Package 'MVCClass'

November 14, 2025

Title Model-View-Controller (MVC) Classes

Version 1.85.0

Author Elizabeth Whalen

Description Creates classes used in model-view-controller (MVC) design

Depends R (>= 2.1.0), methods

Maintainer Elizabeth Whalen <ewhalen@hsph.harvard.edu>

License LGPL

biocViews Visualization, Infrastructure, GraphAndNetwork

git_url https://git.bioconductor.org/packages/MVCClass

git_branch devel

git_last_commit 91675ba

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-13

Contents

`dfModel-class` *Class "dfModel": A class to represent a data frame model*

Description

dfModel is a class to represent a data frame model. This class inherits from the virtual class, gModel. An object of dfModel is responsible for storing and updating the data.

Objects from the Class

Objects can be created by calls of the form `new("dfModel", ...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

`modelData`**:** the model data, which is a data frame

`virtualData`**:** data that is needed by views of this model, will be a data frame

`linkData`**:** a list of functions that link this model to its parent and child models (if it has any)

`modelName`**:** the name of this model

`modelVar`**:** a list of variables that refer to the modelData (for instance this may be t-test values that were calculated from the modelData)

Extends

Class `"gModel"`, directly.

Methods

No methods defined with class "dfModel" in the signature. The methods for this class will be created in other packages that use this package like iSNetwork and iSPlot.

Author(s)

Elizabeth Whalen

See Also

gModel-class

---

gAddChildMessage-class

Class "gAddChildMessage": A class to represent an add child MVC message

---

Description

gAddChildMessage class represents an add child MVC message. This class will create a new MVC object that is a child of the current MVC object. gAddChildMessage inherits from the class, gAddMessage, which inherits from the virtual class, gMessage. The initialize and handleMessage methods will be defined in other packages that use this class (for example, the iSNetwork package).

Objects from the Class

Objects can be created by calls of the form `new("gAddChildMessage", ...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork package).

Slots

`dataName`: the name of the new child model (MVC)

`mData`: a list of information needed to add the new model

`type`: the type of the new model (for now, it may be a data.frame, ExpressionSet, or graph)

Extends

Class `"gAddMessage"`, directly. Class `"gMessage"`, by class `"gAddMessage"`.

Methods

No methods defined with class "gAddChildMessage" in the signature.

Author(s)

Elizabeth Whalen

See Also

gAddDataMessage-class, gAddMessage-class, gMessage-class

`gAddDataMessage-class` *Class "gAddDataMessage": A class to represent an add data message*

Description

gAddDataMessage is a class to represent an add data message. Whenever a model needs to be added, a gAddDataMessage object is created and the handleMessage method is called to act on the message. gAddDataMessage inherits from the virtual class, gAddMessage.

Objects from the Class

Objects can be created by calls of the form `new("gAddDataMessage",...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

`dataName`**:** the name of the model to be added

`mData`**:** a list of information that contains the data to be added

`type`**:** the type of model to be added (for now, it may be a data.frame, ExpressionSet, or graph)

Extends

Class `"gAddMessage"`, directly. Class `"gModifyMessage"`, by class `"gAddMessage"`. Class `"gMessage"`, by class `"gAddMessage"`.

Methods

No methods defined with class "gAddDataMessage" in the signature.

Author(s)

Elizabeth Whalen

See Also

`gAddViewMessage-class`, `gAddMessage-class`, `gMessage-class`

gAddMessage-class *Class "gAddMessage": A class to represent an add message*

Description

gAddMessage is a virtual class to represent an add message. Both gAddDataMessage and gAddViewMessage classes are inherited from gAddMessage.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`dataName`: the name of the data to be added or the name of the data that the view will visualize

`mData`: a list of information needed to perform the addition

`type`: the type of addition to perform (ex. which type of model to add or which type of view to add)

Extends

Class `"gModifyMessage"`, directly. Class `"gMessage"`, by class `"gModifyMessage"`.

Methods

No methods defined with class "gAddMessage" in the signature.

Author(s)

Elizabeth Whalen

See Also

`gMessage-class`, `gModifyMessage-class`, `gAddViewMessage-class`, `gAddDataMessage-class`, `gAddChildMessage-class`

gAddViewMessage-class *Class "gAddViewMessage": A class to represent an add view message*

Description

gAddViewMessage is a class to represent an add view message. Whenever a view needs to be added, a gAddViewMessage object is created and the handleMessage method is called to act on the message. gAddViewMessage inherits from the virtual class, gAddMessage.

Objects from the Class

Objects can be created by calls of the form `new("gAddViewMessage",...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

`dataName`: the name of the model to be added

`mData`: a list of information that contains the data needed to add the view

`type`: the type of view to be added (ex. scatterplot, spreadsheet, etc.)

Extends

Class `"gAddMessage"`, directly. Class `"gModifyMessage"`, by class `"gAddMessage"`. Class `"gMessage"`, by class `"gAddMessage"`.

Methods

No methods defined with class "gAddViewMessage" in the signature.

Author(s)

Elizabeth Whalen

See Also

gAddDataMessage-class, gAddMessage-class, gMessage-class

gAskAncestorMessage-class

Class "gAskAncestorMessage": A class to represent a question asking for data from an ancestor model

Description

gAskAncestorMessage is a class to ask for model data from an ancestor model (MVC). For example, if a user wants to create a child model, but needs data from both its parent and its grandparent model. Then when the parent model creates the new child model it would ask its parent (the grandparent of the new child model) for data using this class.

A specific example is if the original model contains microarray expression data and a child model of this model is the GO graph. Then if the user wants to look at the expression data of only genes annotated at a particular node, a child model can be created from the GO graph, but this new child model also needs data from the original model (its grandparent). Thus, when the GO graph creates a child model it needs to ask its parent for the expression data.

Objects from the Class

Objects can be created by calls of the form `new("gAskAncestorMessage", ...)`.

Slots

`type`: which model type to look for as an ancestor (this is a character string naming the class)

`mData`: the data that is being requested

`from`: the MVC that asked the question

Extends

Class `"gMessage"`, directly.

Methods

from<- Sets the `from` slot

from Returns the `from` slot

mData<- Sets the `mData` slot

mData Returns the `mData` slot

type<- Sets the `type` slot

type Returns the `type` slot

Note

Even though this class is used to ask for information from an ancestor, each model (MVC) has only parent. This is important when considering how data sets are linked.

Author(s)

Elizabeth Whalen

See Also

gMessage-class

---

genView-class *Class "genView": A virtual class to describe a view*

Description

genView is a virtual class that all view classes inherit from. All views will contain the information of what window (slot win) they are stored in, what data (slot dataName) is shown in the view, and the number of the window (slot winNum) that shows the view.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`dataName`: a character string describing what data are shown in the view

`win`: an object of class `"GtkWindow"` that holds the view

`winNum`: a number that tells what number view this is (for example, the first view created will have winNum=1)

Methods

dataName<- Sets the `dataName` slot

dataName Returns the `dataName` slot

win<- Sets the `win` slot

win Returns the `win` slot

winNum<- Sets the `winNum` slot

winNum Returns the `winNum` slot

Also, all view objects will have two methods, redrawView and updateView, that will define how a view should be redrawn when the underlying data has changed. The redrawView method will completely redraw the view, while the updateView method will only redraw the parts of the view that have changed. Both of these methods, redrawView and updateView, will be defined in packages that use this package, such as iSPlot and iSNetwork. Also, all views will have an identifyView method that will identify an object (such as a point, a node, a row, etc.) on the view given some location on the view (such as the user coordinates).

Author(s)

Elizabeth Whalen

See Also

plotView-class, sPlotView-class, spreadView-class

---

gEventFun-class *A class to link a callback function with an event*

Description

The gEventFun class creates an object which contains all the needed information to link a callback function to an event. The gEventFun object will hold all of the callback function information, including the callback function, a short description of what the callback function does, and a list of all preprocessing functions that must be called.

Objects from the Class

Objects can be created by calls of the form `new("gEventFun", ...)`.

Slots

`callFun`**:** the callback function

`shortName`**:** a short description of what the callback function does

`preprocessFun`**:** a list of preprocessing functions that must be called before the callback function (this can be NULL)

Methods

callFun<- Sets the `callFun` slot

callFun Returns the `callFun` slot

preprocessFun<- Sets the `preprocessFun` slot

preprocessFun Returns the `preprocessFun` slot

shortName<- Sets the `shortName` slot

shortName Returns the `preprocessFun` slot

Author(s)

Elizabeth Whalen

gMessage-class *Class "gMessage": A virtual class for messages*

Description

gMessage is a virtual class from which all other message classes will inherit. Message objects will be created whenever there is communication between the model, view, and controller components or between MVC objects.

Objects from the Class

A virtual Class: No objects may be created from it.

Methods

No methods defined with class "gMessage" in the signature. However, all message classes will have a handleMessage method that will read the message and notify the appropriate components so that the information in the message can be acted upon. The handleMessage methods will be defined in packages that use this package, such as iSPlot and iSNetwork.

Author(s)

Elizabeth Whalen

See Also

gUpdateMessage-class, gUpdateViewMessage-class, gUpdateDataMessage-class, gAddMessage-class, gAddViewMessage-class, gAddDataMessage-class, gAddChildMessage-class, gSendParentMessage-class, gSendChildMessage-class

gModel-class *Class "gModel": A virtual class for models*

Description

gModel is a virtual class from which all other model classes will inherit. Model objects will be responsible for storing and updating the data sets.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`modelData`: the actual model data

`linkData`: a list of the functions to link this data to its parent and child models (if it has any)

`virtualData`: data that pertains to the views of this model

`modelName`: the name of the model (will be the same name as the MVC)

`modelVar`: a list of variables that refer to the modelData (for instance this may be t-test values that were calculated from the modelData)

Methods

linkData<- Sets the `linkData` slot

linkData Returns the `linkData` slot

modelData<- Sets the `modelData` slot

modelData Returns the `modelData` slot

modelName<- Sets the `modelName` slot

modelName Returns the `modelName` slot

virtualData<- Sets the `virtualData` slot

virtualData Returns the `virtualData` slot

modelVar<- Sets the `modelVar` slot

modelVar Returns the `modelVar` slot

Also, all models will have an updateModel method that will be defined in the packages that use this package (for example, iSPlot and iSNetwork). The updateModel method will be called by a gUpdateDataMessage object when the data needs to be updated. Another method that may be defined for certain models is the provideInfo method that will return model information when it is asked for by a gAskAncestorMessage object.

Author(s)

Elizabeth Whalen

See Also

`dfModel-class`

`gModifyMessage-class` *Class "gModifyMessage": A class to represent a modify message*

Description

gModifyMessage is a virtual class to represent a modify message. Both gAddMessage and gUpdateMessage classes are inherited from gModifyMessage.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`dataName`**:** the name of the data to be added or the name of the data that the view will visualize

`mData`**:** a list of information needed to perform the addition

`type`**:** the type of addition to perform (ex. which type of model to add or which type of view to add)

Extends

Class `"gMessage"`, directly.

Methods

dataName<- Sets the `dataName` slot

dataName Returns the `dataName` slot

mData<- Sets the `mData` slot

mData Returns the `mData` slot

type<- Sets the `type` slot

type Returns the `type` slot

Author(s)

Elizabeth Whalen

See Also

`gMessage-class`, `gAddMessage-class`, `gUpdateMessage-class`

gSendChildMessage-class

Class "gSendChildMessage": A class to represent a send child MVC message

Description

gSendChildMessage is a class to represent a send child MVC message. Whenever a model is updated and that model has a child MVC, then an object of this class will be created to notify the child MVC that a parent model was updated. gSendChildMessage inherits from the virtual class, gMessage. The initialize and handleMessage methods will be defined in other packages that use this class (for example, the iSNetwork package).

Objects from the Class

Objects can be created by calls of the form `new("gSendChildMessage",...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork package).

Slots

`parentUpdateDataMessage`**:** an object of class gUpdateDataMessage that was used to update the parent model (MVC)

`childName`**:** the name of the child model that this message is directed towards (because a parent MVC can have multiple child MVCs)

Extends

Class `"gMessage"`, directly.

Methods

childName<- Sets the `childName` slot

childName Returns the `childName` slot

parentUpdateDataMessage<- Sets the `parentUpdateDataMessage` slot

parentUpdateDataMessage Returns the `parentUpdateDataMessage` slot

Author(s)

Elizabeth Whalen

See Also

`gSendParentMessage-class`, `gMessage-class`

gSendParentMessage-class *Class "gSendParentMessage": A class to represent a send parent MVC message*

Description

gSendParentMessage is a class to represent a send parent MVC message. Whenever a model is updated and that model has a parent MVC, then an object of this class will be created to notify the parent MVC that a child model was updated. gUpdateParentMessage inherits from the virtual class, gMessage. The initialize and handleMessage methods will be defined in other packages that use this class (for example, the iSNetwork package).

Objects from the Class

Objects can be created by calls of the form `new("gSendParentMessage",...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork package).

Slots

`childUpdateDataMessage`**:** an object of class gUpdateDataMessage that was used to update the child model (MVC)

Extends

Class `"gMessage"`, directly.

Methods

childUpdateDataMessage<- Sets the `childUpdateDataMessage` slot

childUpdateDataMessage Returns the `childUpdateDataMessage` slot

Author(s)

Elizabeth Whalen

See Also

`gSendChildMessage-class`, `gMessage-class`

gUpdateDataMessage-class *Class "gUpdateDataMessage": A class to represent an update data message*

Description

gUpdateDataMessage is a class to represent an update data message. gUpdateDataMessage is inherited from the virtual class, gUpdateMessage. Whenever a gUpdateDataMessage is created (initialized), the next step is to call the handleMessage method to act upon that message and update the data.

Objects from the Class

Objects can be created by calls of the form `new("gUpdateDataMessage",...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

`from`: the name of the MVC (model) that this update message came from

`type`: the type of update to perform

`mData`: a list of the information needed to perform the update

`dataName`: the name of the data set to be updated

Extends

Class `"gUpdateMessage"`, directly. Class `"gModifyMessage"`, by class `"gUpdateMessage"`. Class `"gMessage"`, by class `"gUpdateMessage"`.

Methods

from<- Sets the `from` slot

from Returns the `from` slot

Author(s)

Elizabeth Whalen

See Also

gUpdateViewMessage-class, gUpdateMessage-class, gMessage-class

---

`gUpdateMessage-class` *Class "gUpdateMessage": A class to represent an update message*

Description

gUpdateMessage is a virtual class to represent an update message. Both gUpdateViewMessage and gUpdateDataMessage classes are inherited from gUpdateMessage.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`type`: the type of update to be performed, will be a character string

`mData`: a list of information needed to perform the update

`dataName`: the name of the data set to be updated

Extends

Class `"gModifyMessage"`, directly. Class `"gMessage"`, by class `"gModifyMessage"`.

Methods

No methods defined with class "gUpdateMessage" in the signature.

Author(s)

Elizabeth Whalen

See Also

`gMessage-class`, `gModifyMessage-class`, `gUpdateViewMessage-class`, `gUpdateDataMessage-class`

`gUpdateViewMessage-class`
Class "gUpdateViewMessage": A class to represent an update view message

---

Description

gUpdateViewMessage is a class to represent an update view message. gUpdateViewMessage is inherited from the virtual class, gUpdateMessage. Whenever a gUpdateViewMessage is created (initialized), the next step is to call the handleMessage method to act upon that message and update the views.

Objects from the Class

Objects can be created by calls of the form `new("gUpdateViewMessage",...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

`type`: the type of update to be performed

`mData`: a list of the information needed to perform the update

`dataName`: the name of the data set that was updated (views are updated after the data set has been updated)

Extends

Class `"gUpdateMessage"`, directly. Class `"gModifyMessage"`, by class `"gUpdateMessage"`. Class `"gMessage"`, by class `"gUpdateMessage"`.

Methods

No methods defined with class "gUpdateViewMessage" in the signature.

Author(s)

Elizabeth Whalen

See Also

gUpdateDataMessage-class, gUpdateMessage-class, gMessage-class

---

linkedModelMVC-class *Class "linkedModelMVC": A class to represent a MVC object that can be linked to other MVC objects*

Description

linkedModelMVC is a class to represent a model-view-controller object that can be linked to other MVC objects. The linkedModelMVC class will combine these three components (model, view and controller) so that the MVC objects can be reused and it has the slots, parentMVC and childMVCList, to link this MVC object with other MVC objects.

Objects from the Class

Objects can be created by calls of the form `new("linkedModelMVC", ...)`.

Slots

`model`: the model object (will inherit from the gModel virtual class)

`viewList`: a list of the view objects that visualize the model

`controller`: the environment that stores information for the MVC object

`parentMVC`: the name of the parent model (MVC) if there is a parent model

`childMVCList`: a list of the names of the child models (MVCs); it may be an empty list if there are no children

Extends

Class `"singleModelMVC"`, directly. Class `"MVC"`, by class `"singleModelMVC"`.

Methods

childMVCList<- Sets the `childMVCList` slot

childMVCList Returns the `childMVCList` slot

parentMVC<- Sets the `parentMVC` slot

parentMVC Returns the `parentMVC` slot

Author(s)

Elizabeth Whalen

See Also

`gModel-class`, `genView-class`, `singleModelMVC-class`, `MVC-class`

`MVC-class` *Class "MVC": A virtual class to represent a model-view-controller object*

Description

MVC is a virtual class to represent a model-view-controller object. The MVC class will combine these three components (model, view and controller) so that the MVC objects can be reused and linked.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`model`: the model object (will inherit from the gModel virtual class)

`viewList`: a list of the view objects that visualize the model

`controller`: the environment that stores information for the MVC object

Methods

controller<- Sets the `controller` slot

controller Returns the `controller` slot

model<- Sets the `model` slot

model Returns the `model` slot

viewList<- Sets the `viewList` slot

viewList Returns the `viewList` slot

Author(s)

Elizabeth Whalen

See Also

`gModel-class`, `genView-class`, `singleModelMVC-class`, `linkedModelMVC-class`

`plotView-class` *Class "plotView": A virtual class to represent a plot view*

Description

plotView is a virtual class to represent a view that is a plot. Any particular types of plots can inherit from this class. For instance, sPlotView, is a class that inherits from plotView and represents a scatterplot view.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`plotDevice`**:** the plot device number

`plotPar`**:** the parameter list for the plot, see par()

`drArea`**:** an object of class `"GtkDrawingArea"`

`dataName`**:** a character string describing what data are shown in the view

`win`**:** an object of class `"GtkWindow"` that holds the view

`winNum`**:** a number that tells what number view this is (for example, the first view created will have winNum=1)

Extends

Class `"genView"`, directly.

Methods

drArea<- Sets the `drArea` slot

drArea Returns the `drArea` slot

plotDevice<- Sets the `plotDevice` slot

plotDevice Returns the `plotDevice` slot

plotPar<- Sets the `plotPar` slot

plotPar Returns the `plotPar` slot

Also, all classes that inherit from plotView will have clickEvent and motionEvent methods. The clickEvent method will be called whenever a user clicks on the plot and the motionEvent method will be called whenever a user moves the cursor over the plot. These methods will be defined in packages that use this package, such as iSPlot and iSNetwork. Note that the clickEvent method will also be defined for the spreadView class and this will correspond to a user selecting a row on the spreadsheet.

Author(s)

Elizabeth Whalen

See Also

`genView-class`, `sPlotView-class`

---

qqPlotView-class *Class "qqPlotView": A class to represent a qq-plot*

Description

qqPlotView is a class to represent a qq-plot. For now, this class will be used to create qq-plot views of gene set enrichment data (stored in the gseModel). qqPlotView inherits from the class, plotView, which inherits from the virtual class, genView.

Objects from the Class

Objects can be created by calls of the form `new("qqPlotView", ...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork package).

Slots

`xval`: the x values for the points plotted

`yval`: the y values for the points plotted

`plotDevice`: the plot device number

`plotPar`: the parameter list for the plot, see par()

`drArea`: an object of class `"GtkDrawingArea"`

`dataName`: a character string describing what data are shown in the view

`win`: an object of class `"GtkWindow"` that holds the view

`winNum`: a number that tells what number view this is (for example, the first view created will have winNum=1)

Extends

Class `"plotView"`, directly. Class `"genView"`, by class `"plotView"`.

Methods

xval<- Sets the `xval` slot

xval Returns the `xval` slot

yval<- Sets the `yval` slot

yval Returns the `yval` slot

Author(s)

Elizabeth Whalen

See Also

`genView-class`, `plotView-class`, `sPlotView-class`

`singleModelMVC-class` *Class "singleModelMVC": A class to represent a single model-view-controller object*

Description

singleModelMVC is a class to represent a model-view-controller object that cannot be linked to other MVC objects (i.e. single refers to the MVC object being unlinked). The singleModelMVC class will combine thee three components (model, view and controller) so that the MVC objects can be reused.

Objects from the Class

Objects can be created by calls of the form `new("singleModelMVC", ...)`.

Slots

`model`: the model object (will inherit from the gModel virtual class)

`viewList`: a list of the view objects that visualize the model

`controller`: the environment that stores information for the MVC object

Extends

Class "MVC", directly.

Methods

No methods defined with class "singleModelMVC" in the signature.

Author(s)

Elizabeth Whalen

See Also

gModel-class, genView-class, MVC-class, linkedModelMVC-class

---

sPlotView-class *Class "sPlotView": A class to represent a scatterplot view*

Description

sPlotView is a class to represent a view that is a scatterplot. sPlotView inherits from the class, plotView, which inherits from the virtual class, genView.

Objects from the Class

Objects can be created by calls of the form new("sPlotView", ...). The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

dfRows: the names of the data frame rows that will be plotted

xvar: the name of the data frame variable (column) that will be plotted on the x axis

yvar: the name of the data frame variable (column) that will be plotted on the y axis

plotDevice: the plot device number

plotPar: the parameter list for the plot, see par()

drArea: an object of class "GtkDrawingArea"

dataName: a character string describing what data are shown in the view

win: an object of class "GtkWindow" that holds the view

winNum: a number that tells what number view this is (for example, the first view created will have winNum=1)

Extends

Class "plotView", directly. Class "genView", by class "plotView".

Methods

xvar<- Sets the `xvar` slot

xvar Returns the `xvar` slot

yvar<- Sets the `yvar` slot

yvar Returns the `yvar` slot

dfRows<- Sets the `dfRows` slot

dfRows Returns the `dfRows` slot

Author(s)

Elizabeth Whalen

See Also

`genView-class`, `plotView-class`, `spreadView-class`

`spreadView-class` *Class "spreadView": A class to represent a spreadsheet view*

Description

spreadView is a class to represent a view that is a spreadsheet. spreadView inherits from the virtual class, genView.

Objects from the Class

Objects can be created by calls of the form `new("spreadView", ...)`. The initialize method for this class will be created in other packages that use this package (for example, the initialize method will be created in the iSNetwork and iSPlot packages).

Slots

`clist`**:** an object of class `"GtkCList"`, i.e. the spreadsheet

`dataName`**:** a character string describing what data are shown in the view

`win`**:** an object of class `"GtkWindow"` that holds the view

`winNum`**:** a number that tells what number view this is (for example, the first view created will have winNum=1)

Extends

Class `"genView"`, directly.

Methods

clist<- Sets the `clist` slot

clist Returns the `clist` slot

This class will also have a clickEvent method, which will be called whenever a user selects a row on the spreadsheet. This method will be defined in packages that use this package, such as iSPlot and iSNetwork.

Author(s)

Elizabeth Whalen

See Also

`genView-class`, `plotView-class`

Index