

ConvertToDatabindingLayoutIntention.kt - databinding-support - [~/dev/databinding-support]

Project: databinding-support ~/dev/databindin

- .gradle
- .idea
- .travis
- build
- gradle
- src
 - main
 - kotlin
 - com.github.shiraji.databinc
 - intentions
 - linemaker
 - DatabindingXmlIntention
 - resources
 - intentionDescriptions
 - META-INF
- test
- websites
- .gitignore
- .travis.yml
- build.gradle
- CHANGELOG.md
- gradlew
- gradlew.bat
- LICENSE
- README.md
- RELEASING.md

plugin.xml

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

import ...

class ConvertToDatabindingLayoutIntention : IntentionAction {

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

    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?) {
        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

9: Version Control Breakpoints Manager Terminal 4: Run 6: TODO Event Log

ConvertToDatabindingLayoutIntention.kt - databinding-support - [~/dev/databinding-support]

Project: databinding-support ~/dev/databinding-support

- .gradle
- .idea
- .travis
- build
- gradle
- src
 - main
 - kotlin
 - com.github.shiraji.databinding-support
 - intentions
 - linemaker
 - DatabindingXmlIntention.kt
 - resources
 - intentionDescriptions
 - META-INF
- test
- websites
- .gitignore
- .travis.yml
- build.gradle
- CHANGELOG.md
- gradlew
- gradlew.bat
- LICENSE
- README.md
- RELEASING.md

plugin.xml

```
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

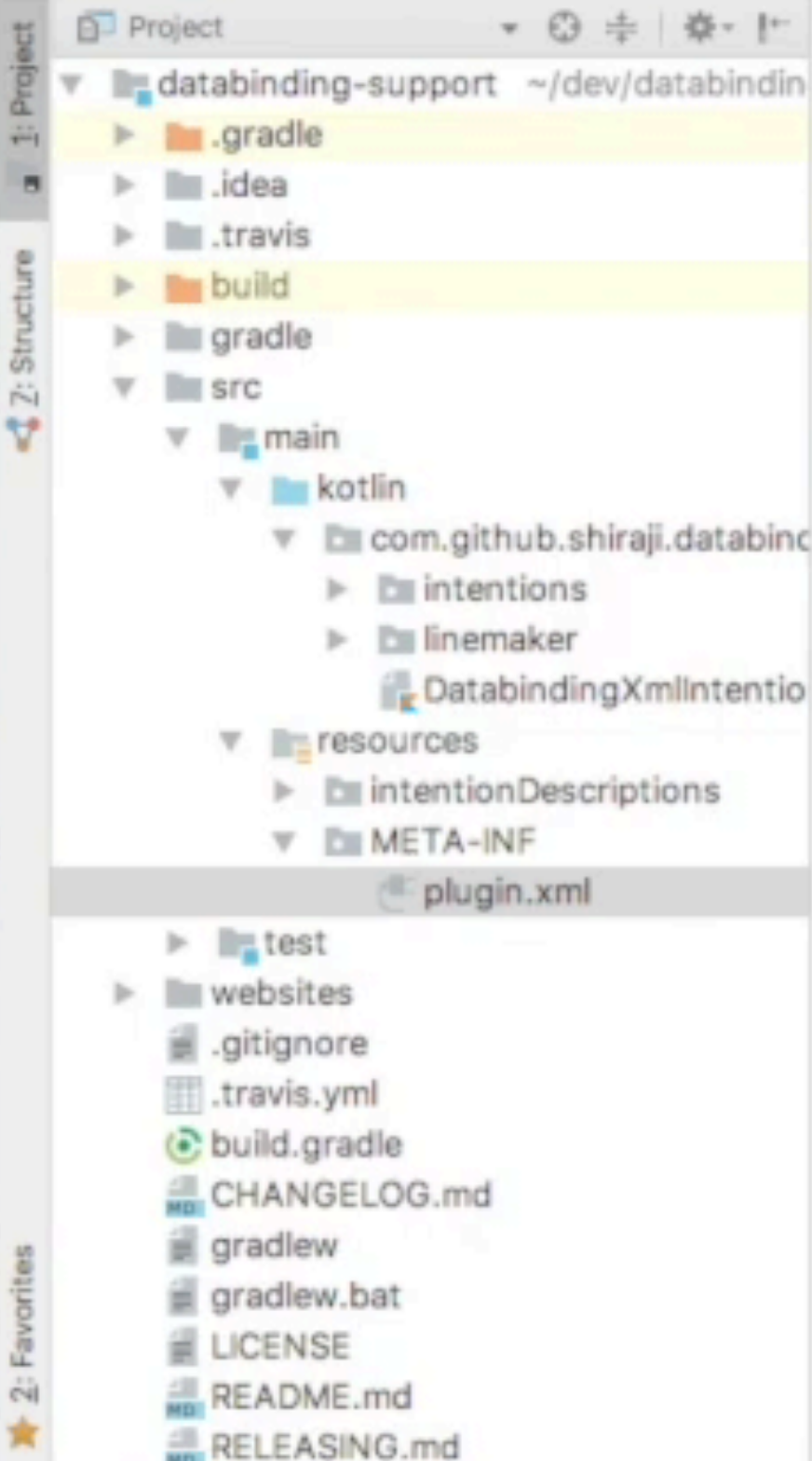
    override fun isAvailable(project: Project, editor: Editor?, file: PsiFile?) {
        val rootTag = file.getRootTag() ?: return false
        if (rootTag.isDatabindingRootTag()) return false
        return !BLACK_LIST.contains(rootTag.name)
    }

    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

9: Version Control Breakpoints Manager Terminal 4: Run 6: TODO 1: Event Log



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

import ...

class ConvertToDatabindingLayoutIntention : IntentionAction {

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

    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

Official Support!



Android Studio 3.0