

Content Assist Template

```
import org.eclipse.jface.text.contentassist.IContentAssistant;
import org.eclipse.jface.text.contentassist.IContextInformation;
import org.eclipse.swt.graphics.Image;
import org.eclipse.jface.text.contentassist.IContentAssistProcessor;
import org.eclipse.jface.text.source.IAnnotationHover;
import org.eclipse.jface.text.contentassist.CompletionProposal;
import org.eclipse.jface.text.TextAttribute;
import org.eclipse.jface.text.contentassist.ContentAssistant;
import org.eclipse.swt.graphics.Color;
import org.eclipse.jface.text.IDocument;
import org.eclipse.jface.text.ITextViewer;
import org.eclipse.jface.text.IAutoEditStrategy;
import org.eclipse.jface.text.source.ISourceViewer;
import org.eclipse.jface.text.contentassist.ICompletionProposal;
import org.eclipse.jface.text.source.SourceViewerConfiguration;
import org.eclipse.jface.text.ITextHover;
import org.eclipse.jface.text.DocumentCommand;

public class AppContentAssistProcessor /* FRL_01 */
implements IContentAssistProcessor {

    public ICompletionProposal[] computeCompletionProposals(ITextViewer viewer, int offset) { /* FRL_02 */
    }

    public char[] getContextInformationAutoActivationCharacters() { /* FRL_03 */
    }

    public char[] getCompletionProposalAutoActivationCharacters() { /* FRL_04 */
    }

    public String getErrorMessage() { /* FRL_05 */
    }
}

public class AppSourceViewerConfiguration /* FRL_06 */
extends SourceViewerConfiguration {

    public String[] getIndentPrefixes (ISourceViewer viewer, String contentType) { /* FRL_07 */
    }

    public IAnnotationHover getAnnotationHover(ISourceViewer viewer) { /* FRL_08 */
        AppAnnotationHover appAnnotationHover = new AppAnnotationHover(); /* FRL_09 */
    }

    public int getTabWidth(ISourceViewer viewer) { /* FRL_10 */
    }

    public IContentAssistant getContentAssistant(ISourceViewer viewer) { /* FRL_11 */
        AppContentAssistProcessor appContentAssistProcessor = new AppContentAssistProcessor(); /* FRL_12 */
        String str = getConfiguredDocumentPartitioning(viewer); /* FRL_13 */
        ContentAssistant contentAssistant = new ContentAssistant(); /* FRL_14 */
        contentAssistant.setDocumentPartitioning(str); /* FRL_15 */
        contentAssistant.setContentAssistProcessor(appContentAssistProcessor); // MAY REPEAT! /* FRL_16 */
        contentAssistant.enableAutoActivation(boolean); /* FRL_17 */
        contentAssistant.setAutoActivationDelay(int); /* FRL_18 */
        contentAssistant.setProposalPopupOrientation(int); /* FRL_19 */
        contentAssistant.setContextInformationPopupOrientation(int); /* FRL_20 */
    }

    public ITextHover getTextHover(ISourceViewer viewer, String contentType)|| /* FRL_21 */
        (ISourceViewer viewer, String contentType, int stateMask) { /* FRL_22 */
        AppTextHover appTextHover = new AppTextHover();
    }

    public String getConfiguredDocumentPartitioning(ISourceViewer viewer) { /* FRL_23 */
    }

    public IAutoEditStrategy[] getAutoEditStrategies(ISourceViewer viewer, String contentType) { /* FRL_24 */
    }
}

public class AppTextHover /* FRL_25 */
implements ITextHover {
}

public class AppAnnotationHover /* FRL_26 */
implements IAnnotationHover {
}

public class SomeClass {

    public void someMethod() { /* FRL_27 */
        TextAttribute textAttribute = new TextAttribute(Color)|| (Color,Color,int); /* FRL_28 */
        int app_int = IDocument.getLineOffset(int); /* FRL_29 */
        AppSourceViewerConfiguration appSourceViewerConfiguration = new AppSourceViewerConfiguration(); /* FRL_30 */
        CompletionProposal completionProposal = new CompletionProposal(String,int,int,int)||
            (String,int,int,int,Image,String,IContextInformation,String);
    }
}
```