# Factories:

Code:

*\*\*  
 \* Constructs a {****@link*** *TreeTableCell} based on the view model of the row and a bunch of specified converter methods.  
 \*  
 \** ***@param*** <*S*> *view model  
 \*/*public class ViewModelTreeTableCellFactory<**S**> implements Callback<TreeTableColumn<**S, S**>**,** TreeTableCell<**S, S**>> {  
  
 private Callback<**S,** String> toText**;** private Callback<**S,** Node> toGraphic**;** private Callback<**S,** EventHandler<? super MouseEvent>> toOnMouseClickedEvent**;** private Callback<**S,** String> toTooltip**;** public ViewModelTreeTableCellFactory<**S**> withText(Callback<**S,** String> toText) {  
 this.toText = toText**;** return this**;** }  
  
 public ViewModelTreeTableCellFactory<**S**> withGraphic(Callback<**S,** Node> toGraphic) {  
 this.toGraphic = toGraphic**;** return this**;** }  
  
 public ViewModelTreeTableCellFactory<**S**> withIcon(Callback<**S,** JabRefIcon> toIcon) {  
 this.toGraphic = viewModel -> toIcon.call(viewModel).getGraphicNode()**;** return this**;** }  
  
 public ViewModelTreeTableCellFactory<**S**> withTooltip(Callback<**S,** String> toTooltip) {  
 this.toTooltip = toTooltip**;** return this**;** }  
  
 public ViewModelTreeTableCellFactory<**S**> withOnMouseClickedEvent(  
 Callback<**S,** EventHandler<? super MouseEvent>> toOnMouseClickedEvent) {  
 this.toOnMouseClickedEvent = toOnMouseClickedEvent**;** return this**;** }

Location:

src/main/java/org/jabref/gui/util/ViewModelTreeTableCellFactory.java

Reasoning:

Twins with src/main/java/org/jabref/gui/util/ViewModelTreeCellFactory.java

Code:

*/\*\*  
 \* Constructs a {****@link*** *TreeTableCell} based on the view model of the row and a bunch of specified converter methods.  
 \*  
 \** ***@param*** <*S*> *view model  
 \** ***@param*** <*T*> *cell value  
 \*/*public class ViewModelTreeCellFactory<**T**> implements Callback<TreeView<**T**>**,** TreeCell<**T**>> {  
  
 private Callback<**T,** String> toText**;** private Callback<**T,** Node> toGraphic**;** private Callback<**T,** EventHandler<? super MouseEvent>> toOnMouseClickedEvent**;** private Callback<**T,** String> toTooltip**;** public ViewModelTreeCellFactory<**T**> withText(Callback<**T,** String> toText) {  
 this.toText = toText**;** return this**;** }  
  
 public ViewModelTreeCellFactory<**T**> withGraphic(Callback<**T,** Node> toGraphic) {  
 this.toGraphic = toGraphic**;** return this**;** }  
  
 public ViewModelTreeCellFactory<**T**> withIcon(Callback<**T,** JabRefIcon> toIcon) {  
 this.toGraphic = viewModel -> toIcon.call(viewModel).getGraphicNode()**;** return this**;** }  
  
 public ViewModelTreeCellFactory<**T**> withTooltip(Callback<**T,** String> toTooltip) {  
 this.toTooltip = toTooltip**;** return this**;** }  
  
 public ViewModelTreeCellFactory<**T**> withOnMouseClickedEvent(Callback<**T,** EventHandler<? super MouseEvent>> toOnMouseClickedEvent) {  
 this.toOnMouseClickedEvent = toOnMouseClickedEvent**;** return this**;** }   
  
public void install(TreeView<**T**> treeView) {  
 treeView.setCellFactory(this)**;**}

Location:

src/main/java/org/jabref/gui/util/ViewModelTreeCellFactory.java

Reasoning:

Twins with src/main/java/org/jabref/gui/util/ViewModelTreeTableCellFactory.java

# Builder:

Code:

public static class Builder {  
  
 private final List<FileChooser.ExtensionFilter> extensionFilters = new ArrayList<>()**;** private Path initialDirectory**;** private FileChooser.ExtensionFilter defaultExtension**;** private String initialFileName**;** public FileDialogConfiguration build() {  
 return new FileDialogConfiguration(initialDirectory**,** extensionFilters**,** defaultExtension**,** initialFileName)**;** }  
  
 public Builder withInitialDirectory(Path directory) {  
 if (directory == null) { // It could be that somehow the path is null, for example if it got deleted in the meantime  
 initialDirectory = null**;** } else { // Dir must be a folder, not a file  
 if (!Files.*isDirectory*(directory)) {  
 directory = directory.getParent()**;** }  
 // The lines above work also if the dir does not exist at all!  
 // NULL is accepted by the filechooser as no inital path  
 // Explicit null check, if somehow the parent is null, as Files.exists throws an NPE otherwise  
 if ((directory != null) && !Files.*exists*(directory)) {  
 directory = null**;** }  
 initialDirectory = directory**;** }  
 return this**;** }  
  
 public Builder withInitialDirectory(String directory) {  
 if (directory != null) {  
 withInitialDirectory(Path.*of*(directory))**;** } else {  
 initialDirectory = null**;** }  
 return this**;** }  
  
 public Builder withInitialFileName(String initialFileName) {  
 this.initialFileName = initialFileName**;** return this**;** }  
  
 public Builder withDefaultExtension(FileChooser.ExtensionFilter extensionFilter) {  
 defaultExtension = extensionFilter**;** return this**;** }  
  
 public Builder withDefaultExtension(FileType fileType) {  
 defaultExtension = FileFilterConverter.*toExtensionFilter*(fileType)**;** return this**;** }  
  
 public Builder withDefaultExtension(String description**,** FileType fileType) {  
 defaultExtension = FileFilterConverter.*toExtensionFilter*(description**,** fileType)**;** return this**;** }  
  
 public Builder withDefaultExtension(String fileTypeDescription) {  
 extensionFilters.stream()  
 .filter(type -> type.getDescription().equalsIgnoreCase(fileTypeDescription))  
 .findFirst()  
 .ifPresent(extensionFilter -> defaultExtension = extensionFilter)**;** return this**;** }  
  
 public Builder addExtensionFilter(FileChooser.ExtensionFilter filter) {  
 extensionFilters.add(filter)**;** return this**;** }  
  
 public Builder addExtensionFilter(List<FileChooser.ExtensionFilter> filters) {  
 extensionFilters.addAll(filters)**;** return this**;** }  
  
 public Builder addExtensionFilter(FileType... fileTypes) {  
 Stream.*of*(fileTypes)  
 .map(FileFilterConverter::*toExtensionFilter*)  
 .forEachOrdered(this::addExtensionFilter)**;** return this**;** }  
  
 public Builder addExtensionFilter(String description**,** FileType fileType) {  
 extensionFilters.add(FileFilterConverter.*toExtensionFilter*(description**,** fileType))**;** return this**;** }  
 }  
}

Location:

src/main/java/org/jabref/gui/util/FileDialogConfiguration.java

Reasoning:

Multiple construction methods under same method