonymous'>"));
6: }
```

C++

```
void Style() override;
```

Class

NCFrontendCoreView Class (see page 12)

Context

DefaultNamespace

Method

Style

Class

NCFrontendCoreView (see page 12)

1.1.4.3.7 NCFrontendCoreView::Toolbar Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::Toolbar()
2: {
3:     TemplateFrontendView::Toolbar();
4: }
```

C++

```
void Toolbar() override;
```

Class

NCFrontendCoreView Class (see page 12)

Context

DefaultNamespace

Method

Toolbar

Class

NCFrontendView (see page 20)

1.1.4.3.8 NCFrontendCoreView::callScene Method

C++

```
void callScene(const QString& url, qint32 delay = 0);
```

Description

This is callScene, a member of class NCFrontendCoreView.

Class

NCFrontendCoreView Class ([↗](#) see page 12)

1.1.4.3.9 NCFrontendCoreView::rowBrand Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendCoreView::rowBrand()
2: {
3:     Open(row());
4:     Open(col().Sm(12).Md(6).OffsetMd(3).Mt(5).Mb(5));
5:     Open(h5().Style("font-weight:900").Class("my-0", "mx-0", "text-center",
"display-4"));
6:     Open(span().Class("fa-stack", "text-primary"));
7:     i().Pad().Icon("fa-circle").Class("fa-stack-2x");
8:     i().Pad().Icon("fa-cube").Class("fa", "fa-inverse", "fa-stack-1x");
9:     Close();
10:    raw(" #notioncube3d");
11:    Close();
12:    Close();
13:    Close();
14: }
```

C++

```
void rowBrand();
```

Class

NCFrontendCoreView Class ([↗](#) see page 12)

Context

DefaultNamespace

Method

RowBrand

Class

NCFrontendView ([↗](#) see page 20)

1.1.5 NCFrontendView Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

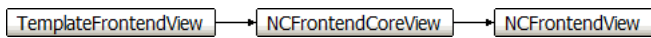
Date

12.05.2020

C++

```
class NCFrontendView : public NCFrontendCoreView;
```

Inheritance



Group

Classes (see page 1)

Legend

	Constructo r
	virtual

Members

Methods

	Name	Description
	NCFrontendCoreView (see page 13)	
	~NCFrontendCoreView (see page 14)	

NCFrontendView Class

	Name	Description
	NCFrontendView (see page 21)	
	~NCFrontendView (see page 22)	

NCFrontendCoreView Methods



	Name	Description
	Footer (see page 14)	
	Layout (see page 15)	
	Leftbar (see page 16)	
	Navigation (see page 16)	
	Script (see page 17)	
	Style (see page 17)	
	Toolbar (see page 18)	
	callScene (see page 19)	
	rowBrand (see page 19)	

Methods










	Name	Description
	NCFrontendCoreView (see page 13)	

	~NCFrontendCoreView (see page 14)	
---	---	--

NCFrontendView Class

	Name	Description
	NCFrontendView (see page 21)	
	~NCFrontendView (see page 22)	

NCFrontendCoreView Methods

	Name	Description
	Footer (see page 14)	
	Layout (see page 15)	
	Leftbar (see page 16)	
	Navigation (see page 16)	
	Script (see page 17)	
	Style (see page 17)	
	Toolbar (see page 18)	
	callScene (see page 19)	
	rowBrand (see page 19)	

Context

DefaultNamespace

Method

NCFrontendView

Class

NCFrontendView

1.1.5.1 NCFrontendView::NCFrontendView Constructor

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: NCFrontendView::NCFrontendView()
```

C++

```
NCFrontendView();
```

ClassNCFrontendView Class ([see page 20](#))**Context**

DefaultNamespace

Method

NCFrontendView

Class

NCFrontendView

1.1.5.2 NCFrontendView::~~NCFrontendView Destructor

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: NCFrontendView::~~NCFrontendView()=
```

C++

```
virtual ~NCFrontendView();
```

ClassNCFrontendView Class ([see page 20](#))**Context**

DefaultNamespace

Method~NCFrontendView ([see page 20](#))**Class**NCFrontendView ([see page 20](#))

1.1.6 NCFrontendViewFull Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date



12.05.2020

C++

```
class NCFrontendViewFull : public NCFrontendView;
```



Inheritance**Group**Classes ([see page 1](#))

Legend



	Constructo r
	virtual

Members



Methods

	Name	Description
	NCFrontendCoreView (see page 13)	
	~NCFrontendCoreView (see page 14)	










NCFrontendView Class

	Name	Description
	NCFrontendView (see page 21)	
	~NCFrontendView (see page 22)	





NCFrontendViewFull Class

	Name	Description
	NCFrontendViewFull (see page 24)	
	~NCFrontendViewFull (see page 25)	

NCFrontendCoreView Methods

	Name	Description
	Footer (see page 14)	
	Layout (see page 15)	
	Leftbar (see page 16)	
	Navigation (see page 16)	
	Script (see page 17)	
	Style (see page 17)	
	Toolbar (see page 18)	
	callScene (see page 19)	
	rowBrand (see page 19)	



NCFrontendViewFull Class

	Name	Description
	Footer (see page 25)	
	Layout (see page 26)	
	Navigation (see page 26)	
	Toolbar (see page 27)	

Methods

	Name	Description
	NCFrontendCoreView (see page 13)	
	~NCFrontendCoreView (see page 14)	

NCFrontendView Class

	Name	Description
	NCFrontendView (see page 21)	
	~NCFrontendView (see page 22)	

NCFrontendViewFull Class

	Name	Description
◆	NCFrontendViewFull (see page 24)	
◆	~NCFrontendViewFull (see page 25)	

NCFrontendCoreView Methods

	Name	Description
◆	Footer (see page 14)	
◆	Layout (see page 15)	
◆	Leftbar (see page 16)	
◆	Navigation (see page 16)	
◆	Script (see page 17)	
◆	Style (see page 17)	
◆	Toolbar (see page 18)	
◆	callScene (see page 19)	
◆	rowBrand (see page 19)	

NCFrontendViewFull Class

	Name	Description
◆	Footer (see page 25)	
◆	Layout (see page 26)	
◆	Navigation (see page 26)	
◆	Toolbar (see page 27)	

Context

DefaultNamespace

Method

NCFrontendViewFull

Class

NCFrontendView (see page 20)

1.1.6.1 NCFrontendViewFull::NCFrontendViewFull Constructor

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: NCFrontendViewFull::NCFrontendViewFull()
```

C++

```
NCFrontendViewFull();
```

Class

NCFrontendViewFull Class (see page 22)

Context

DefaultNamespace

Method

NCFrontendView ([↗](#) see page 20)

Class

NCFrontendView ([↗](#) see page 20)

1.1.6.2 NCFrontendViewFull::~~NCFrontendViewFull Destructor

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: NCFrontendViewFull::~~NCFrontendViewFull()=
```

C++

```
virtual ~NCFrontendViewFull();
```

Class

NCFrontendViewFull Class ([↗](#) see page 22)

Context

DefaultNamespace

Method

~NCFrontendView ([↗](#) see page 20)

Class

NCFrontendView ([↗](#) see page 20)

1.1.6.3 NCFrontendViewFull Methods

1.1.6.3.1 NCFrontendViewFull::Footer Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendViewFull::Footer()  
2: {
```

```
3: }
```

C++

```
void Footer() override;
```

Class

NCFrontendViewFull Class ([see page 22](#))

Context

DefaultNamespace

Method

Footer

Class

NCFrontendView ([see page 20](#))

1.1.6.3.2 NCFrontendViewFull::Layout Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendViewFull::Layout()  
2: {  
3:     Scope("Content");  
4:     raw("<style> body { overflow:hidden; } </style>");  
5: }
```

C++

```
void Layout() override;
```

Class

NCFrontendViewFull Class ([see page 22](#))

Context

DefaultNamespace

Method

Layout

Class

NCFrontendView ([see page 20](#))

1.1.6.3.3 NCFrontendViewFull::Navigation Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendViewFull::Navigation()  
2: {  
3: }
```

C++

```
void Navigation() override;
```

ClassNCFrontendViewFull Class ([↗](#) see page 22)**Context**

DefaultNamespace

Method

Navigation

ClassNCFrontendView ([↗](#) see page 20)

1.1.6.3.4 NCFrontendViewFull::Toolbar Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

12.05.2020

Body Source

```
1: void NCFrontendViewFull::Toolbar()  
2: {  
3: }
```

C++

```
void Toolbar() override;
```

ClassNCFrontendViewFull Class ([↗](#) see page 22)**Context**

DefaultNamespace

Method

Toolbar

ClassNCFrontendView ([↗](#) see page 20)

1.1.7 NHomeController Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

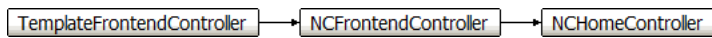
Date

12.05.2020

C++

```
class NHomeController : public NCFrontendController;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

NCFrontendController Methods

	Name	Description
	callScene (see page 10)	
	clearScene() (see page 10)	
	loadScene (see page 12)	

NCFrontendController Methods

	Name	Description
	callScene (see page 10)	
	clearScene() (see page 10)	
	loadScene (see page 12)	

Context

DefaultNamespace

Method

NHomeController

Class

NHomeController

1.1.8 PortalNC Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

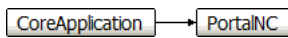
Date

12.05.2020

C++

```
class PortalNC : public CoreApplication;
```

Inheritance



Group

Classes ([see page 1](#))

Context

DefaultNamespace

Method

PortalNC

Class

PortalNC

1.2 Files

The following table lists files in this documentation.

Files

Name	Description
NCDiagramApiController.cpp (see page 30)	
NCDiagramApiController.h (see page 35)	
NCDiagramImageView.cpp (see page 36)	
NCDiagramImageView.h (see page 40)	
NCFrontendController.cpp (see page 41)	
NCFrontendController.h (see page 42)	
NCFrontendCoreView.cpp (see page 43)	
NCFrontendCoreView.h (see page 46)	
NCFrontendView.cpp (see page 47)	
NCFrontendView.h (see page 48)	
NCHomeController.cpp (see page 49)	
NCHomeController.h (see page 53)	
Portal.cpp (see page 54)	

PortalNC.cpp (see page 54)	
PortalNC.h (see page 55)	
main.cpp (see page 55)	

1.2.1 NCDiagramAPIController.cpp

Body Source

```

1: #include "NCDiagramAPIController.h"
2: #include <QDir>
3: //-----
4: // Author: Tobias Post
5: // Company: NOTIONCUBE Software GmbH
6: // Date: 12.06.2020
7: // Context: DefaultNamespace
8: // Class: NCDiagramAPIController
9: // Method: NCDiagramAPIController
10: //-----
11:
12: NCDiagramAPIController::NCDiagramAPIController():TemplateWebController()
13: {
14: }
15:
16: //-----
17: // Author: Tobias Post
18: // Company: NOTIONCUBE Software GmbH
19: // Date: 12.06.2020
20: // Context: DefaultNamespace
21: // Class: NCDiagramAPIController
22: // Method: ~NCDiagramAPIController
23: //-----
24:
25: NCDiagramAPIController::~NCDiagramAPIController()= default;
26:
27: //-----
28: // Author: Tobias Post
29: // Company: NOTIONCUBE Software GmbH
30: // Date: 12.06.2020
31: // Context: DefaultNamespace
32: // Class: NCDiagramAPIController
33: // Method: DiagramCreate
34: // Params: const QString& apikey
35: // Params: const QString& diagramid
36: //-----
37: //
38: http://localhost/NCDiagramAPI/DiagramCreate?apikey=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX&diag
39: ramid=SmallProject
40: void NCDiagramAPIController::DiagramCreate(const QString& apikey, const QString&
41: diagramid)
42: {
43:     if (checkAPI(apikey))
44:     {
45:         QDir().mkpath(diagramPath(apikey, diagramid));
46:         setContent(diagramid);
47:         return;
48:     }
49:     setContent("");
50: }
51: //-----
52: // Author: Tobias Post
53: // Company: NOTIONCUBE Software GmbH
54: // Date: 12.06.2020
55: // Context: DefaultNamespace

```

```

56: // Class: NCDiagramAPIController
57: // Method: DiagramDelete
58: // Params: const QString& apikey
59: // Params: const QString& diagramid
60: //-----
61: //
62: http://localhost/NCDiagramAPI/DiagramDelete?apikey=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX&diag
ramid=SmallProject
63: void NCDiagramAPIController::DiagramDelete(const QString& apikey, const QString&
diagramid)
64: {
65:     if (checkAPI(apikey))
66:     {
67:         const QDir dir(diagramPath(apikey, diagramid));
68:         for (const QFileInfo& fileinfo : dir.entryInfoList(QDir::NoDotAndDotDot |
QDir::System | QDir::Hidden | QDir::Files))
69:         {
70:             QFile::remove(fileinfo.absoluteFilePath());
71:         }
72:         QDir().rmdir(diagramPath(apikey, diagramid));
73:         setContent(diagramid);
74:         return;
75:     }
76:     setContent("");
77: }
78:
79: //-----
80: // Author: Tobias Post
81: // Company: NOTIONCUBE Software GmbH
82: // Date: 12.06.2020
83: // Context: DefaultNamespace
84: // Class: NCDiagramAPIController
85: // Method: FileWritePost
86: // Params: const QString& apikey
87: // Params: const QString& diagramid
88: //-----
89:
90: void NCDiagramAPIController::FileWritePost(const QString& apikey, const QString&
diagramid)
91: {
92:     if (checkAPI(apikey))
93:     {
94:         const QString path = diagramPath(apikey, diagramid);
95:         if (!QDir().exists(path)) QDir().mkpath(path);
96:         const QStringList files = FileRequest(path);
97:         setContent(files.join(';'));
98:         return;
99:     }
100:     setContent("");
101: }
102:
103: //-----
104: // Author: Tobias Post
105: // Company: NOTIONCUBE Software GmbH
106: // Date: 12.06.2020
107: // Context: DefaultNamespace
108: // Class: NCDiagramAPIController
109: // Method: FileWrite
110: // Params: const QString& apikey
111: // Params: const QString& diagramid
112: // Params: const QString& filename
113: // Params: const QString& content
114: //-----
115: //
116: //
117: http://localhost/NCDiagramAPI/FileWrite?apikey=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX&diagrami
d=SmallProject&filename=hello.txt&content=SGFsbG8KRgllcyBpc3QgZWluIFRlc3Q=
118:
119: void NCDiagramAPIController::FileWrite(const QString& apikey, const QString&
diagramid, const QString& filename, const QString& content)

```

```
120: {
121:     if (checkAPI(apikey))
122:     {
123:         const QString path = diagramPath(apikey, diagramid);
124:         if (!QDir().exists(path)) QDir().mkpath(path);
125:
126:         const QByteArray uncompressarray(QByteArray::fromBase64(content.toLatin1()));
127:         QFile file(diagramPath(apikey, diagramid, filename));
128:         if (file.open(QIODevice::WriteOnly))
129:         {
130:             file.write(uncompressarray);
131:         }
132:         setContent(filename);
133:         return;
134:     }
135:     setContent("");
136: }
137:
138: //-----
139: // Author: Tobias Post
140: // Company: NOTIONCUBE Software GmbH
141: // Date: 12.06.2020
142: // Context: DefaultNamespace
143: // Class: NCDiagramAPIController
144: // Method: FileDelete
145: // Params: const QString& apikey
146: // Params: const QString& diagramid
147: // Params: const QString& filename
148: //-----
149:
150: void NCDiagramAPIController::FileDelete(const QString& apikey, const QString&
diagramid, const QString& filename)
151: {
152:     if (checkAPI(apikey))
153:     {
154:         QFile::remove(diagramPath(apikey, diagramid, filename));
155:         setContent(filename);
156:         return;
157:     }
158:     setContent("");
159: }
160:
161: //-----
162: // Author: Tobias Post
163: // Company: NOTIONCUBE Software GmbH
164: // Date: 17.07.2020
165: // Context: DefaultNamespace
166: // Class: NCDiagramAPIController
167: // Method: FilePutBase64Post
168: // Params: QString apikey
169: // Params: const QString& diagramid
170: // Params: const QString& filename
171: //-----
172:
173: void NCDiagramAPIController::FilePutBase64Post(const QString& apikey, const QString&
diagramid, const QString& filename)
174: {
175:     if (checkAPI(apikey))
176:     {
177:         const QString path = diagramPath(apikey, diagramid);
178:         if (!QDir().exists(path)) QDir().mkpath(path);
179:
180:         QFile file(diagramPath(apikey, diagramid, filename));
181:         if (file.open(QIODevice::WriteOnly))
182:         {
183:             file.write(QByteArray::fromBase64(request()->Body()));
184:         }
185:         setContent(filename);
186:         return;
187:     }
188: }
189:     setContent("");
```

```
190: }
191:
192: //-----
193: // Author: Tobias Post
194: // Company: NOTIONCUBE Software GmbH
195: // Date: 17.07.2020
196: // Context: DefaultNamespace
197: // Class: NCDiagramAPIController
198: // Method: FilePutPost
199: // Params: QString apikey
200: // Params: const QString& diagramid
201: // Params: const QString& filename
202: //-----
203:
204: void NCDiagramAPIController::FilePutPost(const QString& apikey, const QString&
diagramid, const QString& filename)
205: {
206:     if (checkAPI(apikey))
207:     {
208:         const QString path = diagramPath(apikey, diagramid);
209:         if (!QDir().exists(path)) QDir().mkpath(path);
210:
211:         QFile file(diagramPath(apikey, diagramid, filename));
212:         if (file.open(QIODevice::WriteOnly))
213:         {
214:             file.write(request()->Body());
215:         }
216:         setContent(filename);
217:         return;
218:     }
219:     setContent("");
220: }
221:
222: //-----
223: // Author: Tobias Post
224: // Company: NOTIONCUBE Software GmbH
225: // Date: 12.06.2020
226: // Context: DefaultNamespace
227: // Class: NCDiagramAPIController
228: // Method: DiagramList
229: // Params: const QString& apikey
230: //-----
231:
232: void NCDiagramAPIController::DiagramList(const QString& apikey)
233: {
234:     if (checkAPI(apikey))
235:     {
236:         QStringList dirs;
237:         const QDir dir(diagramPath(apikey));
238:         for (const QFileInfo& fileinfo : dir.entryInfoList(QDir::NoDotAndDotDot |
QDir::Dirs))
239:         {
240:             dirs.append(fileinfo.fileName());
241:         }
242:         setContent(dirs.join(';'));
243:     }
244: }
245:
246: //-----
247: // Author: Tobias Post
248: // Company: NOTIONCUBE Software GmbH
249: // Date: 12.06.2020
250: // Context: DefaultNamespace
251: // Class: NCDiagramAPIController
252: // Method: DiagramFiles
253: // Params: const QString& apikey
254: // Params: const QString& diagramid
255: //-----
256:
257: void NCDiagramAPIController::DiagramFiles(const QString& apikey, const QString&
diagramid)
258: {
```

```
259:     if (checkAPI(apikey))
260:     {
261:         QStringList files;
262:         const QDir dir(diagramPath(apikey, diagramid));
263:         for (const QFileInfo& fileinfo : dir.entryInfoList(QDir::NoDotAndDotDot |
QDir::Files))
264:         {
265:             files.append(fileinfo.fileName());
266:         }
267:         setContent(files.join(';'));
268:         return;
269:     }
270:     setContent("");
271: }
272:
273: //-----
274: // Author: Tobias Post
275: // Company: NOTIONCUBE Software GmbH
276: // Date: 12.06.2020
277: // Context: DefaultNamespace
278: // Class: NCDiagramAPIController
279: // Method: FileRead
280: // Params: const QString& apikey
281: // Params: const QString& diagramid
282: // Params: const QString& filename
283: //-----
284:
285: void NCDiagramAPIController::FileRead(const QString& apikey, const QString& diagramid,
const QString& filename)
286: {
287:     setFile(diagramPath(apikey, diagramid,filename));
288: }
289:
290:
291: //-----
292: // Author: Tobias Post
293: // Company: NOTIONCUBE Software GmbH
294: // Date: 12.06.2020
295: // Context: DefaultNamespace
296: // Class: NCDiagramAPIController
297: // Method: checkAPI
298: // Params: const QString& apikey
299: // Params:
300: //-----
301:
302: bool NCDiagramAPIController::checkAPI(const QString& apikey)
303: {
304:     if (QDir(".").exists(diagramPath(apikey)))
305:         return true;
306:     //if (apikey == "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX") return true;
307:     return false;
308: }
309:
310: //-----
311: // Author: Tobias Post
312: // Company: NOTIONCUBE Software GmbH
313: // Date: 12.06.2020
314: // Context: DefaultNamespace
315: // Class: NCDiagramAPIController
316: // Method: checkDiagram
317: // Params: const QString& diagramid
318: // Params:
319: //-----
320:
321: bool NCDiagramAPIController::checkDiagram(const QString& diagramid)
322: {
323:     Q_UNUSED(diagramid);
324:     return true;
325: }
```

Description

This is file NCDiagramApiController.cpp.

Group

Files (see page 29)

1.2.2 NCDiagramApiController.h

Body Source


```
1: #pragma once
2:
3: #include <QObject>
4:
5: #include "Controller/TemplateWebController.h"
6:
7: //-----
8: // Author: Tobias Post
9: // Company: NOTIONCUBE Software GmbH
10: // Date: 12.06.2020
11: // Context: DefaultNamespace
12: // Class: NCDiagramApiController
13: // Method: NCDiagramApiController
14: //-----
15:
16: class NCDiagramApiController : public TemplateWebController
17: {
18:     Q_OBJECT
19:     CONTROLLER(NCDiagramApiController)
20:
21:     NCDiagramApiController();
22:     virtual ~NCDiagramApiController();
23:
24: public slots:
25:     virtual void DiagramCreate(const QString& apikey, const QString& diagramid);
26:     virtual void DiagramDelete(const QString& apikey, const QString& diagramid);
27:     virtual void FileWritePost(const QString& apikey, const QString& diagramid);
28:     virtual void FileWrite(const QString& apikey, const QString& diagramid, const
QString& filename, const QString& content);
29:     virtual void FileDelete(const QString& apikey, const QString& diagramid, const
QString& filename);
30:
31:     virtual void FilePutBase64Post(const QString& apikey, const QString& diagramid,
const QString& filename);
32:     virtual void FilePutPost(const QString& apikey, const QString& diagramid, const
QString& filename);
33:
34:     virtual void DiagramList(const QString& apikey);
35:     virtual void DiagramFiles(const QString& apikey, const QString& diagramid);
36:     virtual void FileRead(const QString& apikey, const QString& diagramid, const
QString& filename);
37: private:
38:     bool checkAPI(const QString& apikey);
39:     bool checkDiagram(const QString& diagramid);
40:
41:     QString diagramPath(const QString& apikey, const QString& diagramid=QString(),
const QString& filename= QString())
42:     {
43:         if (diagramid.isEmpty())
44:             return "data//diagram//" + apikey;
45:         if (filename.isEmpty())
46:             return "data//diagram//" + apikey + "/" + diagramid;
47:
48:         return "data//diagram//" + apikey + "/" + diagramid+"/"+filename;
49:     }
```

```
50: };
51:
52:
```

Description

This is file NCDiagramApiController.h.


Classes

	Name	Description
	NCDiagramApiController (see page 1)	

Group

Files ([see page 29](#))

Legend

	Class
---	-------

1.2.3 NCDiagramImageView.cpp

Body Source

```
1: #include "NCDiagramImageView.h"
2:
3: #include "Tools/paint/paintImage.h"
4:
5:
6: //-----
7: // Author: Tobias Post
8: // Company: NOTIONCUBE Software GmbH
9: // Date: 06.09.2020
10: // Context: DefaultNamespace
11: // Class: NCDiagramImageView
12: // Method: NCDiagramImageView
13: // Params: const QString& content
14: // Params: const qint32 width
15: // Params: const qint32 height
16: // Params: const QString& path
17: //-----
18:
19: NCDiagramImageView::NCDiagramImageView(const QString& content, const qint32 width,
const qint32 height, const QString& path): ImageView(width, height)
20: {
21:     if (imageResult())
22:         paintImage::clear(*imageResult()->image());
23:
24:
25:     m_Path = path;
26:     QStringList lines = content.split('\n', Qt::SkipEmptyParts);
27:     for (const QString& line : lines)
28:     {
29:         QStringList val = line.split(':');
30:         if (val.count() == 2)
31:         {
32:             m_Strings[val[0]] = val[1].trimmed();
33:             m_Values[val[0]] = val[1].trimmed().remove('%').toInt();
34:         }
35:     }
36: }
37: //-----
38: // Author: Tobias Post
39: // Company: NOTIONCUBE Software GmbH
40: // Date: 06.09.2020
41: // Context: DefaultNamespace
42: // Class: NCDiagramImageView
43: // Method: drawRing
```



```
44: //-----
45:
46: void NCDiagramImageView::drawRing()
47: {
48:     const qint32 outer = valInt("RingOuter");
49:     const qint32 inner = valInt("RingInner");
50:
51:     const qint32 w2 = width() / 2;
52:     const qint32 val = w2 - w2 * outer / 100;
53:     const qint32 val2 = w2 - w2 * inner / 100;
54:
55:     if (outer || inner)
56:     {
57:         QPainterPath path;
58:         if (outer) path.addEllipse(val, val, width() - val - val, height() - val -
val);
59:         if (inner) path.addEllipse(val2, val2, width() - val2 - val2, height() - val2
- val2);
60:
61:         painter().setPen(QPen(QColor(255, 255, 255), 8));
62:         painter().setBrush(QBrush(QColor(128, 128, 128, 64)));
63:         painter().drawPath(path);
64:     }
65: }
66:
67: //-----
68: // Author: Tobias Post
69: // Company: NOTIONCUBE Software GmbH
70: // Date: 06.09.2020
71: // Context: DefaultNamespace
72: // Class: NCDiagramImageView
73: // Method: drawBorder
74: //-----
75:
76: void NCDiagramImageView::drawBorder()
77: {
78:     painter().setPen(QPen(QColor(255, 255, 255), 8));
79:     painter().setBrush(Qt::NoBrush);
80:     painter().drawRect(QRect(8, 8, width() - 16, height() - 16));
81: }
82:
83: //-----
84: // Author: Tobias Post
85: // Company: NOTIONCUBE Software GmbH
86: // Date: 06.09.2020
87: // Context: DefaultNamespace
88: // Class: NCDiagramImageView
89: // Method: drawTitle
90: //-----
91:
92: void NCDiagramImageView::drawTitle()
93: {
94:     const QString title = val("Title");
95:     if (!title.isEmpty())
96:     {
97:         painter().save();
98:         painter().setFont(QFont("Helvetica", 80));
99:         painter().drawText(QRectF(0, 20, width(), 128), Qt::AlignCenter, title);
100:        painter().restore();
101:    }
102: }
103:
104: //-----
105: // Author: Tobias Post
106: // Company: NOTIONCUBE Software GmbH
107: // Date: 06.09.2020
108: // Context: DefaultNamespace
109: // Class: NCDiagramImageView
110: // Method: drawValue
111: //-----
112:
113: void NCDiagramImageView::drawValue()
```

```
114: {
115:     const QString value = val("Value");
116:     if (!value.isEmpty())
117:     {
118:         painter().save();
119:         painter().setFont(QFont("Helvetica", 4 * 60, QFont::Bold));
120:         painter().drawText(QRectF(0, 128, width(), 128 + 4 * 60), Qt::AlignCenter,
value);
121:         painter().restore();
122:     }
123: }
124:
125: //-----
126: // Author: Tobias Post
127: // Company: NOTIONCUBE Software GmbH
128: // Date: 06.09.2020
129: // Context: DefaultNamespace
130: // Class: NCDiagramImageView
131: // Method: drawValues
132: //-----
133:
134: void NCDiagramImageView::drawValues()
135: {
136:     const quint32 stepx = (width() - 16) / 10;
137:
138:     drawVal(valInt("Val1"), 0, stepx);
139:     drawVal(valInt("Val2"), 1, stepx);
140:     drawVal(valInt("Val3"), 2, stepx);
141:     drawVal(valInt("Val4"), 3, stepx);
142:     drawVal(valInt("Val5"), 4, stepx);
143:     drawVal(valInt("Val6"), 5, stepx);
144:     drawVal(valInt("Val7"), 6, stepx);
145:     drawVal(valInt("Val8"), 7, stepx);
146:     drawVal(valInt("Val9"), 8, stepx);
147:     drawVal(valInt("Val10"), 9, stepx);
148: }
149:
150: //-----
151: // Author: Tobias Post
152: // Company: NOTIONCUBE Software GmbH
153: // Date: 06.09.2020
154: // Context: DefaultNamespace
155: // Class: NCDiagramImageView
156: // Method: drawVal
157: // Params: const quint32 val
158: // Params: const quint32 i
159: // Params: const quint32 stepx
160: //-----
161:
162: void NCDiagramImageView::drawVal(const quint32 val, const quint32 i, const quint32 stepx)
163: {
164:     if (val == 0)
165:         return;
166:     //quint32 value = val*(height()/6)/100;
167:     painter().setPen(QPen(QColor(255, 255, 255), 2));
168:     painter().setBrush(QColor(255, 255, 255, 128));
169:     painter().drawRect(16 + i * stepx, height() - stepx - stepx - stepx - 8 - val,
stepx - 8, stepx - 8 + val);
170: }
171:
172: //-----
173: // Author: Tobias Post
174: // Company: NOTIONCUBE Software GmbH
175: // Date: 06.09.2020
176: // Context: DefaultNamespace
177: // Class: NCDiagramImageView
178: // Method: drawStatus
179: //-----
180:
181: void NCDiagramImageView::drawStatus()
182: {
183:     const quint32 blue = valInt("StatusBlue");
```

```

184:     const qint32 green = valInt("StatusGreen");
185:     const qint32 yellow = valInt("StatusYellow");
186:     const qint32 red = valInt("StatusRed");
187:
188:     if (blue || green || yellow || red)
189:     {
190:         const qint32 stepx = (width() - 16) / 10;
191:         for (qint32 pos = 0; pos < 10; pos++)
192:         {
193:             drawStatus(pos, stepx, blue, green, yellow, red);
194:         }
195:     }
196: }
197:
198: //-----
199: // Author: Tobias Post
200: // Company: NOTIONCUBE Software GmbH
201: // Date: 06.09.2020
202: // Context: DefaultNamespace
203: // Class: NCDiagramImageView
204: // Method: drawStatus
205: // Params: const qint32 pos
206: // Params: const qint32 stepx
207: // Params: const qint32 blue
208: // Params: const qint32 green
209: // Params: const qint32 yellow
210: // Params: const qint32 red
211: //-----
212:
213: void NCDiagramImageView::drawStatus(const qint32 pos, const qint32 stepx, const qint32
blue, const qint32 green, const qint32 yellow, const qint32 red)
214: {
215:     if (pos < blue)
216:         painter().setBrush(QColor(0, 128, 255, 255));
217:     else if (pos < blue + green)
218:         painter().setBrush(QColor(128, 255, 0, 255));
219:     else if (pos < blue + green + yellow)
220:         painter().setBrush(QColor(255, 128, 0, 255));
221:     else if (pos < blue + green + yellow + red)
222:         painter().setBrush(QColor(255, 0, 0, 255));
223:     else
224:         painter().setBrush(QColor(255, 255, 255, 32));
225:     painter().drawRect(16 + pos * stepx, height() - stepx - 8, stepx - 8, stepx - 8);
226: }
227:
228: //-----
229: // Author: Tobias Post
230: // Company: NOTIONCUBE Software GmbH
231: // Date: 06.09.2020
232: // Context: DefaultNamespace
233: // Class: NCDiagramImageView
234: // Method: drawImage
235: // Params: const bool overflow
236: //-----
237:
238: void NCDiagramImageView::drawImage(const bool overflow)
239: {
240:     const QString imagefilename = val(overflow ? "Overflow" : "Image");
241:
242:     if (!imagefilename.isEmpty())
243:     {
244:         const QImage image(m_Path + imagefilename);
245:
246:         const QImage scaledimage = image.scaled(width(), height(),
Qt::KeepAspectRatio, Qt::SmoothTransformation);
247:         {
248:             const qint32 x = width() / 2 - scaledimage.width() / 2;
249:             const qint32 y = height() / 2 - scaledimage.height() / 2;
250:             painter().save();
251:             painter().setRenderHint(QPainter::SmoothPixmapTransform);
252:             painter().setRenderHint(QPainter::Antialiasing);
253:             painter().drawImage(QRect(x, y, scaledimage.width(),

```

```

scaledimage.height()), scaledimage);
254:     painter().restore();
255:     }
256: }
257: }
258:
259: //-----
260: // Author: Tobias Post
261: // Company: NOTIONCUBE Software GmbH
262: // Date: 06.09.2020
263: // Context: DefaultNamespace
264: // Class: NCDiagramImageView
265: // Method: Paint
266: //-----
267:
268: void NCDiagramImageView::Paint()
269: {
270:     O_STACK(Img, Paint);
271:     painter().setRenderHint(QPainter::Antialiasing);
272:
273:     drawImage();
274:     drawRing();
275:     drawBorder();
276:     drawValues();
277:     drawStatus();
278:     drawTitle();
279:     drawValue();
280:     drawImage(true);
281: }

```

Description

This is file NCDiagramImageView.cpp.

Group

Files (see page 29)

1.2.4 NCDiagramImageView.h

Body Source

```

1: #pragma once
2: #include <QPainterPath>
3:
4: #include "View/ImageView.h"
5:
6: //-----
7: // Author: Tobias Post
8: // Company: NOTIONCUBE Software GmbH
9: // Date: 09.06.2020
10: // Context: DefaultNamespace
11: // Class: NCHomeController
12: // Method: ImagePNG
13: // Params: QString id
14: //-----
15:
16: class NCDiagramImageView final :public ImageView
17: {
18: public:
19:
20:     NCDiagramImageView(const QString& content, qint32 width, qint32 height, const
QString& path);
21:
22:     QString val(const QString& name)
23:     {
24:         return m_Strings[name];
25:     }

```

```

26:     qint32 valInt(const QString& name)
27:     {
28:         return m_Values[name];
29:     }
30:
31:     void drawRing();
32:     void drawBorder();
33:     void drawTitle();
34:     void drawValue();
35:     void drawValues();
36:     void drawVal(qint32 val, qint32 i, qint32 stepx);
37:     void drawStatus();
38:     void drawStatus(qint32 pos, qint32 stepx, qint32 blue, qint32 green, qint32 yellow,
qint32 red);
39:     void drawImage(bool overflow = false);
40:
41:     void Paint() override;
42:
43: private:
44:     QMap<QString, QString> m_Strings;
45:     QMap<QString, qint32> m_Values;
46:     QString m_Path;
47: };

```

Description

This is file NCDiagramImageView.h.


Classes

	Name	Description
	NCDiagramImageView (↗ see page 2)	

Group

Files ([↗](#) see page 29)

Legend

	Class
---	-------

1.2.5 NCFrontendController.cpp

Body Source

```

1: #include "NCFrontendController.h"
2: #include "NCFrontendView.h"
3: #include "Core/CoreService.h"
4: //-----
5: // Author: Tobias Post
6: // Company: NOTIONCUBE Software GmbH
7: // Date: 12.05.2020
8: // Context: DefaultNamespace
9: // Class: NCFrontendController
10: // Method: NCFrontendController
11: //-----
12: NCFrontendController::NCFrontendController()=default;
13: //-----
14: // Author: Tobias Post
15: // Company: NOTIONCUBE Software GmbH
16: // Date: 12.05.2020
17: // Context: DefaultNamespace
18: // Class: NCFrontendController
19: // Method: ~NCFrontendController
20: //-----
21: NCFrontendController::~NCFrontendController()=default;
22: //-----
23: // Author: Tobias Post
24: // Company: NOTIONCUBE Software GmbH

```

```

25: // Date: 18.05.2020
26: // Context: DefaultNamespace
27: // Class: NCFrontendController
28: // Method: callScene
29: // Params: QString url
30: // Params: qint32 delay
31: //-----
32: QString NCFrontendController::callScene(const QString & url, const qint32 delay)
33: {
34:     return Core3DCanvas::callScene(url, delay);
35: }
36: //-----
37: // Author: Tobias Post
38: // Company: NOTIONCUBE Software GmbH
39: // Date: 18.05.2020
40: // Context: DefaultNamespace
41: // Class: NCFrontendController
42: // Method: loadScene
43: // Params: QString name
44: //-----
45: QString NCFrontendController::loadScene(const QString& name, const QVector3D pos) const
46: {
47:     return Core3DCanvas::loadScene(name, pos);
48: }
49: //-----
50: // Author: Tobias Post
51: // Company: NOTIONCUBE Software GmbH
52: // Date: 18.05.2020
53: // Context: DefaultNamespace
54: // Class: NCFrontendController
55: // Method: clearScene
56: // Params: QString name
57: //-----
58: QString NCFrontendController::clearScene(const QString& name) const
59: {
60:     return Core3DCanvas::clearScene(name);
61: }
62: //-----
63: // Author: Tobias Post
64: // Company: NOTIONCUBE Software GmbH
65: // Date: 18.05.2020
66: // Context: DefaultNamespace
67: // Class: NCFrontendController
68: // Method: clearScene
69: //-----
70: QString NCFrontendController::clearScene() const
71: {
72:     return Core3DCanvas::clearScene();
73: }

```

Description

This is file NCFrontendController.cpp.

Group

Files (see page 29)

1.2.6 NCFrontendController.h

Body Source

```

1: #pragma once
2:
3: #include "Controller/TemplateFrontendController.h"
4: #include <QString>
5: #include <QVector3D>
6:

```

```


7: //-----
8: // Author: Tobias Post
9: // Company: NOTIONCUBE Software GmbH
10: // Date: 12.05.2020
11: // Context: DefaultNamespace
12: // Class: NCFrontendController
13: // Method: NCFrontendController
14: //-----
15: class NCFrontendController : public TemplateFrontendController
16: {
17:     Q_OBJECT
18: public:
19:     NCFrontendController();
20:     virtual ~NCFrontendController();
21:
22:     QString currentRole() const override
23:     {
24:         return "";
25:     }
26:
27:     void loginRedirect() override
28:     {
29:     }
30: protected:
31:     QString callScene(const QString& url, qint32 delay);
32:     QString loadScene(const QString& name, QVector3D pos=QVector3D()) const;
33:     QString clearScene(const QString& name) const;
34:     QString clearScene() const;
35: private:
36:     QString m_Directory;
37: };

```

Description

This is file NCFrontendController.h.


Classes

	Name	Description
	NCFrontendController (see page 9)	

Group

Files ([see page 29](#))

Legend

	Class
---	-------

1.2.7 NCFrontendCoreView.cpp

Body Source

```

1: #include "NCFrontendView.h"
2: #include "Controller/CoreController.h"
3: //-----
4: // Author: Tobias Post
5: // Company: NOTIONCUBE Software GmbH
6: // Date: 12.05.2020
7: // Context: DefaultNamespace
8: // Class: NCFrontendCoreView
9: // Method: NCFrontendCoreView
10: //-----
11: NCFrontendCoreView::NCFrontendCoreView()= default;
12: //-----
13: // Author: Tobias Post
14: // Company: NOTIONCUBE Software GmbH
15: // Date: 12.05.2020
16: // Context: DefaultNamespace

```

```
17: // Class:   NCFrontendView
18: // Method:  ~NCFrontendView
19: //-----
20: NCFrontendCoreView::~NCFrontendCoreView()= default;
21: //-----
22: // Author:  Tobias Post
23: // Company: NOTIONCUBE Software GmbH
24: // Date:    12.05.2020
25: // Context: DefaultNamespace
26: // Class:   NCFrontendView
27: // Method:  Layout
28: //-----
29: void NCFrontendCoreView::Layout()
30: {
31:     //LayoutFrontend();
32:
33:     LayoutHeader(true, true);
34:
35:     Open(containerfluid());
36:     Scope("Content");
37:     Close();
38:
39:     LayoutFooter(false);
40: }
41: //-----
42: // Author:  Tobias Post
43: // Company: NOTIONCUBE Software GmbH
44: // Date:    12.05.2020
45: // Context: DefaultNamespace
46: // Class:   NCFrontendView
47: // Method:  Navigation
48: //-----
49: void NCFrontendCoreView::Navigation()
50: {
51:     FullNavigation("#notioncube3d", "fa-cube");
52: }
53: //-----
54: // Author:  Tobias Post
55: // Company: NOTIONCUBE Software GmbH
56: // Date:    12.05.2020
57: // Context: DefaultNamespace
58: // Class:   NCFrontendView
59: // Method:  Toolbar
60: //-----
61: void NCFrontendCoreView::Toolbar()
62: {
63:     TemplateFrontendView::Toolbar();
64: }
65: //-----
66: // Author:  Tobias Post
67: // Company: NOTIONCUBE Software GmbH
68: // Date:    12.05.2020
69: // Context: DefaultNamespace
70: // Class:   NCFrontendCoreView
71: // Method:  Leftbar
72: //-----
73: void NCFrontendCoreView::Leftbar()
74: {
75:     TemplateFrontendView::Leftbar();
76: }
77: //-----
78: // Author:  Tobias Post
79: // Company: NOTIONCUBE Software GmbH
80: // Date:    12.05.2020
81: // Context: DefaultNamespace
82: // Class:   NCFrontendView
83: // Method:  Footer
84: //-----
85: void NCFrontendCoreView::Footer()
86: {
87:     //TemplateFrontendView::Footer();
88:     Open(div().Py(2).Class("organismfooterlinks"));
```



```

89:         Open(footer().Class("footer", "text-center"));
90:
91:         Open(container().Py(2).Class("text-center"));
92:         Open(span().Class("fa-stack", "text-primary"));
93:         i().Fad().Icon("fa-circle").Class("fa-stack-2x");
94:         i().Fad().Icon("fa-cube").Class("fa", "fa-inverse", "fa-stack-1x");
95:         Close();
96:
97:         raw(" #notioncube3d (" + QString(__TIMESTAMP__) + ")");
98:
99:         Close();
100:        Close();
101:    Close();
102: }
103:
104: //-----
105: // Author: Tobias Post
106: // Company: NOTIONCUBE Software GmbH
107: // Date: 12.05.2020
108: // Context: DefaultNamespace
109: // Class: NCFrontendCoreView
110: // Method: Style
111: //-----
112:
113: void NCFrontendCoreView::Style()
114: {
115:     raw("<title>");raw(bag("Title"));raw("</title>");
116:     raw(R("<link rel='stylesheet"
href="https://pro.fontawesome.com/releases/v5.13.0/css/all.css"
integrity="sha384-IIED/eyOkM6ihtOiQsX2zizxFBphgnv1zbelbKA+njdFzkr6cDNy16jfkIKWu4FNH"
crossorigin="anonymous">")");
117:     raw(R("<link rel='stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxt2MZw1T"
crossorigin="anonymous" >")");
118: }
119:
120: //-----
121: // Author: Tobias Post
122: // Company: NOTIONCUBE Software GmbH
123: // Date: 12.05.2020
124: // Context: DefaultNamespace
125: // Class: NCFrontendCoreView
126: // Method: Script
127: //-----
128: void NCFrontendCoreView::Script()
129: {
130:     raw(R("<script src='https://code.jquery.com/jquery-3.5.1.min.js"
integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin =
"anonymous"></script>")");
131:     raw(R("<script
src='https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>")");
132:     Core3DScript();
133: }
134: //-----
135: // Author: Tobias Post
136: // Company: NOTIONCUBE Software GmbH
137: // Date: 12.05.2020
138: // Context: DefaultNamespace
139: // Class: NCFrontendView
140: // Method: RowBrand
141: //-----
142: void NCFrontendCoreView::rowBrand()
143: {
144:     Open(row());
145:     Open(col().Sm(12).Md(6).OffsetMd(3).Mt(5).Mb(5));
146:     Open(h5().Style("font-weight:900").Class("my-0", "mx-0", "text-center",
"display-4"));
147:     Open(span().Class("fa-stack", "text-primary"));
148:     i().Fad().Icon("fa-circle").Class("fa-stack-2x");

```

```

149:             i().Fad().Icon("fa-cube").Class("fa", "fa-inverse", "fa-stack-1x");
150:             Close();
151:             raw(" #notioncube3d");
152:             Close();
153:             Close();
154:             Close();
155: }
156:
157: //-----
158: // Author: Tobias Post
159: // Company: NOTIONCUBE Software GmbH
160: // Date: 18.05.2020
161: // Context: DefaultNamespace
162: // Class: NCFrontendCoreView
163: // Method: callScene
164: // Params: QString url
165: // Params: qint32 delay
166: //-----
167:
168: void NCFrontendCoreView::callScene(const QString& url, const qint32 delay)
169: {
170:     raw("<script>$(document).ready(function() {");
171:     if (delay)
172:         raw("setTimeout(function() {$.get(\""+url+"\", function(data, status){
eval(data); }); }, " + QString("%1").arg(delay) + ");");
173:     else
174:         raw("$.get(\"" + url + "\", function(data, status){ eval(data); });");
175:
176:     raw("});</script>");
177: }
178:

```

Description

This is file NCFrontendCoreView.cpp.

Group

Files (see page 29)

1.2.8 NCFrontendCoreView.h

Body Source

```

1: #pragma once
2:
3: #include "View/TemplateFrontendView.h"
4: //-----
5: // Author: Tobias Post
6: // Company: NOTIONCUBE Software GmbH
7: // Date: 12.05.2020
8: // Context: DefaultNamespace
9: // Class: NCFrontendView
10: // Method: NCFrontendView
11: //-----
12: class NCFrontendCoreView : public TemplateFrontendView
13: {
14: public:
15:     NCFrontendCoreView();
16:     virtual ~NCFrontendCoreView();
17:
18:     void Layout() override;
19:     void Navigation() override;
20:     void Toolbar() override;
21:     void Leftbar() override;
22:     void Footer() override;
23:     void Style() override;
24:     void Script() override;

```

```

25:
26:     void rowBrand();
27:
28:     void callScene(const QString& url, qint32 delay=0);
29: };
30:
31:

```

Description

This is file NCFrontendCoreView.h.

Classes

	Name	Description
	NCFrontendCoreView (see page 12)	

Group

Files [\(see page 29\)](#)

Legend

	Class
---	-------

1.2.9 NCFrontendView.cpp

Body Source

```

1: #include "NCFrontendView.h"
2: //-----
3: // Author: Tobias Post
4: // Company: NOTIONCUBE Software GmbH
5: // Date: 12.05.2020
6: // Context: DefaultNamespace
7: // Class: NCFrontendView
8: // Method: NCFrontendView
9: //-----
10: NCFrontendView::NCFrontendView() = default;
11: //-----
12: // Author: Tobias Post
13: // Company: NOTIONCUBE Software GmbH
14: // Date: 12.05.2020
15: // Context: DefaultNamespace
16: // Class: NCFrontendView
17: // Method: ~NCFrontendView
18: //-----
19: NCFrontendView::~NCFrontendView()=default;
20:
21: //-----
22: // Author: Tobias Post
23: // Company: NOTIONCUBE Software GmbH
24: // Date: 12.05.2020
25: // Context: DefaultNamespace
26: // Class: NCFrontendView
27: // Method: NCFrontendView
28: //-----
29: NCFrontendViewFull::NCFrontendViewFull()= default;
30: //-----
31: // Author: Tobias Post
32: // Company: NOTIONCUBE Software GmbH
33: // Date: 12.05.2020
34: // Context: DefaultNamespace
35: // Class: NCFrontendView
36: // Method: ~NCFrontendView
37: //-----
38: NCFrontendViewFull::~NCFrontendViewFull()= default;
39:
40: //-----

```

```

41: // Author: Tobias Post
42: // Company: NOTIONCUBE Software GmbH
43: // Date: 12.05.2020
44: // Context: DefaultNamespace
45: // Class: NCFrontendView
46: // Method: Layout
47: //-----
48: void NCFrontendViewFull::Layout()
49: {
50:     Scope("Content");
51:     raw("<style> body { overflow:hidden; } </style>");
52: }
53: //-----
54: // Author: Tobias Post
55: // Company: NOTIONCUBE Software GmbH
56: // Date: 12.05.2020
57: // Context: DefaultNamespace
58: // Class: NCFrontendView
59: // Method: Navigation
60: //-----
61: void NCFrontendViewFull::Navigation()
62: {
63: }
64: //-----
65: // Author: Tobias Post
66: // Company: NOTIONCUBE Software GmbH
67: // Date: 12.05.2020
68: // Context: DefaultNamespace
69: // Class: NCFrontendView
70: // Method: Toolbar
71: //-----
72: void NCFrontendViewFull::Toolbar()
73: {
74: }
75: //-----
76: // Author: Tobias Post
77: // Company: NOTIONCUBE Software GmbH
78: // Date: 12.05.2020
79: // Context: DefaultNamespace
80: // Class: NCFrontendView
81: // Method: Footer
82: //-----
83: void NCFrontendViewFull::Footer()
84: {
85: }

```

Description

This is file NCFrontendView.cpp.

Group

Files (🗄️ see page 29)

1.2.10 NCFrontendView.h

Body Source

```

1: #pragma once
2:
3: #include "NCFrontendCoreView.h"
4: //-----
5: // Author: Tobias Post
6: // Company: NOTIONCUBE Software GmbH
7: // Date: 12.05.2020
8: // Context: DefaultNamespace
9: // Class: NCFrontendView
10: // Method: NCFrontendView

```

```


11: //-----
12: class NCFrontendView : public NCFrontendCoreView
13: {
14: public:
15:     NCFrontendView();
16:     virtual ~NCFrontendView();
17: };
18: //-----
19: // Author: Tobias Post
20: // Company: NOTIONCUBE Software GmbH
21: // Date: 12.05.2020
22: // Context: DefaultNamespace
23: // Class: NCFrontendView
24: // Method: NCFrontendViewFull
25: //-----
26: class NCFrontendViewFull : public NCFrontendView
27: {
28: public:
29:     NCFrontendViewFull();
30:     virtual ~NCFrontendViewFull();
31:     void Layout() override;
32:     void Navigation() override;
33:     void Toolbar() override;
34:     void Footer() override;
35: };

```

Description

This is file NCFrontendView.h.

Classes

	Name	Description
	NCFrontendView (see page 20)	
	NCFrontendViewFull (see page 22)	

Group

Files ([see page 29](#))

Legend

	Class
---	-------

1.2.11 NCHomeController.cpp

Body Source

```

1: #include "NCHomeController.h"
2: #include "NCFrontendView.h"
3: #include <QCoreApplication>
4:
5: #include "View/Core3DView.h"
6: #include "PortalNC.h"
7:
8: #include "NCDiagramImageView.h"
9: #include "Tools/Advanced3DExport.h"
10: #include "Tools/paint/paintCore.h"
11: #include "View/ImageView.h"
12:
13: //-----
14: // Author: Tobias Post
15: // Company: NOTIONCUBE Software GmbH
16: // Date: 12.05.2020
17: // Context: DefaultNamespace
18: // Class: NCHomeController
19: // Method: NCHomeController
20: //-----

```

```

21: NCHomeController::NCHomeController()=default;
22: //-----
23: // Author: Tobias Post
24: // Company: NOTIONCUBE Software GmbH
25: // Date: 12.05.2020
26: // Context: DefaultNamespace
27: // Class: NCHomeController
28: // Method: ~NCHomeController
29: //-----
30: NCHomeController::~NCHomeController()=default;
31: //-----
32: // Author: Tobias Post
33: // Company: NOTIONCUBE Software GmbH
34: // Date: 12.05.2020
35: // Context: DefaultNamespace
36: // Class: NCHomeController
37: // Method: Index
38: //-----
39: void NCHomeController::Index()
40: {
41:     setCulture("DE");
42:     LOCALVIEW(NCFrontendView)
43:     {
44:         rowBrand();
45:     }
46:     LOCALVIEWEND
47: }
48: //-----
49: // Author: Tobias Post
50: // Company: NOTIONCUBE Software GmbH
51: // Date: 12.06.2020
52: // Context: DefaultNamespace
53: // Class: NCHomeController
54: // Method: View
55: // Params: const QString& apikey
56: // Params: const QString& diagramid
57: // Params: const QString& filename
58: //-----
59:
//http://localhost/NCHome/View?apikey=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX&diagramid=SmallPr
object&filename=test570
60: //http://localhost/View/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/SmallProject/test570
61: //
62: void NCHomeController::View(const QString&apikey, const QString& diagramid, const
QString& filename)
63: {
64:     O_CRITICAL("RESTARTLOG");
65:
66:     setCulture("DE");
67:
68:     setBag("apikey",apikey);
69:     setBag("diagramid",diagramid);
70:     setBag("filename",filename);
71:
72:     LOCALVIEW(NCFrontendViewFull)
73:     {
74:         canvas3D().Orbit().Ground().Fog().Light().Design("#background-theme
#foreground-theme #theme-sandstone ").FullScreen();
75:         eventControls3D();
76:
callScene(QString("/NCHome/Open?apikey=%1&diagramid=%2&filename=%3").arg(bag("apikey")).arg(
bag("diagramid")).arg(bag("filename")))
);
77:     }
78:     LOCALVIEWEND
79: }
80:
81: //-----
82: // Author: Tobias Post
83: // Company: NOTIONCUBE Software GmbH
84: // Date: 12.06.2020
85: // Context: DefaultNamespace

```

```

86: // Class:   NCHomeController
87: // Method:  Open
88: // Params:  const QString& apikey
89: // Params:  const QString& diagramid
90: // Params:  const QString& filename
91: //-----
92:
93: void NCHomeController::Open(const QString& apikey, const QString& diagramid, const
QString& filename)
94: {
95:     O_PARAM(bag("apikey"));
96:
97:     QFile file("data//diagram//"+apikey+"//"+diagramid+"//" + filename);
98:     if (file.open(QIODevice::ReadOnly))
99:     {
100:         Advanced3DExport scene;
101:         setContent(scene.createScene3DJS(file.readAll()));
102:     }
103: }
104:
105: //-----
106: // Author:   Tobias Post
107: // Company:  NOTIONCUBE Software GmbH
108: // Date:     10.06.2020
109: // Context:  DefaultNamespace
110: // Class:    NCHomeController
111: // Method:   ImagePNG
112: // Params:   QString boardsize
113: //-----
114:
115: void NCHomeController::ImagePNG(const QString& apikey, const QString& diagramid, const
QString& filename, const QString& boardsize)
116: {
117:
118:     QFile file("data/diagram/"+ apikey + "/" + diagramid + "/" + filename);
119:     if (file.open(QIODevice::ReadOnly))
120:     {
121:         qint32 width = 1024;
122:         qint32 height = 1024;
123:         Advanced3DExport::calcDimension(boardsize, width, height);
124:
125:         setImage(new NCDiagramImageView(file.readAll(),width, height, "data/diagram/"
+ apikey + "/" + diagramid + "/"));
126:     }
127: }
128: }
129: //-----
130: // Author:   Tobias Post
131: // Company:  NOTIONCUBE Software GmbH
132: // Date:     18.05.2020
133: // Context:  DefaultNamespace
134: // Class:    NCHomeController
135: // Method:   Content
136: //-----
137:
138: void NCHomeController::Content(const QString& id)
139: {
140:     Q_UNUSED(id);
141:     QString content;
142:     content += "#theme-docs #striped #box2x2 #sided \n";
143:     content += "Root #layout-manhattan #hidden\n";
144:
145:     srand(QDateTime::currentSecsSinceEpoch());
146:
147:     const qint32 m = rand() % 2;
148:
149:     const QString apikey = "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX";
150:     const QString diagramid = "SmallProject";
151:     const QString filename="test.ncobject";
152:
153:     const QString call = "/ImagePNG/" + apikey + "/" + diagramid + "/" + filename +
"/";

```

```

154:
155:     for (qint32 i = 0; i < 1+m; i++)
156:     {
157:         //QString id = QString("%1").arg(i);
158:         content += "    A #layout-stack #sided #color7\n";
159:         content += "        A1 #color7\n";
160:         if (rand()%3==0) content += "        A1 #tower2x2 #color8\n";
161:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
162:         if (rand() % 5 == 0) content += "        A1 #color8 #icon-car\n";
163:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
164:         if (rand() % 5 == 0) content += "        A1 #tower2x2 #color5\n";
165:         if (rand() % 5 == 0) content += "        A1 #color5\n";
166:         if (rand() % 5 == 0) content += "        X #invisible\n";
167:         if (rand() % 2 == 0) content += "        X board:"+call+"XLXL boardsize:XLXL
planefrontback:" + call + "XLXL \n";
168:         content += "    B #layout-stack #sided #color7\n";
169:         if (rand() % 5 == 0) content += "        A1 #color7 planeleftright:" + call +
"XLXL \n";
170:         if (rand() % 5 == 0) content += "        A1 #tower2x2 #color8 planefrontback:"
+ call + "XLXL \n";
171:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
172:         if (rand() % 5 == 0) content += "        A1 #color8 #icon-car
#modify-cylinder\n";
173:         if (rand() % 5 == 0) content += "        A1 #tower2x2 #color5 planeleftright:"
+ call + "XLXL \n";
174:         if (rand() % 5 == 0) content += "        A1 #color5 planeleftright:" + call +
"XLXL \n";
175:         if (rand() % 5 == 0) content += "        X #invisible\n";
176:         if (rand() % 2 == 0) content += "        X +++ board:" + call + "MDXL
boardsize:MDXL planeleftright:" + call + "XLXL \n";
177:         content += "    C #theme-cube #layout-stack #sided #color7\n";
178:         if (rand() % 5 == 0) content += "        A1 #color7\n";
179:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
180:         if (rand() % 5 == 0) content += "        A1 #tower2x2 #color8 #invisible\n";
181:         if (rand() % 5 == 0) content += "        A1 #color8 #icon-car planefrontback:"
+ call + "XLXL \n";
182:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
183:         if (rand() % 5 == 0) content += "        A1 #tower1x2 #color5
#modify-cylinder\n";
184:         if (rand() % 5 == 0) content += "        A1 #tower2x3 #color8 planefrontback:"
+ call + "XLXL \n";
185:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
186:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
187:         if (rand() % 5 == 0) content += "        A1 #color8 #icon-car\n";
188:         if (rand() % 5 == 0) content += "        A1 #tower2x3 #color5
#modify-cylinder\n";
189:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
190:         if (rand() % 5 == 0) content += "        A1 #color5 planeleftright:" + call +
"XLXL \n";
191:         if (rand() % 5 == 0) content += "        X #invisible\n";
192:         if (rand() % 2 == 0) content += "        X ++ board:" + call + "XLMD
boardsize:XLMD planetop:" + call + "XLXL \n";
193:         content += "    D #layout-stack #sided #color7\n";
194:         if (rand() % 5 == 0) content += "        A1 #color7\n";
195:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
196:         if (rand() % 5 == 0) content += "        A1 #tower2x2 #color8
#modify-cylinder\n";
197:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
198:         if (rand() % 5 == 0) content += "        A1 #color8 #icon-car planefrontback:"
+ call + "XLXL \n";
199:         if (rand() % 5 == 0) content += "        A1 #tower1x2 #color5\n";
200:         if (rand() % 5 == 0) content += "        A1 #invisible\n";
201:         if (rand() % 5 == 0) content += "        A1 #color5\n";
202:         content += "    Z call:/NCHome/Content?id=YY #tower1x3 #color9 \n";
203:     }
204:
205:     const qint32 v = rand() % 1000;
206:
207:     QFile file("data/diagram/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX/SmallProject/" +
QString("test%1").arg(v));
208:     if (file.open(QIODevice::WriteOnly))
209:         file.write(content.toLatin1());

```



```

210:
211:     Advanced3DExport scene;
212:     const QString content3js = scene.createScene3DJS(content);
213:
214:     setContent(content3js);
215: }

```

Description

This is file NCHomeController.cpp.

Group

Files ([see page 29](#))

1.2.12 NCHomeController.h

Body Source

```

1: #pragma once
2:
3: #include <qabstractaspect.h>
4: #include "NCFrontendController.h"
5:
6: -----
7: // Author: Tobias Post
8: // Company: NOTIONCUBE Software GmbH
9: // Date: 12.05.2020
10: // Context: DefaultNamespace
11: // Class: NCHomeController
12: // Method: NCHomeController
13: -----
14: class NCHomeController : public NCFrontendController
15: {
16:     Q_OBJECT
17:     CONTROLLER(NCHomeController)
18:
19:     NCHomeController();
20:     ~NCHomeController();
21: public slots:
22:     void Index();
23:     void View(const QString& apikey, const QString& diagramid, const QString& filename);
24:     void Open(const QString& apikey, const QString& diagramid, const QString& filename);
25:     void ImagePNG(const QString& apikey, const QString& diagramid, const QString&
filename, const QString& boardsize);
26:     void Content(const QString& id);
27: };
28:

```

Description

This is file NCHomeController.h.

Classes

	Name	Description
	NCHomeController (see page 28)	

Group

Files ([see page 29](#))

Legend

	Class
---	-------

1.2.13 Portal.cpp

This is file Portal.cpp.

Group

Files (🔗 see page 29)

1.2.14 PortalNC.cpp

Body Source

```

1: #include "PortalNC.h"
2:
3: #include "NCDiagramAPIController.h"
4: #include "NCHomeController.h"
5:
6: //-----
7: // Author: Tobias Post
8: // Company: NOTIONCUBE Software GmbH
9: // Date: 12.05.2020
10: // Context: WebServer
11: // Class: PortalBasic
12: // Method: PortalBasic
13: //-----
14: PortalNC::PortalNC()=default;
15: //-----
16: // Author: Tobias Post
17: // Company: NOTIONCUBE Software GmbH
18: // Date: 12.05.2020
19: // Context: WebServer
20: // Class: PortalBasic
21: // Method: ~PortalBasic
22: //-----
23: PortalNC::~PortalNC()=default;
24: //-----
25: // Author: Tobias Post
26: // Company: NOTIONCUBE Software GmbH
27: // Date: 12.05.2020
28: // Context: DefaultNamespace
29: // Class: PortalBasic
30: // Method: Register
31: //-----
32: void PortalNC::Register()
33: {
34:     O_STACK(PortalNC, Register);
35:     NCHomeController::registerController();
36:     NCDiagramAPIController::registerController();
37: }
38: //-----
39: // Author: Tobias Post
40: // Company: NOTIONCUBE Software GmbH
41: // Date: 12.05.2020
42: // Context: Frontend.Shop.Controller
43: // Class: PortalBasic
44: // Method: Routing
45: //-----
46: void PortalNC::Routing()
47: {
48:     O_STACK(PortalNC, Routing);
49:     AutoRoute("NC");
50: }

```

Description

This is file PortalNC.cpp.

Group

Files (🔗 see page 29)

1.2.15 PortalNC.h

Body Source

```

1: #pragma once
2:
3: #include "Controller/CoreApplication.h"
4: #include <qfuture.h>
5: //-----
6: // Author: Tobias Post
7: // Company: NOTIONCUBE Software GmbH
8: // Date: 12.05.2020
9: // Context: DefaultNamespace
10: // Class: PortalNC
11: // Method: PortalNC
12: //-----
13: class PortalNC : public CoreApplication
14: {
15:     Q_OBJECT
16: public:
17:     PortalNC();
18:     virtual ~PortalNC();
19: public slots:
20:     void Register() override;
21:     void Routing() override;
22: };

```

Description

This is file PortalNC.h.


Classes

	Name	Description
	PortalNC (🔗 see page 29)	

Group

Files (🔗 see page 29)

Legend

	Class
---	-------

1.2.16 main.cpp

Body Source

```

1:
2: #include "WebServer.h"
3: #include "PortalNC.h"
4: #include <QApplication>
5: #include <QCommandLineParser>
6:
7: //-----
8: // Author: Tobias Post
9: // Company: NOTIONCUBE Software GmbH

```

```
10: // Date: 12.05.2020
11: // Context: DefaultNamespace
12: // Class: main
13: // Method: main
14: // Params: int argc
15: // Params: char* argv[]
16: //-----
17: int main(int argc, char *argv[])
18: {
19:     QApplication app(argc, argv);
20:
21:     oApplication::initLog();
22:
23:
24:     QCommandLineParser parser;
25:     parser.setApplicationDescription("PortalNC");
26:     const QCommandLineOption portOption(QStringList() << "p" << "port",
QCoreApplication::translate("main", "Webserver <port> to open."),
QCoreApplication::translate("main", "port"));
27:     parser.addOption(portOption);
28:     parser.process(app);
29:
30:     qint32 port = parser.value(portOption).toInt();
31:     if(port==0) port = 400;
32:     O_STACK(main,main);
33:     O_PARAM(port);
34:     WebServer webserver(new PortalNC(), app,"PortalNC","2.0");
35:     webserver.StartServer();
36:
37:     QApplication::exec();
38:
39: }
40:
```

Description

This is file main.cpp.

Group

Files ([🔗](#) see page 29)

Index

C

Classes 1

F

Files 29

N

NCDiagramAPIController 1

NCDiagramAPIController class 1

about NCDiagramAPIController class 1

NCDiagramAPIController.cpp 30

NCDiagramAPIController.h 35

NCDiagramImageView 2

NCDiagramImageView class 2

NCDiagramImageView 3

Paint 3

about NCDiagramImageView class 2

drawBorder 4

drawImage 5

drawRing 5

drawStatus 6

drawTitle 6

drawVal 7

drawValue 7

drawValues 8

val 9

valInt 9

NCDiagramImageView.cpp 36

NCDiagramImageView.h 40

NCDiagramImageView::NCDiagramImageView 3

NCDiagramImageView::Paint 3

NCDiagramImageView::drawBorder 4

NCDiagramImageView::drawImage 5

NCDiagramImageView::drawRing 5

NCDiagramImageView::drawStatus 6

NCDiagramImageView::drawTitle 6

NCDiagramImageView::drawVal 7

NCDiagramImageView::drawValue 7

NCDiagramImageView::drawValues 8

NCDiagramImageView::val 9

NCDiagramImageView::valInt 9

NCFrontendController 9

NCFrontendController class 9

about NCFrontendController class 9

callScene 10

clearScene 10, 11

loadScene 12

NCFrontendController.cpp 41

NCFrontendController.h 42

NCFrontendController::callScene 10

NCFrontendController::clearScene 10, 11

NCFrontendController::loadScene 12

NCFrontendCoreView 12

NCFrontendCoreView class 12

Footer 14

Layout 15

Leftbar 16

NCFrontendCoreView 13

Navigation 16

Script 17

Style 17

Toolbar 18

about NCFrontendCoreView class 12

callScene 19

rowBrand 19

~NCFrontendCoreView 14

NCFrontendCoreView.cpp 43

NCFrontendCoreView.h 46

NCFrontendCoreView::Footer 14

NCFrontendCoreView::Layout 15

NCFrontendCoreView::Leftbar 16

NCFrontendCoreView::NCFrontendCoreView 13

NCFrontendCoreView::Navigation 16

NCFrontendCoreView::Script 17

NCFrontendCoreView::Style 17

NCFrontendCoreView::Toolbar 18

NCFrontendCoreView::callScene 19

NCFrontendCoreView::rowBrand 19

NCFrontendCoreView::~~NCFrontendCoreView 14

NCFrontendView 20

NCFrontendView class 20

NCFrontendView 21

 about NCFrontendView class 20

 ~NCFrontendView 22

NCFrontendView.cpp 47

NCFrontendView.h 48

NCFrontendView::NCFrontendView 21

NCFrontendView::~~NCFrontendView 22

NCFrontendViewFull 22

NCFrontendViewFull class 22

 Footer 25

 Layout 26

 NCFrontendViewFull 24

 Navigation 26

 Toolbar 27

 about NCFrontendViewFull class 22

 ~NCFrontendViewFull 25

NCFrontendViewFull::Footer 25

NCFrontendViewFull::Layout 26

NCFrontendViewFull::NCFrontendViewFull 24

NCFrontendViewFull::Navigation 26

NCFrontendViewFull::Toolbar 27

NCFrontendViewFull::~~NCFrontendViewFull 25

NHomeController 28

NHomeController class 28

 about NHomeController class 28

NHomeController.cpp 49

NHomeController.h 53

P

Portal.cpp 54

PortalNC 29

PortalNC class 29

 about PortalNC class 29

PortalNC.cpp 54

PortalNC.h 55

M

main.cpp 55