ICSE’24, IDE Workshop

Developing IDE Plugins

Zarina Kurbatova
Researcher at JetBrains

Why develop IDE plugins?

- **Increase research impact**
  - Plugins can be easily integrated into developers workflows
- **Extend behaviour of your favorite IDE in different ways**
  - Integration with external services like OpenAI
  - Support of third party tools such as Spring
  - Additional UI elements such as menus or tool windows
- **Educate next generation of software developers**
  - JetBrains Academy plugin provides possibilities for creating educational courses inside IDE

What is the IntelliJ Platform?

A platform for building IDEs and language-aware developer tools

- Code analysis
  - PSI (Program Structure Interface) responsible for parsing code and building syntactic trees
- UI Toolkit (Tool Windows, popup menus, dialogs, …)
- Code transformations
  - Refactorings
  - Quick fixes
- Syntax highlighting, code folding, code completion, …
- Integration with Git

Program Structure Interface (PSI)

A layer responsible for parsing files and creating syntactic and semantic models

All tree elements implement a base interface **PsiElement**

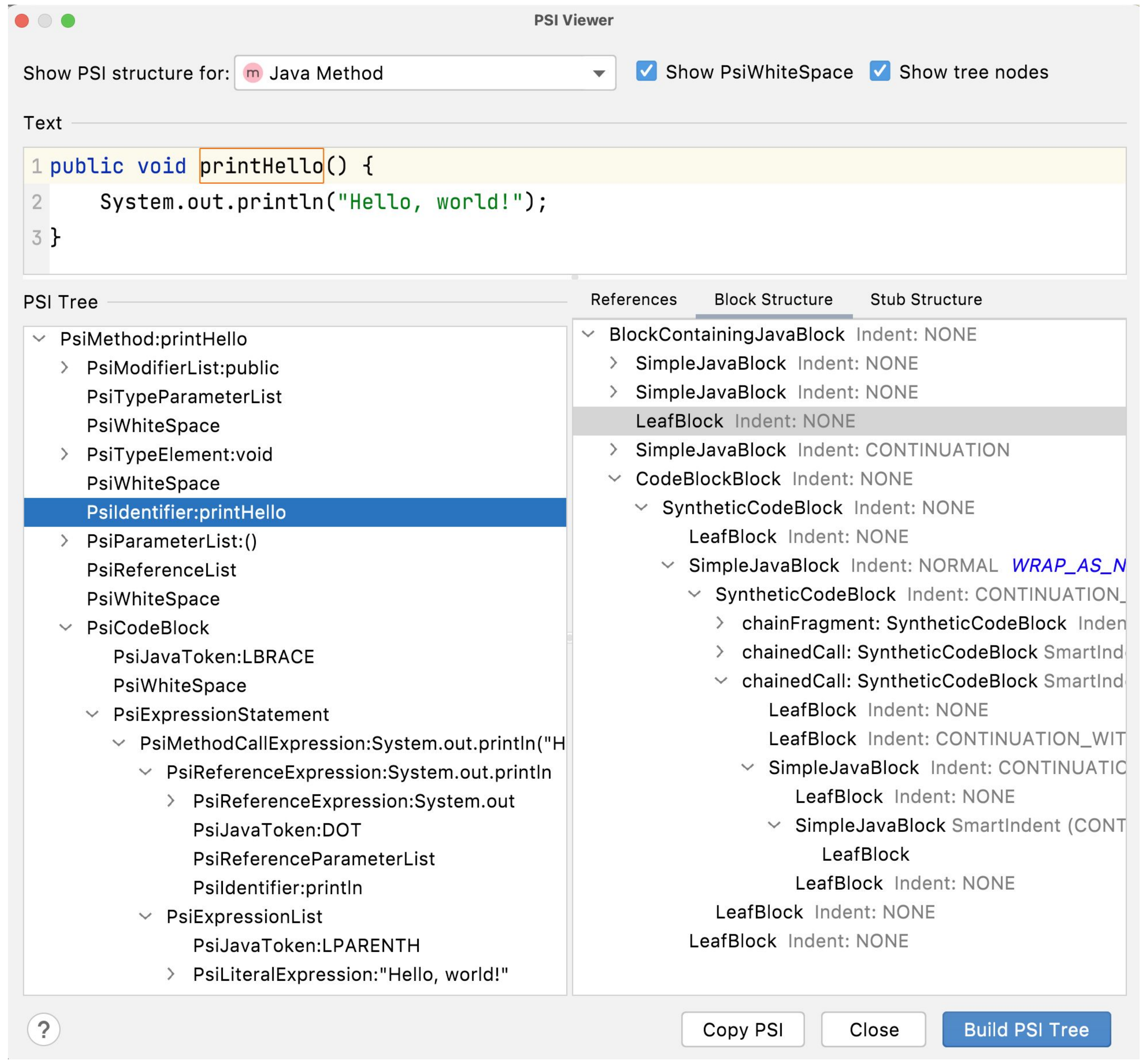
Each language has its own implementation of PSI elements

PSI allows to:

- Extract entities of the specific type
- Resolve types
- Search for usages of some entity

PSIViewer - plugin that visualizes PSI structure of code

Documentation



Show PSI structure for: Java Method   ✓ Show PsiWhiteSpace   ✓ Show tree nodes

Text

```
public void printHello() {
    System.out.println("Hello, world!");
}
```

PSI Tree

```
PsiMethod:printHello
  PsiModifierList:public
  PsiTypeParameterList
  PsiWhiteSpace
  PsiTypeElement:void
  PsiWhiteSpace
  PsiIdentifier:printHello
  PsiParameterList:()
  PsiReferenceList
  PsiWhiteSpace
  PsiCodeBlock
    PsiJavaToken:LBRACE
    PsiWhiteSpace
    PsiExpressionStatement
      PsiMethodCallExpression:System.out.println("H
        PsiReferenceExpression:System.out.println
          PsiReferenceExpression:System.out
          PsiJavaToken:DOT
          PsiReferenceParameterList
          PsiIdentifier:println
        PsiExpressionList
          PsiJavaToken:LPARENTH
          PsiLiteralExpression:"Hello, world!"
```

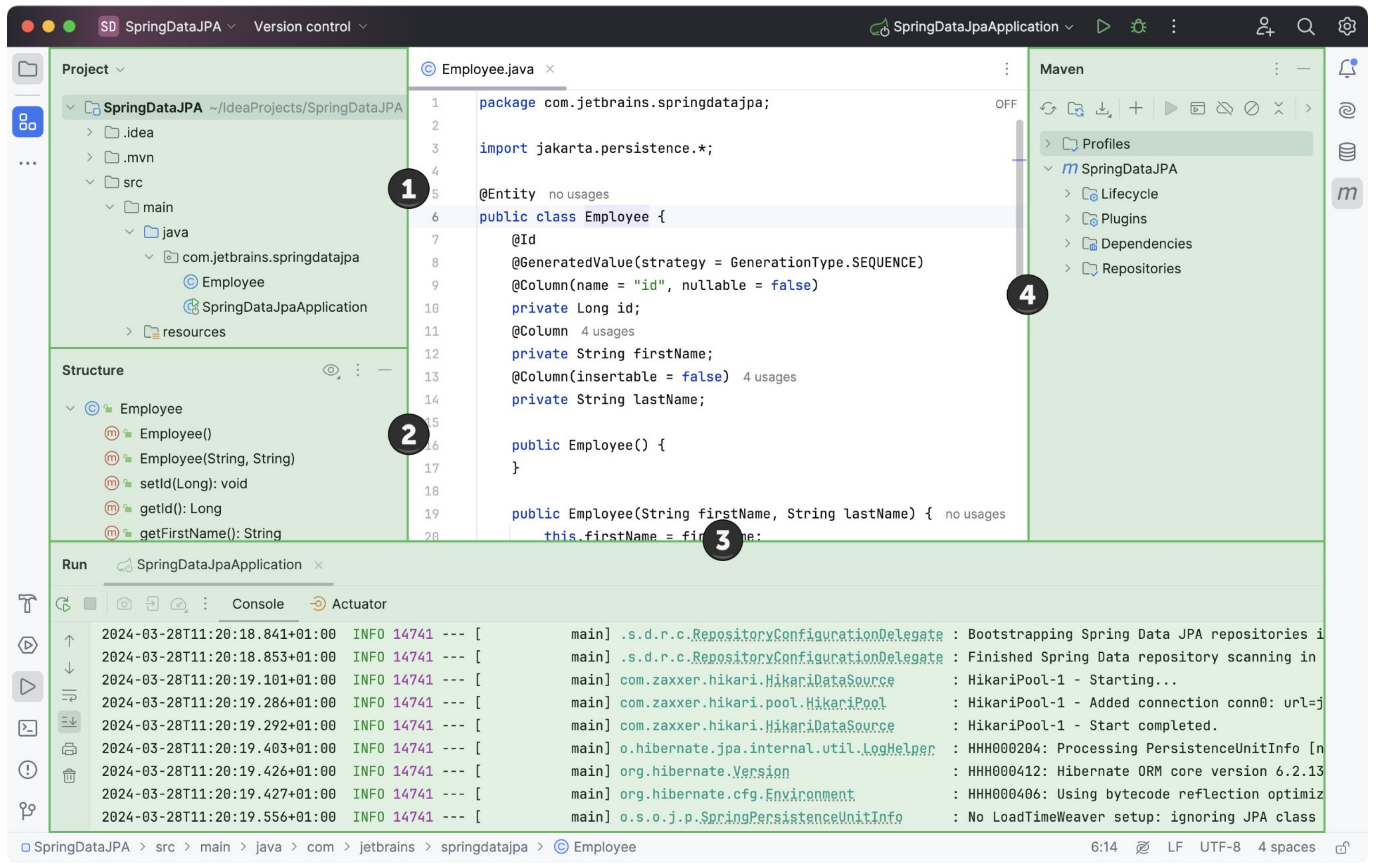
References | Block Structure | Stub Structure

```
BlockContainingJavaBlock  Indent: NONE
  SimpleJavaBlock  Indent: NONE
  SimpleJavaBlock  Indent: NONE
  LeafBlock  Indent: NONE
  SimpleJavaBlock  Indent: CONTINUATION
  CodeBlockBlock  Indent: NONE
    SyntheticCodeBlock  Indent: NONE
      LeafBlock  Indent: NONE
      SimpleJavaBlock  Indent: NORMAL  WRAP_AS_N
        SyntheticCodeBlock  Indent: CONTINUATION_
          chainFragment: SyntheticCodeBlock  Inden
          chainedCall: SyntheticCodeBlock SmartInd
          chainedCall: SyntheticCodeBlock SmartInd
            LeafBlock  Indent: NONE
            LeafBlock  Indent: CONTINUATION_WIT
            SimpleJavaBlock  Indent: CONTINUATIC
              LeafBlock  Indent: NONE
              SimpleJavaBlock SmartIndent (CONT
                LeafBlock
              LeafBlock  Indent: NONE
        LeafBlock  Indent: NONE
      LeafBlock  Indent: NONE
```

Extensions

The most common way to extend IDE functionality

- 1500+ extension points
- Should be registered in `plugin.xml` in `<extensions>` section
- For example, the `com.intellij.toolWindow` extension point allows to add tool windows (panels displayed at the sides of the IDE user interface)

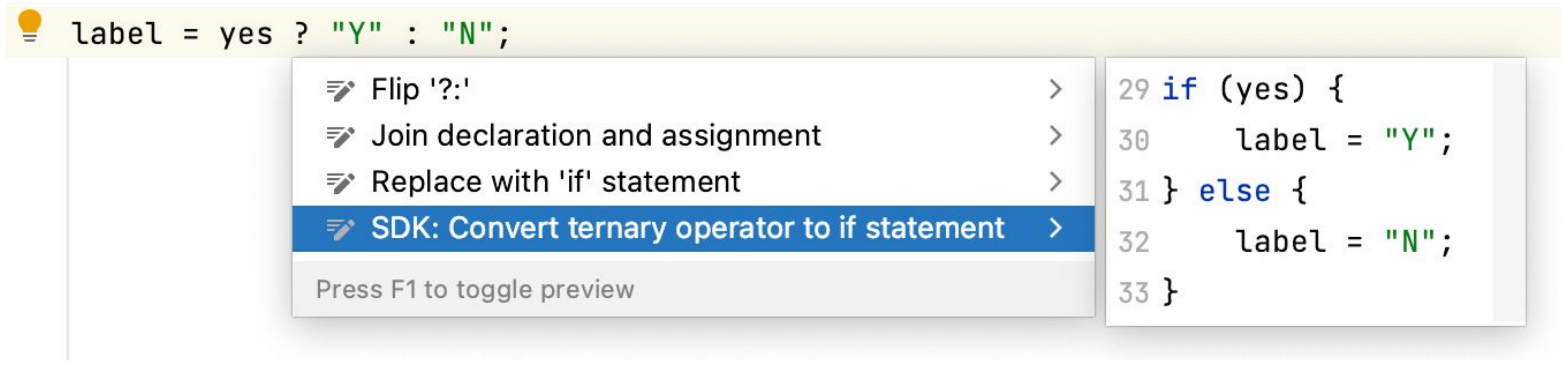


```java
package com.jetbrains.springdatajpa;

import jakarta.persistence.*;

@Entity  no usages
public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    @Column(name = "id", nullable = false)
    private Long id;
    @Column  4 usages
    private String firstName;
    @Column(insertable = false)  4 usages
    private String lastName;

    public Employee() {
    }

    public Employee(String firstName, String lastName) {  no usages
        this.firstName = fi[illegible]e;
```

```
2024-03-28T11:20:18.841+01:00  INFO 14741 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories i
2024-03-28T11:20:18.853+01:00  INFO 14741 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in
2024-03-28T11:20:19.101+01:00  INFO 14741 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-03-28T11:20:19.286+01:00  INFO 14741 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection conn0: url=j
2024-03-28T11:20:19.292+01:00  INFO 14741 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-03-28T11:20:19.403+01:00  INFO 14741 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [n
2024-03-28T11:20:19.426+01:00  INFO 14741 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.2.13
2024-03-28T11:20:19.427+01:00  INFO 14741 --- [           main] org.hibernate.cfg.Environment            : HHH000406: Using bytecode reflection optimiz
2024-03-28T11:20:19.556+01:00  INFO 14741 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup: ignoring JPA class
```

Intention actions

- Analyzes the currently opened file and suggests code improvements
- Should be implemented using `IntentionAction` interface and registered in the `plugin.xml` in `<extensions>` section using the `com.intellij.intentionAction` extension point



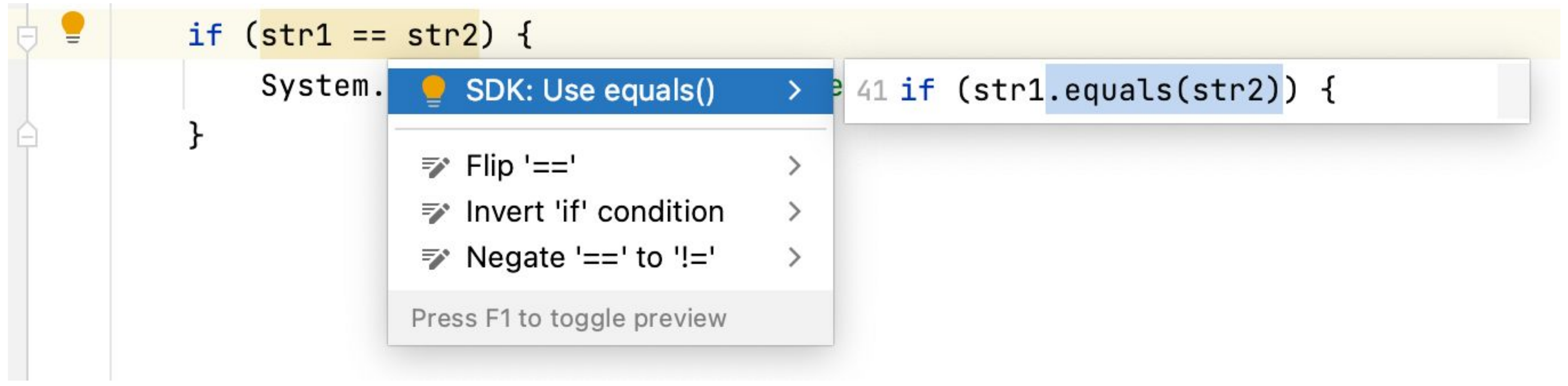
```
label = yes ? "Y" : "N";
```

Flip '?:'
Join declaration and assignment
Replace with 'if' statement
SDK: Convert ternary operator to if statement
Press F1 to toggle preview

```
if (yes) {
    label = "Y";
} else {
    label = "N";
}
```

Inspections

- Run static analysis on the code in the background mode
- Highlight the code that has some problems and suggest automatic quick-fixes
- Scope is configurable (file, package, project)
- Should be implemented using `LocalInspectionTool` and registered in `plugin.xml` in `<extensions>` section using `com.intellij.localInspection` extension point



```
if (str1 == str2) {
    System.
}
```

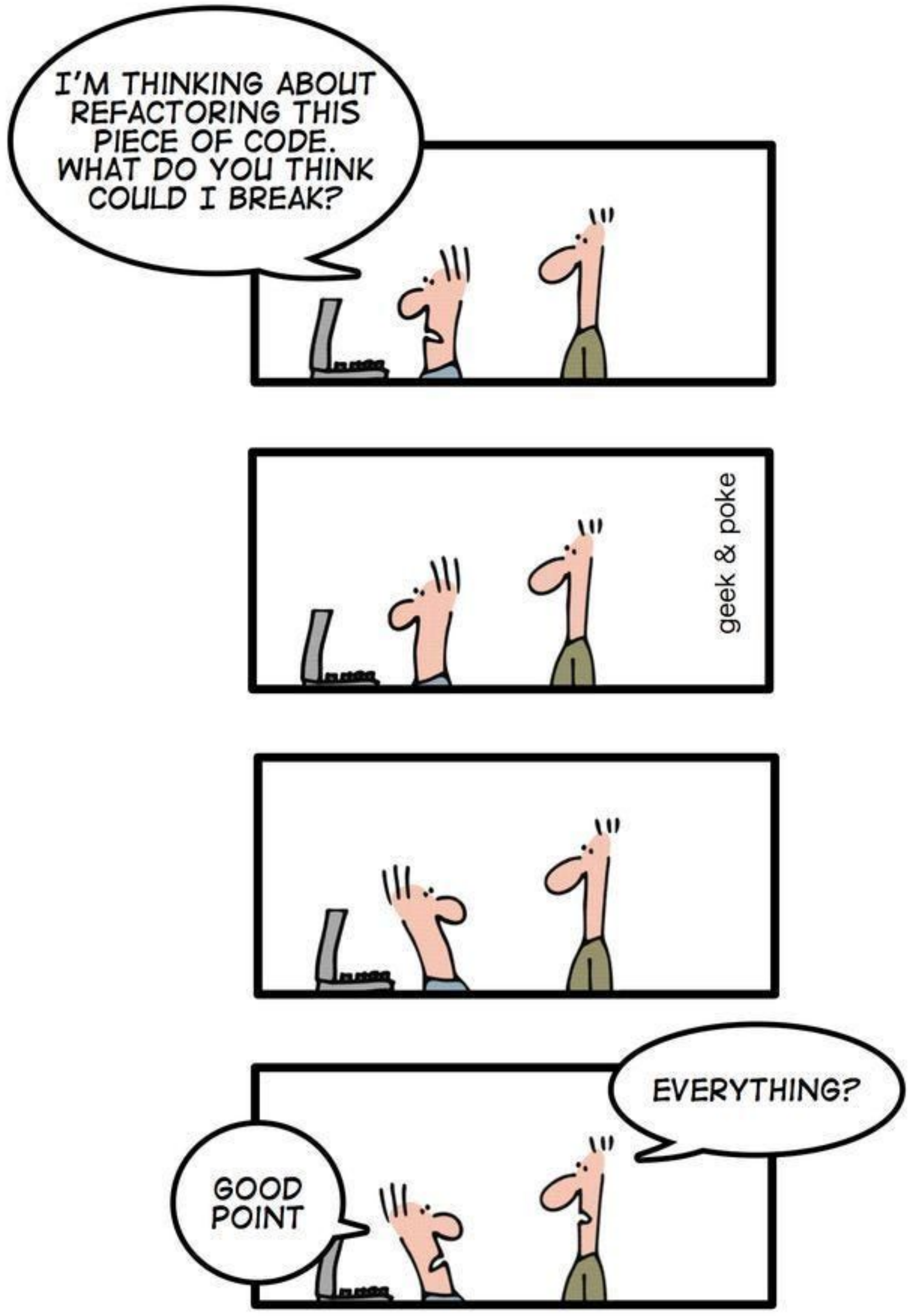
```
if (str1.equals(str2)) {
```

Refactorings

The are already implemented refactorings

See package `com.intellij.refactoring`

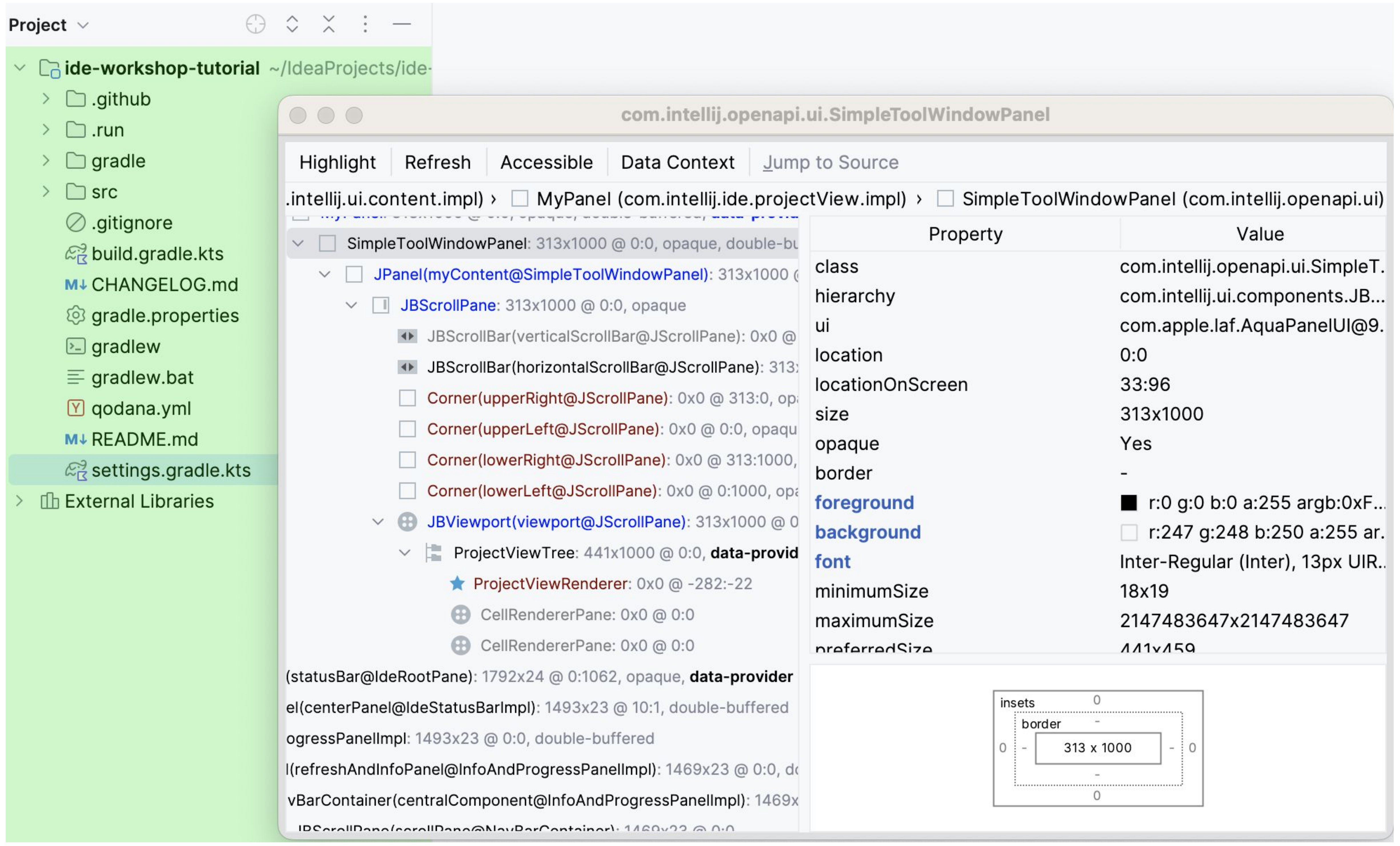
- Move Method/Class
- Extract Method/Class/Variable
- Rename Method/Variable/Class
- Inline Method/Variable
- Pull Up Method
- Push Down Method


I'M THINKING ABOUT REFACTORING THIS PIECE OF CODE. WHAT DO YOU THINK COULD I BREAK?
geek & poke
EVERYTHING?
GOOD POINT

UI Toolkit

Add custom UI elements to IDE

- The IntelliJ Platform has a lot of custom Swing components that you can reuse in your plugin
- Menus, Tool Windows, Popups, Notifications, and so on
- UI Inspector allows to inspect existing UI component of IDE
- Platform UI guidelines provides tips on how to create consistent user interfaces



```
.intellij.ui.content.impl) › MyPanel (com.intellij.ide.projectView.impl) › SimpleToolWindowPanel (com.intellij.openapi.ui)
SimpleToolWindowPanel: 313x1000 @ 0:0, opaque, double-bu
  JPanel(myContent@SimpleToolWindowPanel): 313x1000 @
    JBScrollPane: 313x1000 @ 0:0, opaque
      JBScrollBar(verticalScrollBar@JScrollPane): 0x0 @
      JBScrollBar(horizontalScrollBar@JScrollPane): 313:
      Corner(upperRight@JScrollPane): 0x0 @ 313:0, op
      Corner(upperLeft@JScrollPane): 0x0 @ 0:0, opaqu
      Corner(lowerRight@JScrollPane): 0x0 @ 313:1000,
      Corner(lowerLeft@JScrollPane): 0x0 @ 0:1000, op
      JBViewport(viewport@JScrollPane): 313x1000 @ 0
        ProjectViewTree: 441x1000 @ 0:0, data-provid
          ProjectViewRenderer: 0x0 @ -282:-22
          CellRendererPane: 0x0 @ 0:0
          CellRendererPane: 0x0 @ 0:0
(statusBar@IdeRootPane): 1792x24 @ 0:1062, opaque, data-provider
el(centerPanel@IdeStatusBarImpl): 1493x23 @ 10:1, double-buffered
ogressPanelImpl: 1493x23 @ 0:0, double-buffered
l(refreshAndInfoPanel@InfoAndProgressPanelImpl): 1469x23 @ 0:0, d
vBarContainer(centralComponent@InfoAndProgressPanelImpl): 1469x
```

Property	Value
class	com.intellij.openapi.ui.SimpleT...
hierarchy	com.intellij.ui.components.JB...
ui	com.apple.laf.AquaPanelUI@9.
location	0:0
locationOnScreen	33:96
size	313x1000
opaque	Yes
border	-
foreground	■ r:0 g:0 b:0 a:255 argb:0xF...
background	□ r:247 g:248 b:250 a:255 ar.
font	Inter-Regular (Inter), 13px UIR..
minimumSize	18x19
maximumSize	2147483647x2147483647
preferredSize	441x459

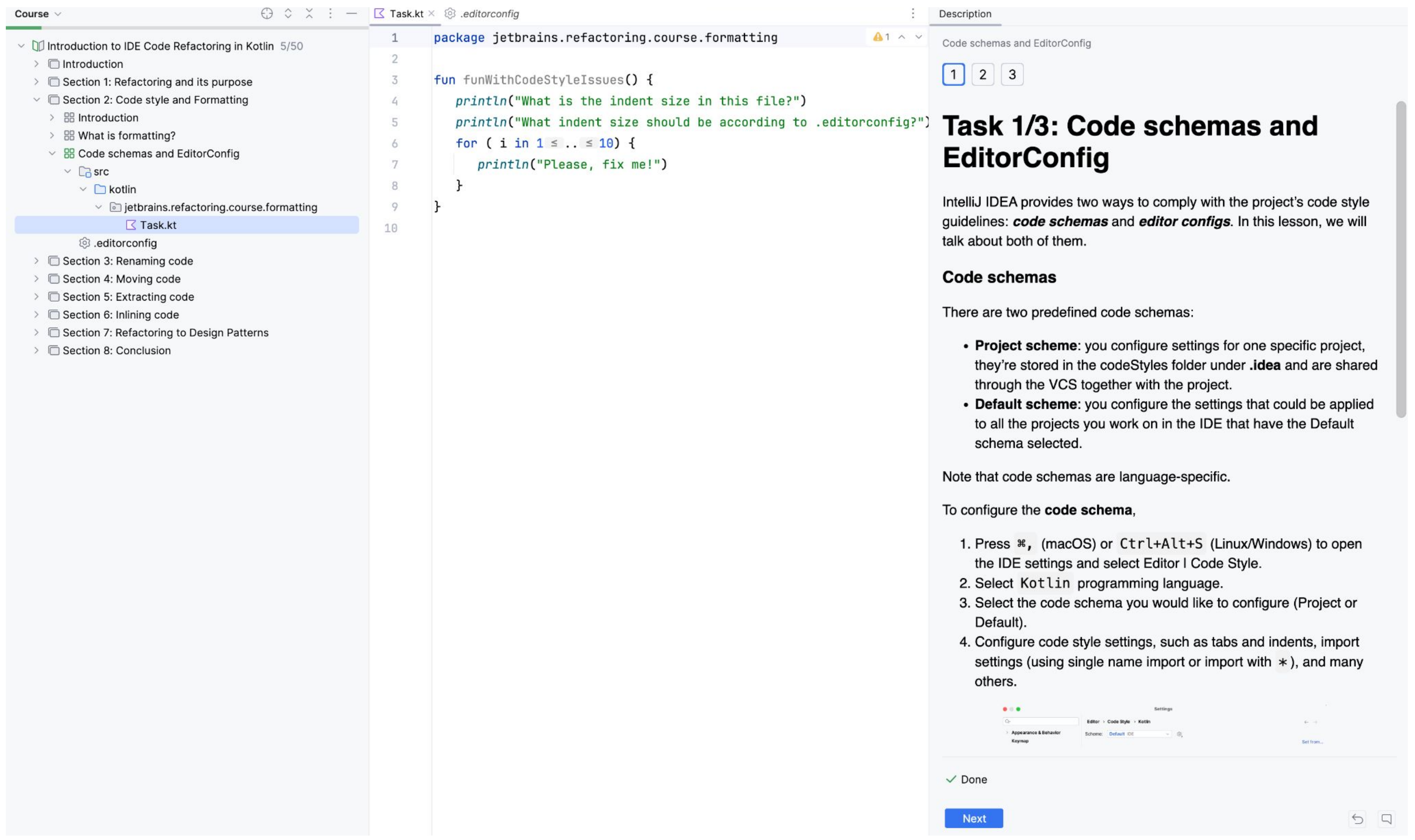
Run ML models in plugins

- KInference library allows to execute ML models written in ONNX format
- KotlinDL - is a high-level Deep Learning API written in Kotlin and inspired by Keras
  - uses TensorFlow Java API and ONNX Runtime API for Java
- ONNX Runtime Java API allows to inference ML models written in ONNX format
- Tribuo is a machine learning library written in Java
  - provides tools for classification, regression, clustering, model development, and more

Educational plugins

Wrap your educational course into an IDE plugin

- Several types of tasks: **theory**, **coding**, **quizzes**
- Testing system
- Share a course on Marketplace
- Course creator start guide
- Introduction to IDE Code Refactoring in Kotlin



Course
Introduction to IDE Code Refactoring in Kotlin 5/50
Introduction
Section 1: Refactoring and its purpose
Section 2: Code style and Formatting
Introduction
What is formatting?
Code schemas and EditorConfig
src
kotlin
jetbrains.refactoring.course.formatting
Task.kt
.editorconfig
Section 3: Renaming code
Section 4: Moving code
Section 5: Extracting code
Section 6: Inlining code
Section 7: Refactoring to Design Patterns
Section 8: Conclusion
Task.kt
.editorconfig
Description

```
package jetbrains.refactoring.course.formatting

fun funWithCodeStyleIssues() {
   println("What is the indent size in this file?")
   println("What indent size should be according to .editorconfig?")
   for ( i in 1 ≤ .. ≤ 10) {
      println("Please, fix me!")
   }
}
```

Code schemas and EditorConfig

1 2 3

Task 1/3: Code schemas and EditorConfig

IntelliJ IDEA provides two ways to comply with the project's code style guidelines: ***code schemas*** and ***editor configs***. In this lesson, we will talk about both of them.

Code schemas

There are two predefined code schemas:

- **Project scheme**: you configure settings for one specific project, they're stored in the codeStyles folder under **.idea** and are shared through the VCS together with the project.
- **Default scheme**: you configure the settings that could be applied to all the projects you work on in the IDE that have the Default schema selected.

Note that code schemas are language-specific.

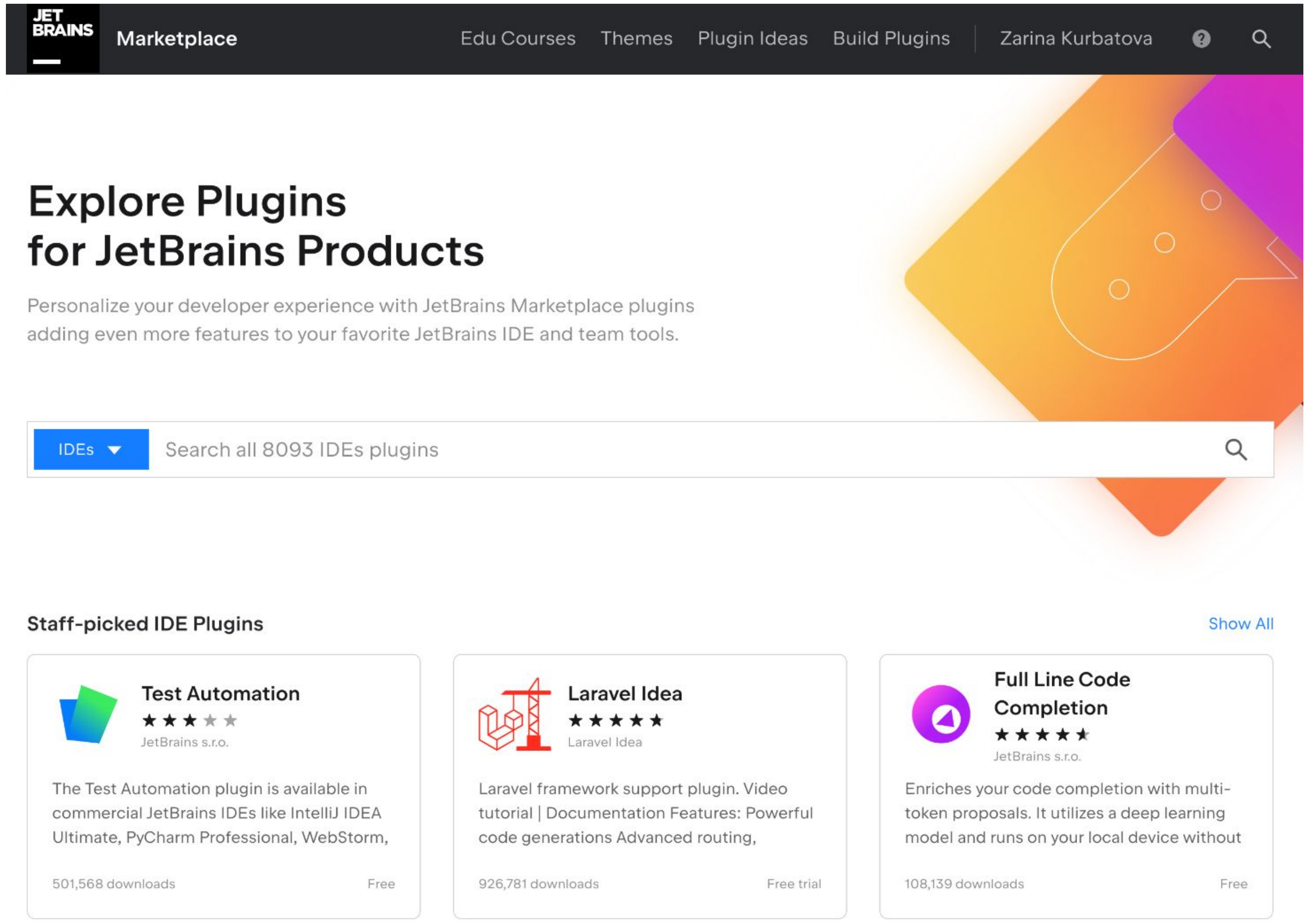
To configure the **code schema**,

1. Press `⌘,` (macOS) or `Ctrl+Alt+S` (Linux/Windows) to open the IDE settings and select Editor | Code Style.
2. Select `Kotlin` programming language.
3. Select the code schema you would like to configure (Project or Default).
4. Configure code style settings, such as tabs and indents, import settings (using single name import or import with `*`), and many others.

Distributing your plugins

Marketplace – official JetBrains plugin repository

- Free and paid plugins
- **6.3M** monthly plugin downloads
- **86%** of JetBrains IDEs users have at least one plugin installed
- **8000+** plugins
- Automatic compatibility verification



Explore Plugins for JetBrains Products

Personalize your developer experience with JetBrains Marketplace plugins adding even more features to your favorite JetBrains IDE and team tools.

Staff-picked IDE Plugins

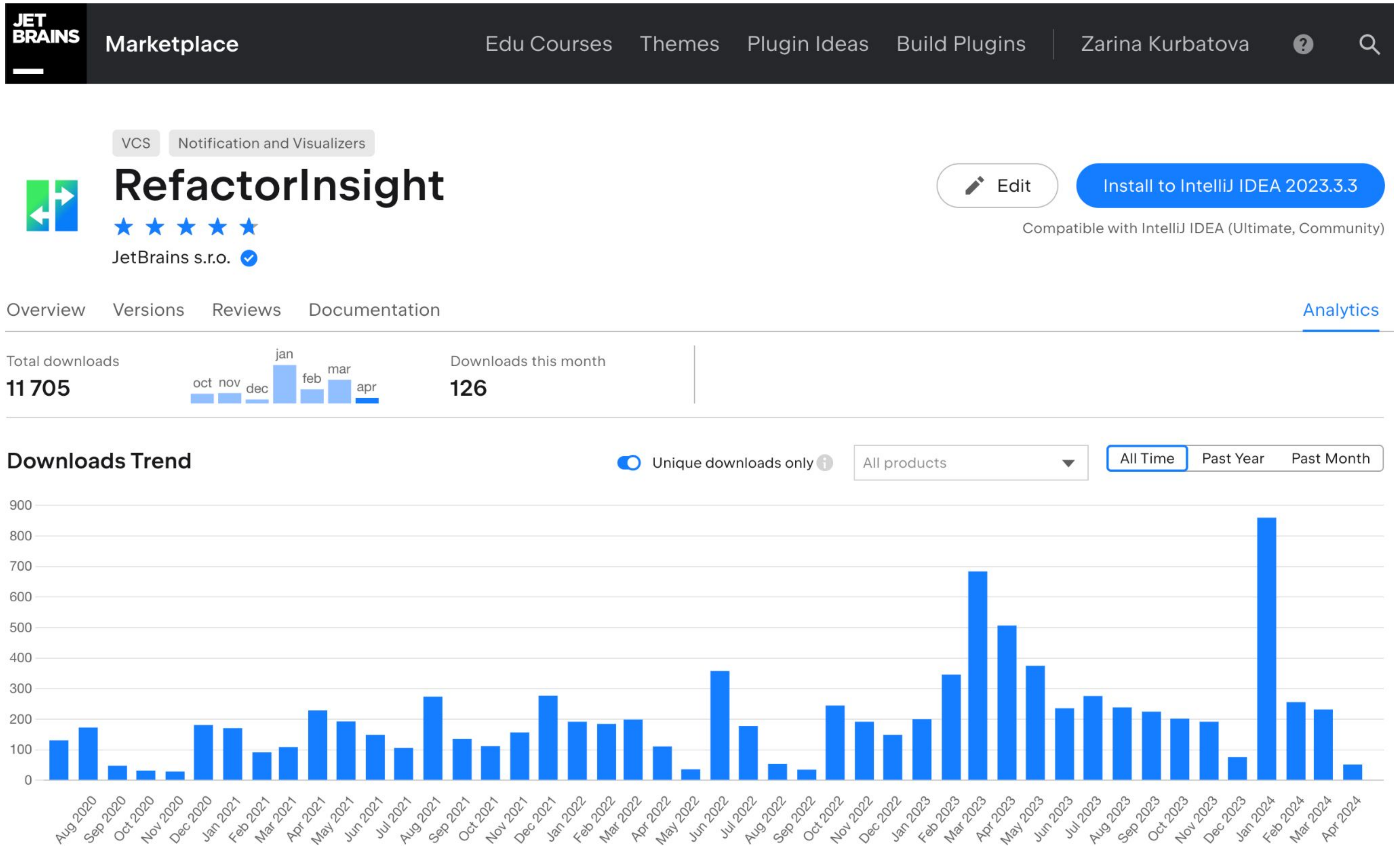
Test Automation
JetBrains s.r.o.
The Test Automation plugin is available in commercial JetBrains IDEs like IntelliJ IDEA Ultimate, PyCharm Professional, WebStorm,
501,568 downloads — Free

Laravel Idea
Laravel Idea
Laravel framework support plugin. Video tutorial | Documentation Features: Powerful code generations Advanced routing,
926,781 downloads — Free trial

Full Line Code Completion
JetBrains s.r.o.
Enriches your code completion with multi-token proposals. It utilizes a deep learning model and runs on your local device without
108,139 downloads — Free

Downloads statistics

- Downloads trends
- Downloads by product (IntelliJ IDEA Ultimate/Community Edition)
- Plugin page visitors
- Page visits by country/region
- Page referrals



Marketplace — Edu Courses, Themes, Plugin Ideas, Build Plugins, Zarina Kurbatova

VCS · Notification and Visualizers

RefactorInsight

JetBrains s.r.o.

Edit · Install to IntelliJ IDEA 2023.3.3

Compatible with IntelliJ IDEA (Ultimate, Community)

Overview · Versions · Reviews · Documentation · Analytics

Total downloads: 11 705

Downloads this month: 126

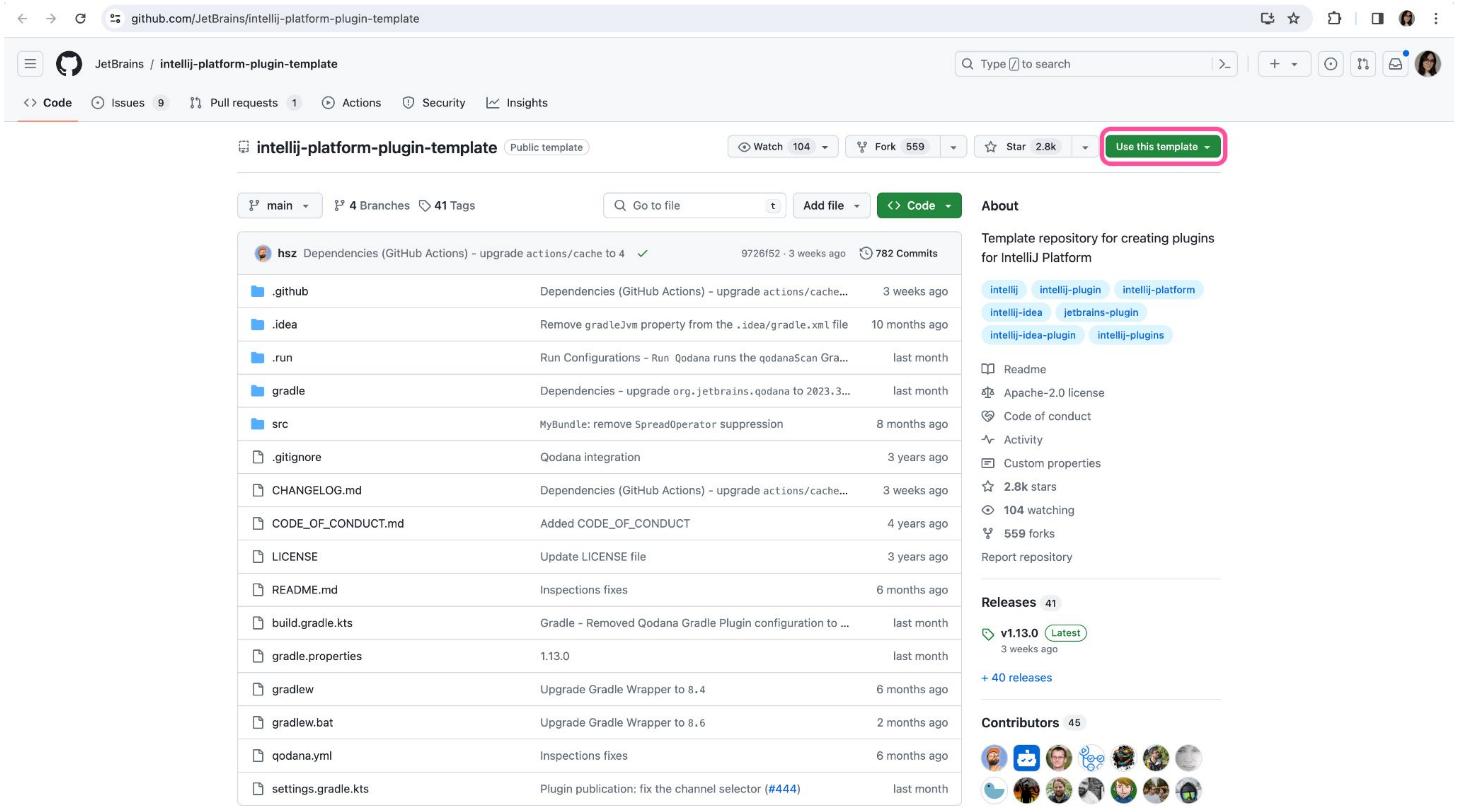
Downloads Trend

Unique downloads only: on · All products · All Time · Past Year · Past Month

Where to start

Use IntelliJ Platform Plugin Template

- Already configured Gradle project
- CI setup
- Sample code



github.com/JetBrains/intellij-platform-plugin-template

JetBrains / intellij-platform-plugin-template

intellij-platform-plugin-template Public template

Use this template

main · 4 Branches · 41 Tags

About

Template repository for creating plugins for IntelliJ Platform

intellij · intellij-plugin · intellij-platform · intellij-idea · jetbrains-plugin · intellij-idea-plugin · intellij-plugins

Readme · Apache-2.0 license · Code of conduct · Activity · Custom properties · 2.8k stars · 104 watching · 559 forks

Releases 41 · v1.13.0 Latest · 3 weeks ago · + 40 releases

Contributors 45

hsz Dependencies (GitHub Actions) - upgrade actions/cache to 4 ✓ 9726f52 · 3 weeks ago 782 Commits

.github	Dependencies (GitHub Actions) - upgrade actions/cache...	3 weeks ago
.idea	Remove gradleJvm property from the .idea/gradle.xml file	10 months ago
.run	Run Configurations - Run Qodana runs the qodanaScan Gra...	last month
gradle	Dependencies - upgrade org.jetbrains.qodana to 2023.3...	last month
src	MyBundle: remove SpreadOperator suppression	8 months ago
.gitignore	Qodana integration	3 years ago
CHANGELOG.md	Dependencies (GitHub Actions) - upgrade actions/cache...	3 weeks ago
CODE_OF_CONDUCT.md	Added CODE_OF_CONDUCT	4 years ago
LICENSE	Update LICENSE file	3 years ago
README.md	Inspections fixes	6 months ago
build.gradle.kts	Gradle - Removed Qodana Gradle Plugin configuration to ...	last month
gradle.properties	1.13.0	last month
gradlew	Upgrade Gradle Wrapper to 8.4	6 months ago
gradlew.bat	Upgrade Gradle Wrapper to 8.6	2 months ago
qodana.yml	Inspections fixes	6 months ago
settings.gradle.kts	Plugin publication: fix the channel selector (#444)	last month

Demo

Useful links

- IntelliJ Platform Plugin SDK
  - Official documentation on plugin development for JetBrains IDEs
- JetBrains Platform Slack
  - Community of plugin developers
- Busy Plugin Developers blog
  - News about the IntelliJ Platform, Marketplace, webinars
- People Behind Plugins series
  - Interviews with IDE plugin developers
- "The IntelliJ Platform: a Framework for Building Plugins and Mining Software Data" paper
- **Demo plugin from the IDE workshop: https://github.com/JetBrains-Research/ide-workshop-tutorial**

Demo plugin