Unlocking the Power of IntelliJ

Creating Plugins for Ultimate Productivity

Thessaloniki – Java Meetup
19 October 2023
Eleftherios Chrysochoidis

Agenda

- Introduction
- SDK Overview
- Plugin Development
- Hands-on / CodeTour
- Resources & QA
- Quiz

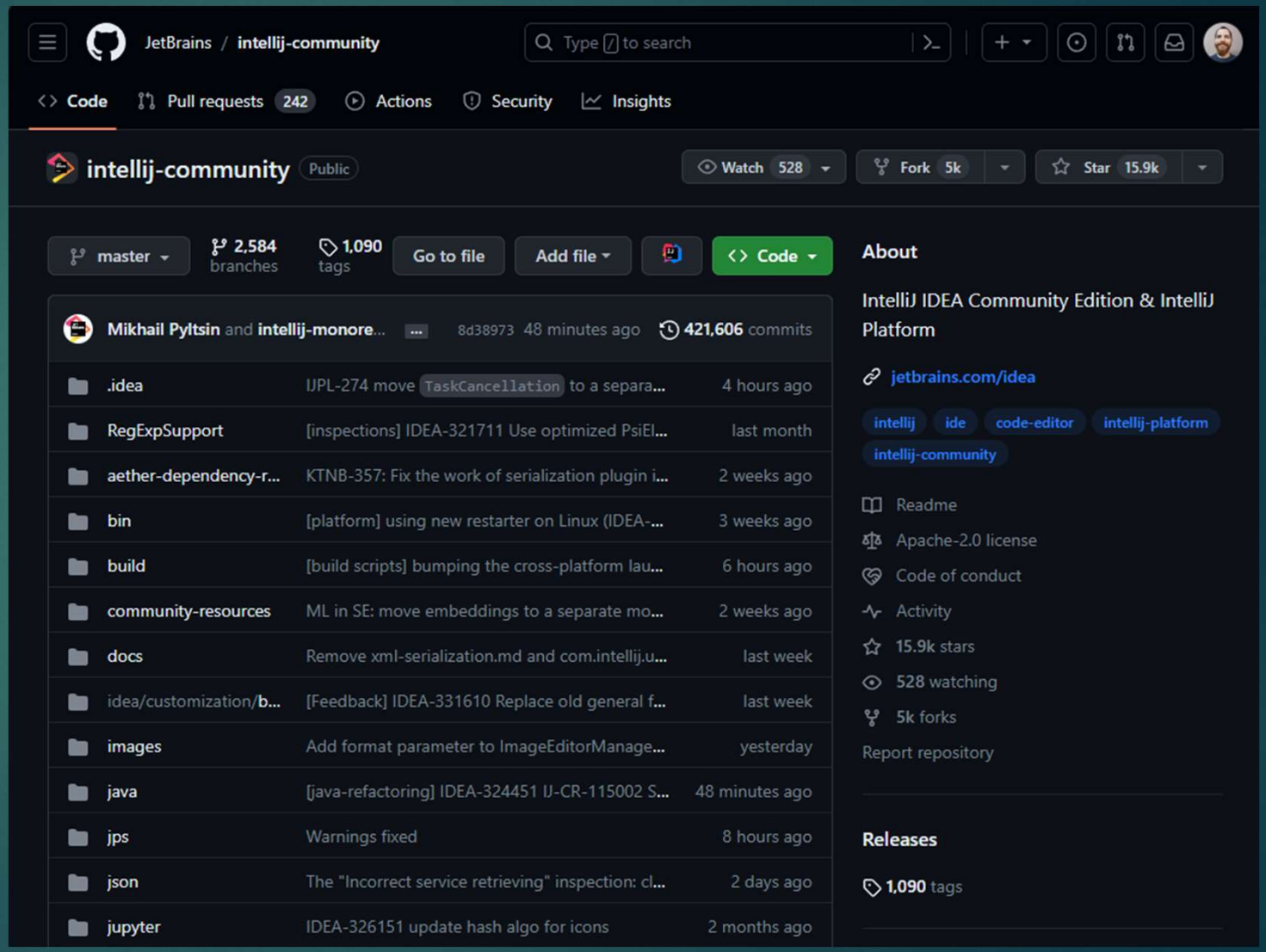
Introduction

- **Platform Overview**
- Required Experience

- IntelliJ Platform is not a product itself!
- Component-driven, JVM application
- Language agnostic – abstract implementations!
  - Full-text Editor
  - Syntax highlighting
  - Built in Fast Search using Indexes
  - Code Completion
  - Breakpoints and Debugging features
- Open Source!

Introduction

- **Platform Overview**
- Required Experience



Introduction

- Platform Overview
- **Required Experience**

- **Java**, **Kotlin** and its standard and 3rd-party libraries
- **Gradle** or **Maven** as build system (Gradle is preferred)
- **Swing** for building user interfaces
- Java Concurrency Model (Threads, Executors, Locks)
- Experience with IntelliJ Platform-based IDE (e.g., IntelliJ IDEA)

SDK Overview

- **Actions**
- Virtual File System
- Project Model
- PSI
- Editors

- Action = Code Implementation + Registration
- Code Implementation determines the context and its functionality
- Registration determines where an action appears in UI
- Callbacks from IntelliJ Platform in response to user gestures

SDK Overview

- Actions
- **Virtual File System**
- Project Model
- PSI
- Editors

- Encapsulation of all the activity of a Virtual File
- A Virtual file, might be:
  - Normal File (on disk)
  - Decompiled File (in a lib/archive)
  - Remote File (on an HTTP server)
- Why?
  - Universal API for working with files
  - Track File modifications - File versioning
  - Associate with persistent metadata - snapshot

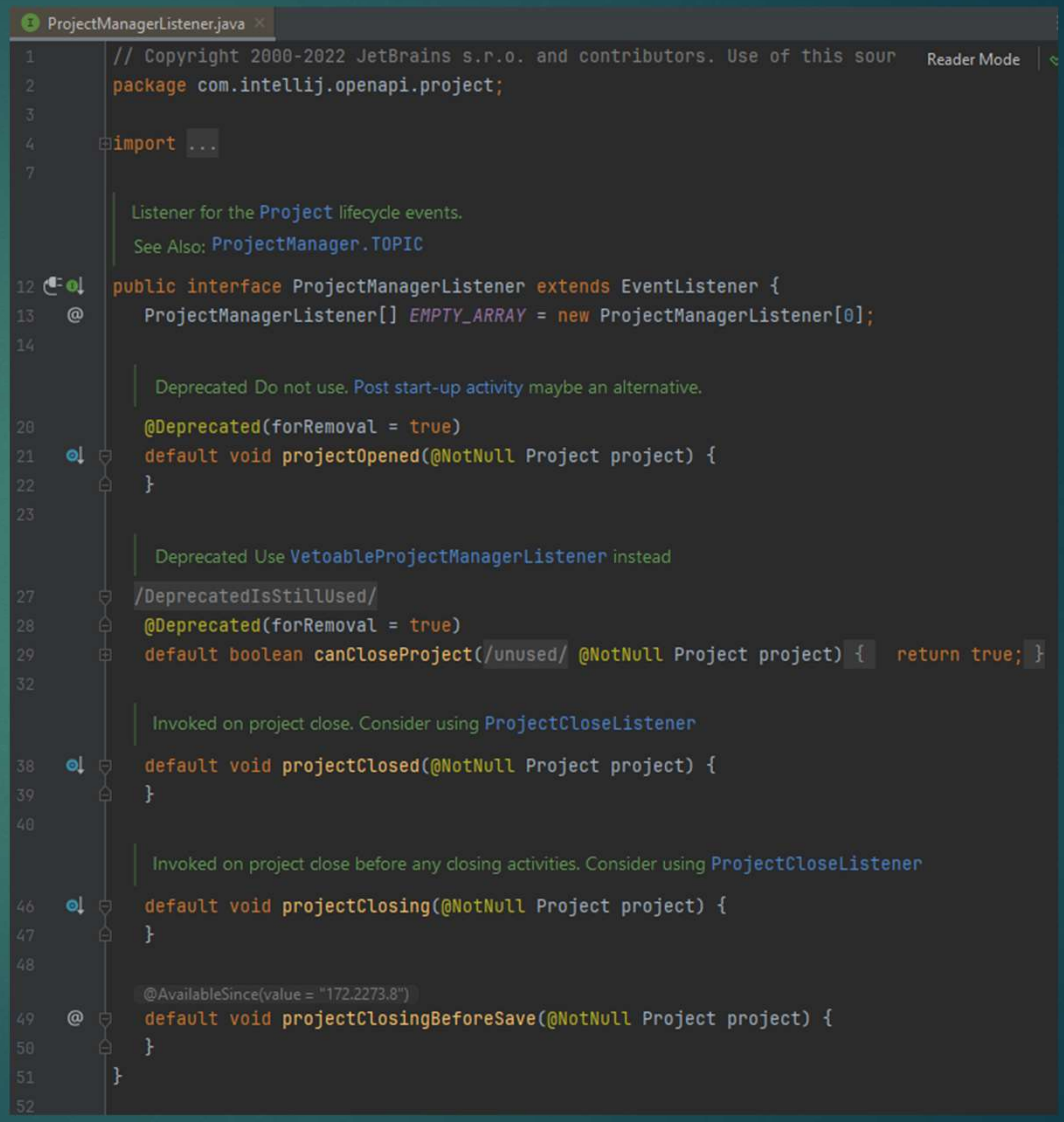
SDK Overview

- Actions
- Virtual File System
- **Project Model**
- PSI
- Editors

- Project | Source code, build instructions, libraries
- Module | unit of functionality
- Library | 3rd party compiled code
- SDK | For building the project
- Facet | Set of configuration

SDK Overview

- Actions
- Virtual File System
- **Project Model**
- PSI
- Editors



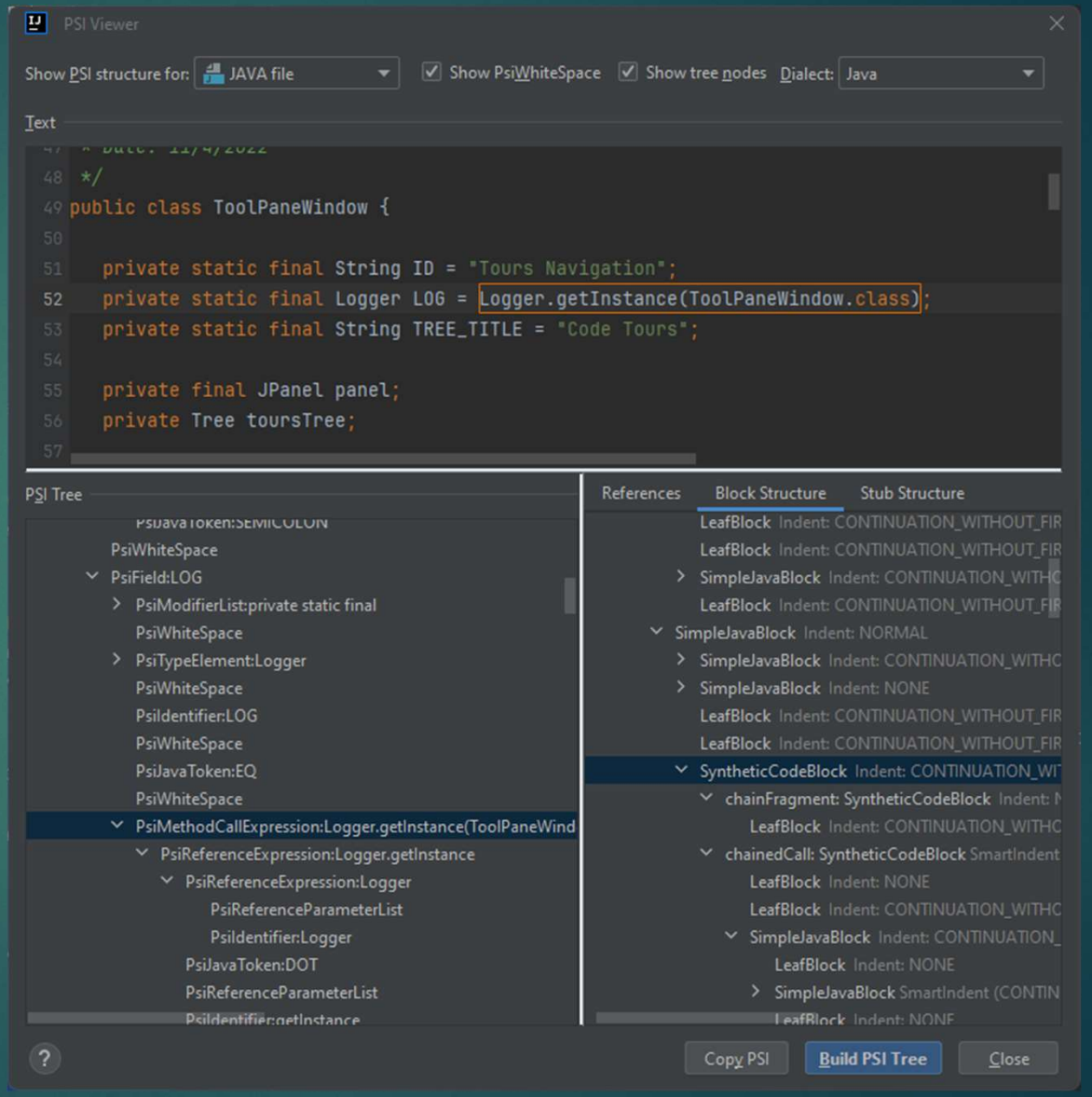
SDK Overview

- Actions
- Virtual File System
- Project Model
- **PSI**
- Editors

- A PSI (Program Structure Interface) file is the tree-like representation of a file's contents
- Language Specific Implementation – Generic Parsers are available
- Frequently Updated but Ultra Fast!
- Backs the Code formatting, Code completion and highlighting
- PSI Viewer

SDK Overview

- Actions
- Virtual File System
- Project Model
- **PSI**
- Editors

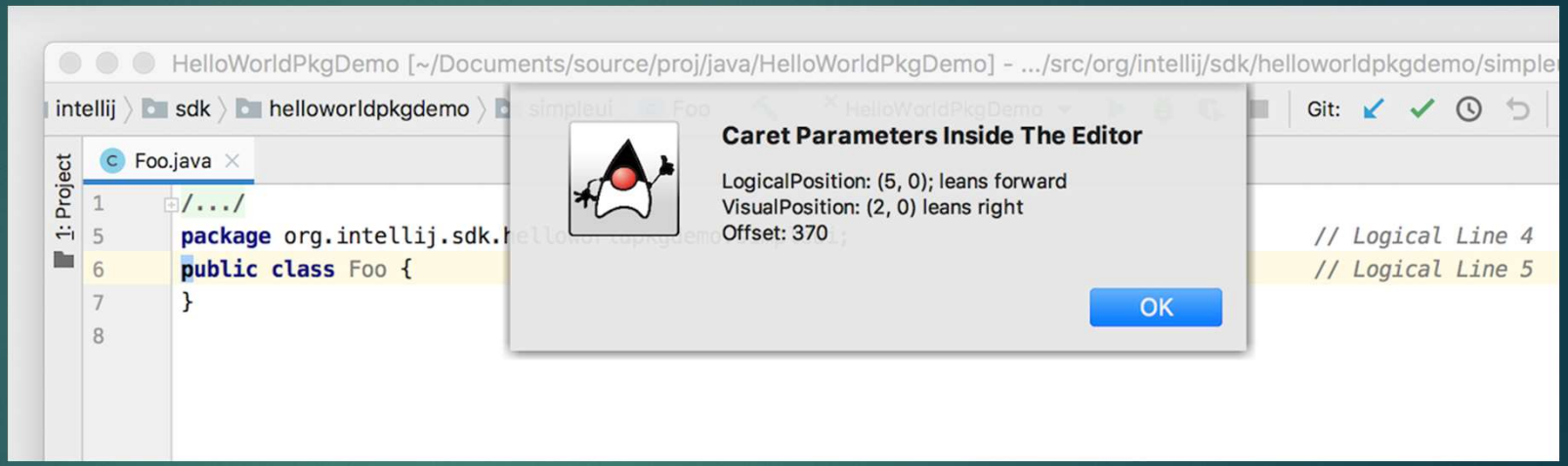


SDK Overview

- Actions
- Virtual File System
- Project Model
- PSI
- **Editors**
  - Editor Active View
  - Caret (or multiple carets) for Actions (e.g. right click)
  - Code Block Selection
  - Logical vs Visual Position

SDK Overview

- Actions
- Virtual File System
- Project Model
- PSI
- **Editors**



Plugin Development

- **Alternative to Plugin**
- Plugin Types
- Plugin Structure

- SSR - Structural Search and Replace Inspections
  - Search patterns
  - Code Inspections!
- IDE Scripting Console
  - Console with access to IntelliJ platform API
- Flora Plugin
  - Scripts (Kotlin or JavaScript) for using IntelliJ platform API
  - Shareable with others!

Plugin Development

- Alternative to Plugin
- **Plugin Types**
- Plugin Structure

- Custom Language Support
- Framework integration
- Tool integration
- User interface enhancements or add-ons
- Themes

Plugin Development

- Alternative to Plugin
- Plugin Types
- **Plugin Structure**

- Plugin Configuration File – plugin.xml
- Plugin Dependencies – Any 3rd party lib
- Actions – Callbacks on User gestures
- Extensions – Extend IDE functionality
- Listeners – Subscribe to IDE events

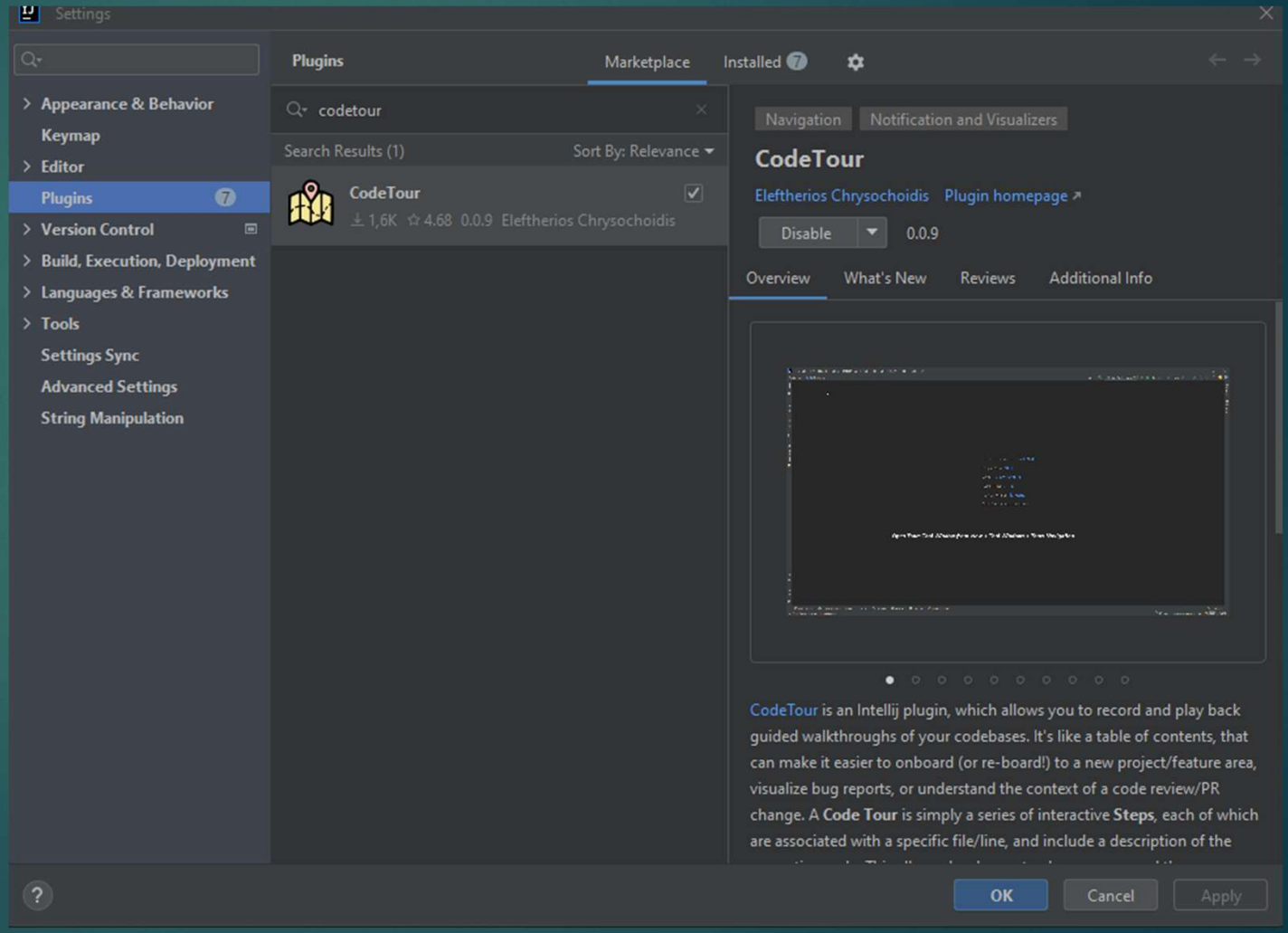
Hands On | CodeTour

- **What is CodeTour**
- Where to get it
- Demo

- Plugin which allows you to record and play back guided walkthroughs of your codebase.
- Markdown and HTML support for Step's description
- Might be useful for
  - Onboard (or re-board) new members on a project
  - Visualize Bugs (providing the path/flow)
  - Explain a Pull Request (explaining the logic flow)
  - Document a Feature (feature's flow is depicted on the code)

Hands On | CodeTour

- What is CodeTour
- **Where to get it**
- Demo



Hands On | CodeTour

- What is CodeTour
- **Where to get it**
- Demo

- CodeTour - IntelliJ IDEs Plugin | Marketplace
  - https://plugins.jetbrains.com/plugin/19227-codetour
- LefterisXris/CodeTour: IntelliJ Plugin | Github
  - https://github.com/LefterisXris/CodeTour

Hands On | CodeTour

- What is CodeTour
- Where to get it
- **Demo**

Resources

- **Github Template**
  - Setup a Complete Github Repo in minutes!
- Plugin Samples
  - Structure, Config, Testing
- Usage Explorer
- Extension Point and Listener List
  - Continuous integration based on GitHub Actions
    - Dependencies management with Dependabot
    - Changelog maintenance with the Gradle Changelog Plugin
    - Release flow using GitHub Releases
    - Plugin signing with your private certificate
    - Publishing the plugin with the Gradle IntelliJ Plugin
- UI Inspector
- Internal Actions
  - https://github.com/JetBrains/intellij-platform-plugin-template

Resources

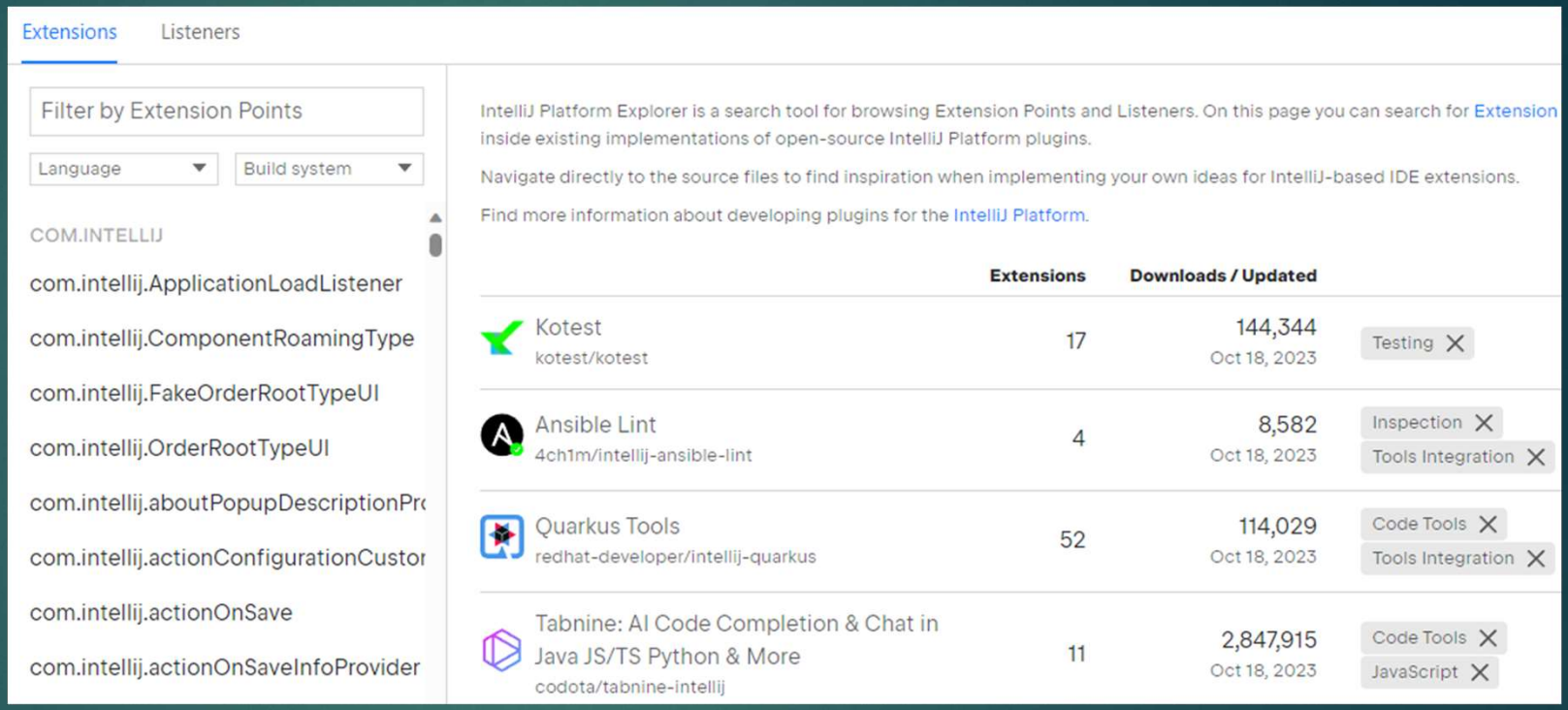
- Github Template
- **Plugin Samples**
- Usage Explorer
- Extension Point and Listener List
- UI Inspector
- Internal Actions

Code Sample	Description
Action Basics	Action and Action Group patterns implementation, adds entries to the Tools menu.
Comparing References Inspection	Local Inspection Tool, adds entries to **Settings | Editor | Inspections | Java | Probable Bugs**.
Conditional Operator Intention	Intention action, suggests converting a ternary operator into an `if` block and adds entry to **Settings | Editor | Intentions | SDK Intentions**.
Editor Basics	Basic Editor APIs example with editor popup menu with extra actions.
Framework Basics	Basic *SDK Demo Framework* support added to the **File | New | Project | Java** wizard.
Kotlin Demo	Kotlin example extending the *Main Menu* with a **Greeting** menu group.
Live Templates	Live templates for Markdown language, adds an entry to the **Settings | Editor | Live Templates** dialog.
Max Opened Projects	Application services and listeners, shows warning dialog when more than 3 open projects are opened.
Module	*SDK Demo Module* module type added to the **File | New | Project...** wizard.
Product Specific - PyCharm Sample	Plugin project configuration for the PyCharm IDE.
Project Model	Interacts with the project model, adds menu items to **Tools** and **Editor Context** menus.
Project View Pane	Project View Pane listing only image files.
Project Wizard	Project Wizard example with demo steps.
PSI Demo	PSI Navigation features presentation.
Run Configuration	Run configuration implementation with factory, options and UI.
Settings	Custom settings panel, adds a settings panel to the **Settings** panel under **Tools**.
Simple Language Plugin	Custom language support, defines a new *Simple language* with syntax highlighting, annotations, code completion, and other features.
Theme Basics	Sample *theme* plugin with basic interface modifications.
Tool Window	Custom Tool Window example plugin.
Tree Structure Provider	Tree Structure Provider showing only plain text files.

https://github.com/JetBrains/intellij-sdk-code-samples

Resources

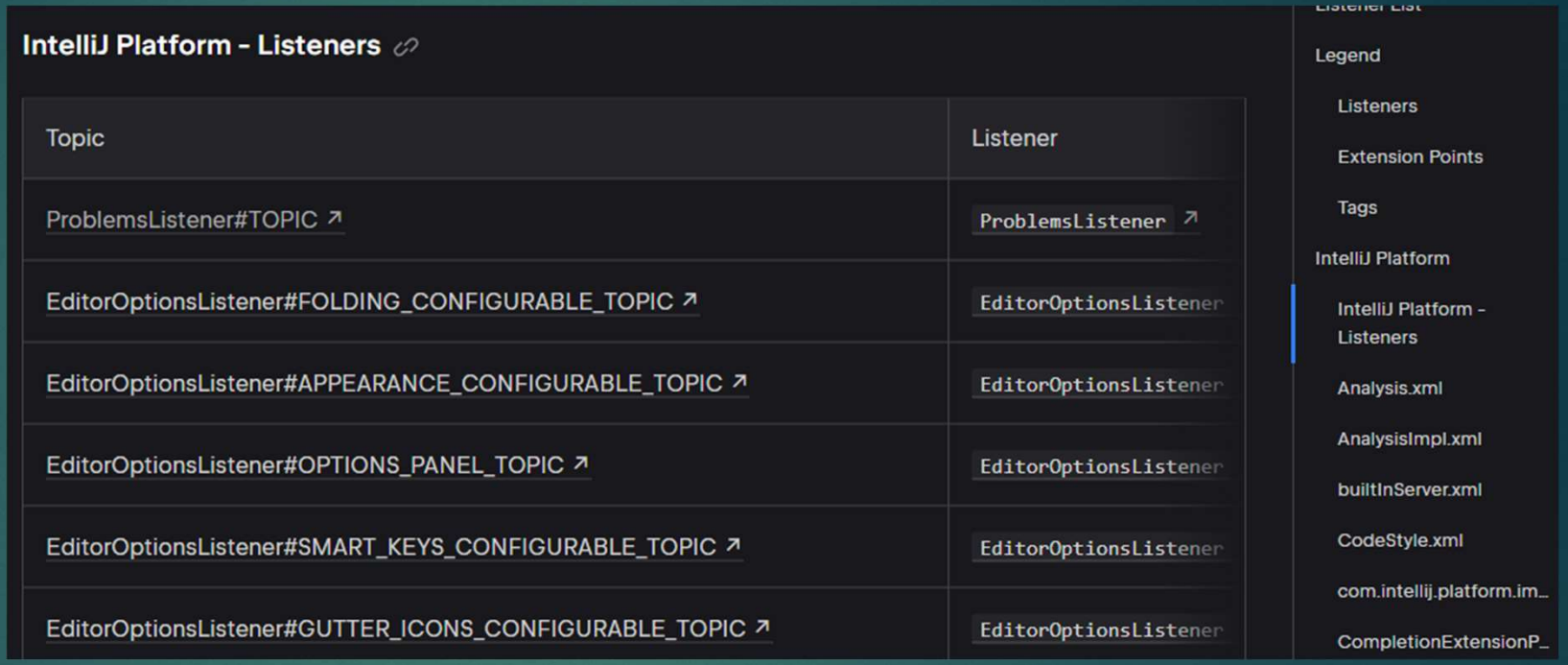
- Github Template
- Plugin Samples
- **Usage Explorer**
- Extension Point and Listener List
- UI Inspector
- Internal Actions



https://plugins.jetbrains.com/intellij-platform-explorer/extensions

Resources

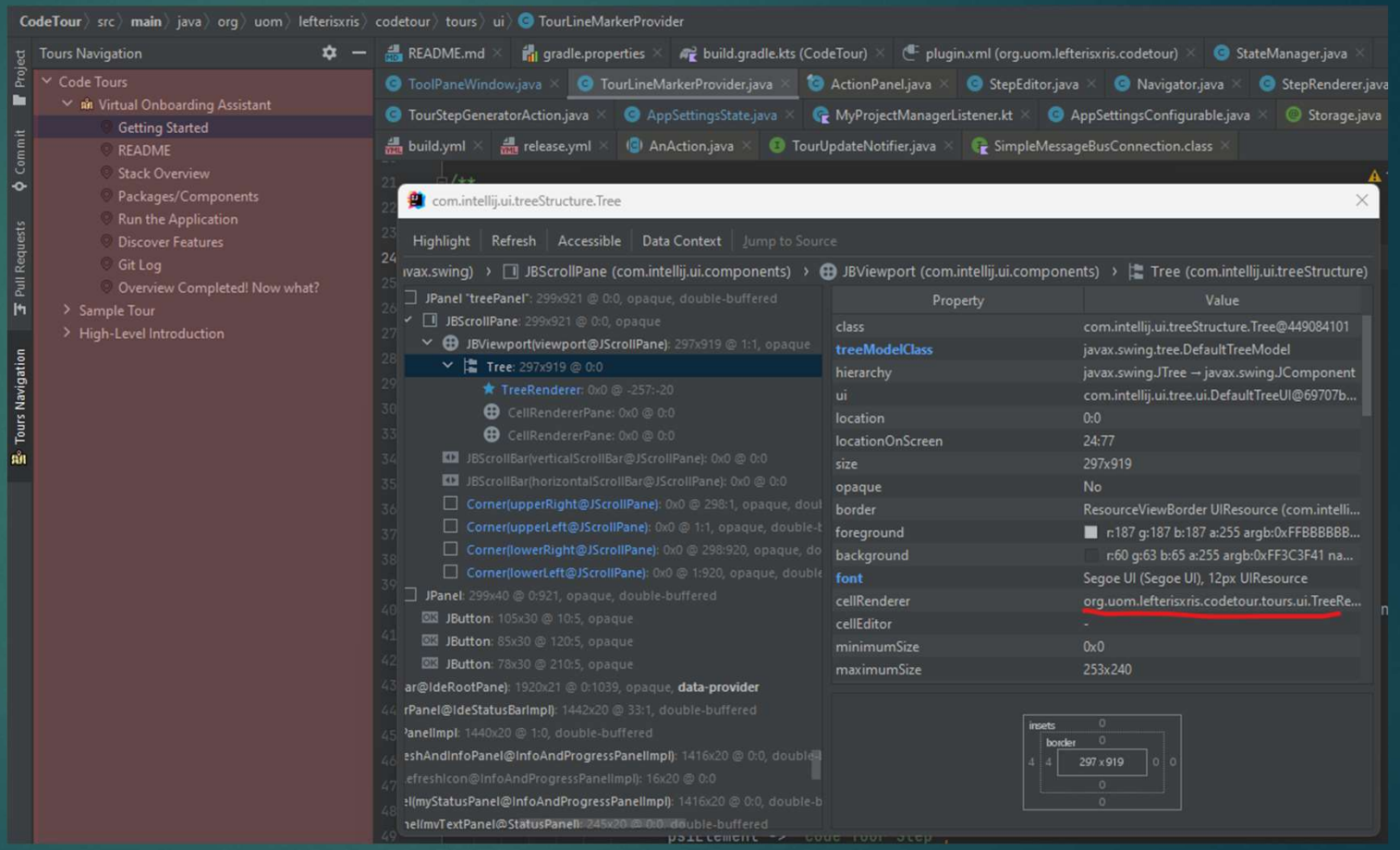
- Github Template
- Plugin Samples
- Usage Explorer
- **Extension Point and Listener List**
- UI Inspector
- Internal Actions



https://plugins.jetbrains.com/docs/intellij/extension-point-list.html

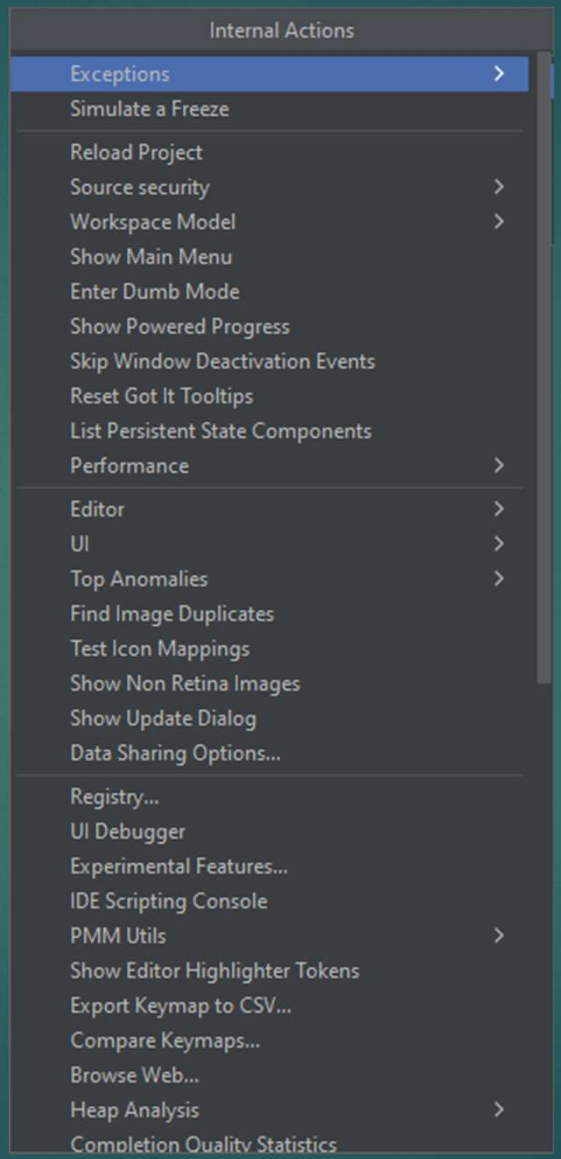
Resources

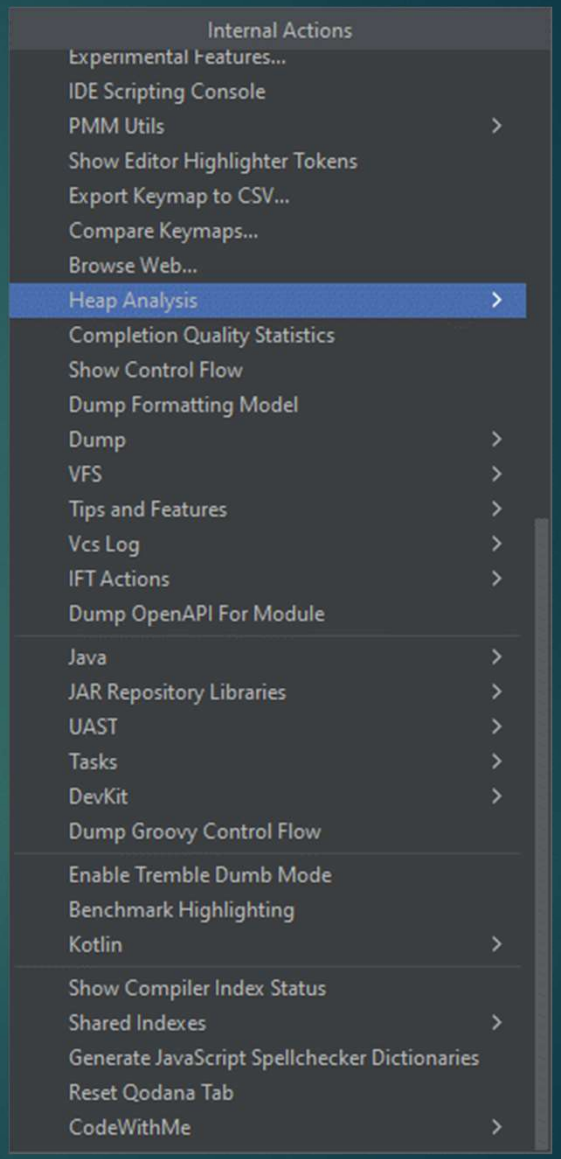
- Github Template
- Plugin Samples
- Usage Explorer
- Extension Point and Listener List
- **UI Inspector**
- Internal Actions



Resources

- Github Template
- Plugin Samples
- Usage Explorer
- Extension Point and Listener List
- UI Inspector
- **Internal Actions**





QA Time!