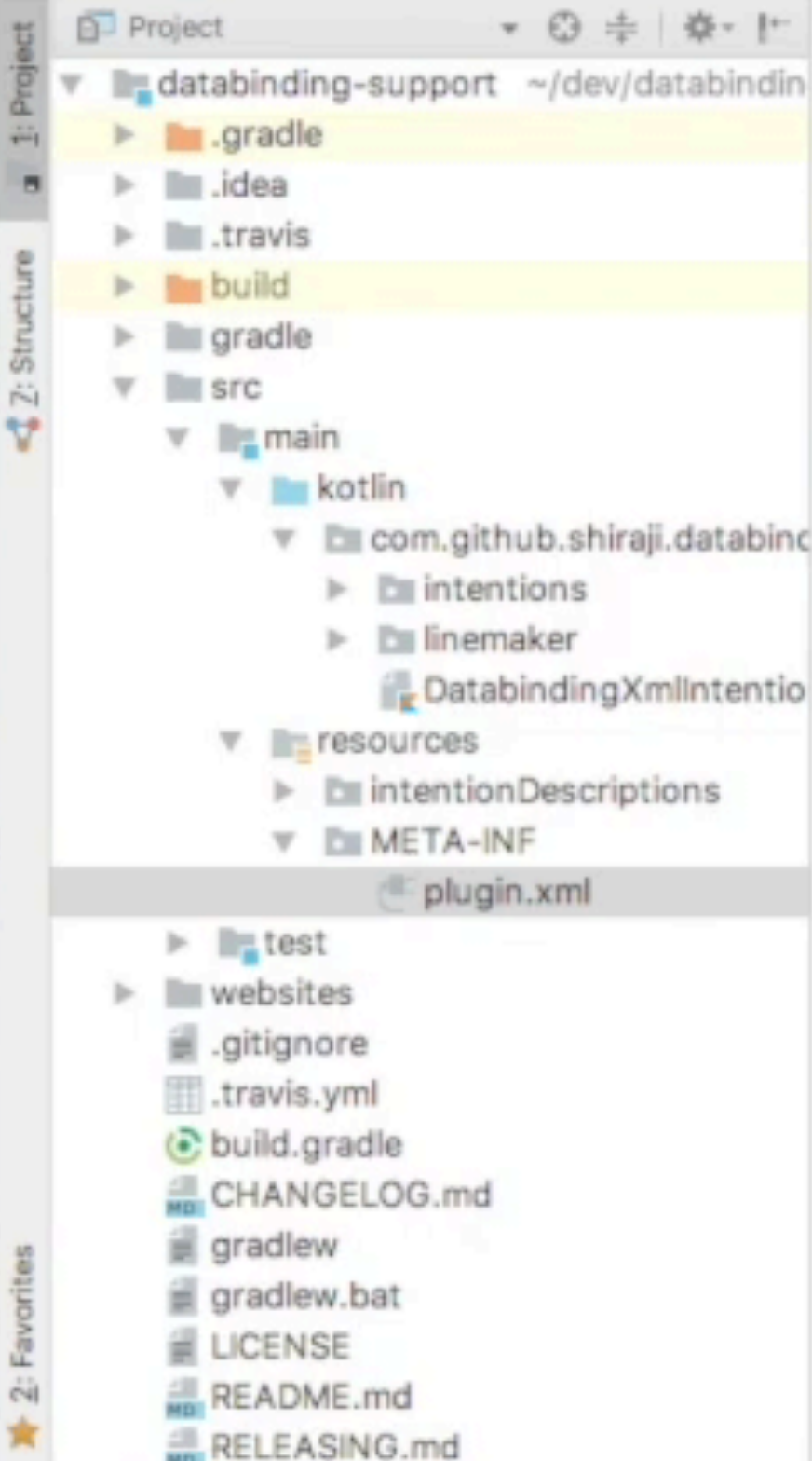


Implement Intention

```
override fun invoke(project: Project, editor: Editor?, file: PsiFile?) {  
    val rootTag = file.getRootTag() ?: return  
    val xmlnsAttribute = rootTag.attributes  
        .filter { it.name.startsWith("xmlns:") }  
    val attributeText = xmlnsAttribute  
        .map { it.text }  
        .joinToString(separator = " ")  
    val factory = XmlElementFactory.getInstance(project)  
  
    xmlnsAttribute.forEach { it.delete() }  
    val layoutXmlTag = factory.createTagFromText("<layout  
$attributeText>${rootTag.text}</layout>", XMLLanguage.INSTANCE)  
    rootTag.replace(layoutXmlTag)  
}
```

```
<layout xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:tools="http://schemas.android.com/tools"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
    <LinearLayout  
        android:orientation="vertical"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent">  
  
    </LinearLayout>  
</layout>
```



```
package com.github.shiraji.databindinglayout.intentions

import ...

class ConvertToDatabindingLayoutIntention : IntentionAction {

    companion object {
        val BLACK_LIST = listOf("manifest", "project", "component", "module", "activity")
    }

    override fun getText() = "Wrap with data binding layout"
    override fun getFamilyName() = "Wrap with data binding layout"
    override fun startInWriteAction() = true

    override fun isAvailable(project: Project, editor: Editor?, file: PsiFile?): Boolean {
        val rootTag = file.getRootTag() ?: return false
        if (rootTag.isDatabindingRootTag()) return false
        return !BLACK_LIST.contains(rootTag.name)
    }

    override fun invoke(project: Project, editor: Editor?, file: PsiFile?) {
        val rootTag = file.getRootTag() ?: return
        val xmlnsAttribute = rootTag.attributes
            .filter { it.name.startsWith("xmlns:") }
        val attributeText = xmlnsAttribute
            .map { it.text }
            .joinToString(separator = " ")
        val factory = XmlElementFactory.getInstance(project)

        xmlnsAttribute.forEach { it.delete() }
        val layoutXmlTag = factory.createTagFromText(p0: "<layout $attributeText")
        rootTag.replace(layoutXmlTag)
    }
}
```

ConvertToDatabindingLayoutIntention