Building with AI in Kotlin

Márton Braun
@zsmb13

Sebastian Aigner
@sebi_io

JETBRAINS

Part 1

AI-assisted coding in Kotlin

Part 2

Building AI-powered apps in Kotlin

Levels of AI-assisted programming

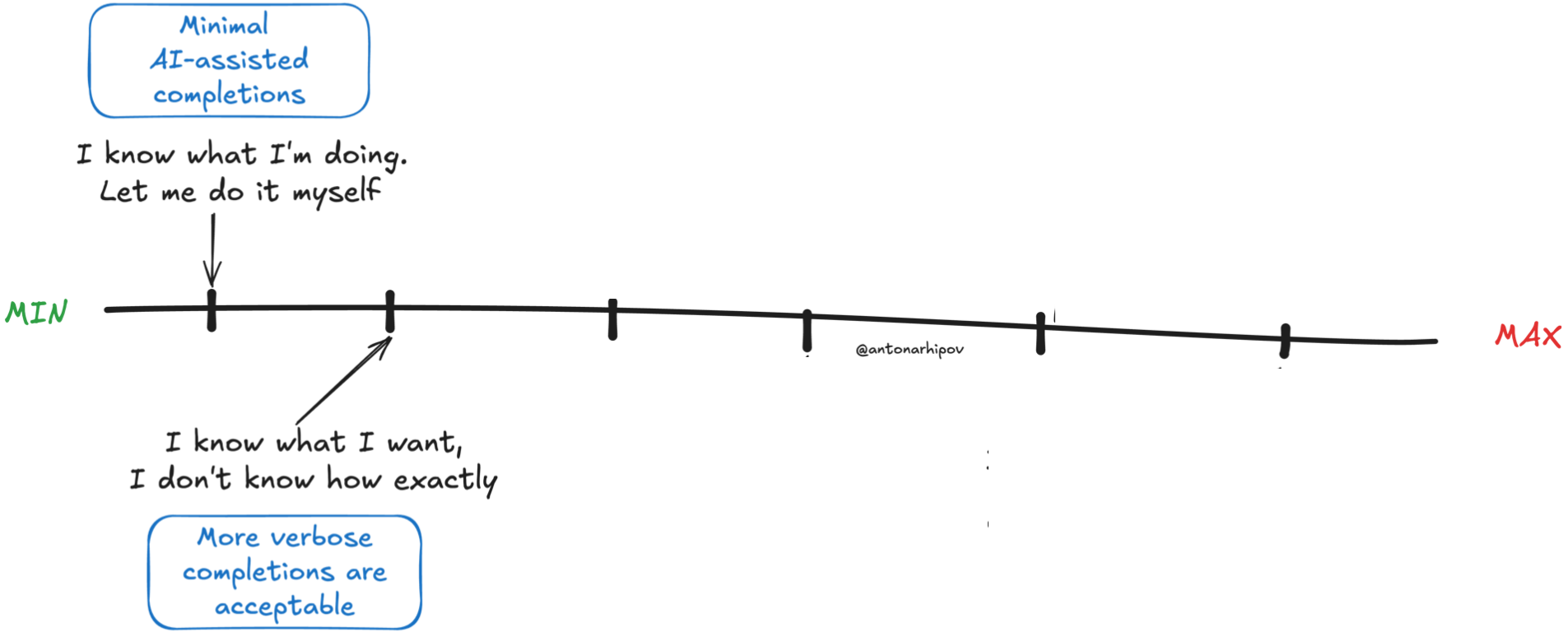
Levels of AI-assisted programming

MIN

MAX

Levels of AI-assisted programming

Levels of AI-assisted programming



Settings

- Appearance & Behavior
- Keymap
- Editor
  - General
    - Auto Import
    - Appearance
    - Breadcrumbs
    - Code Completion
    - Code Folding
    - Console
    - Editor Tabs
    - Gutter Icons
    - Inline Completion
    - Postfix Completion
    - Smart Keys
    - Sticky Lines
  - Code Editing
  - Font

Editor › General › Inline Completion

Go to Code Completion settings page to adjust lookup completion settings

- [x] Enable local Full Line completion suggestions

  Runs entirely on your **local** device without sending anything over the internet

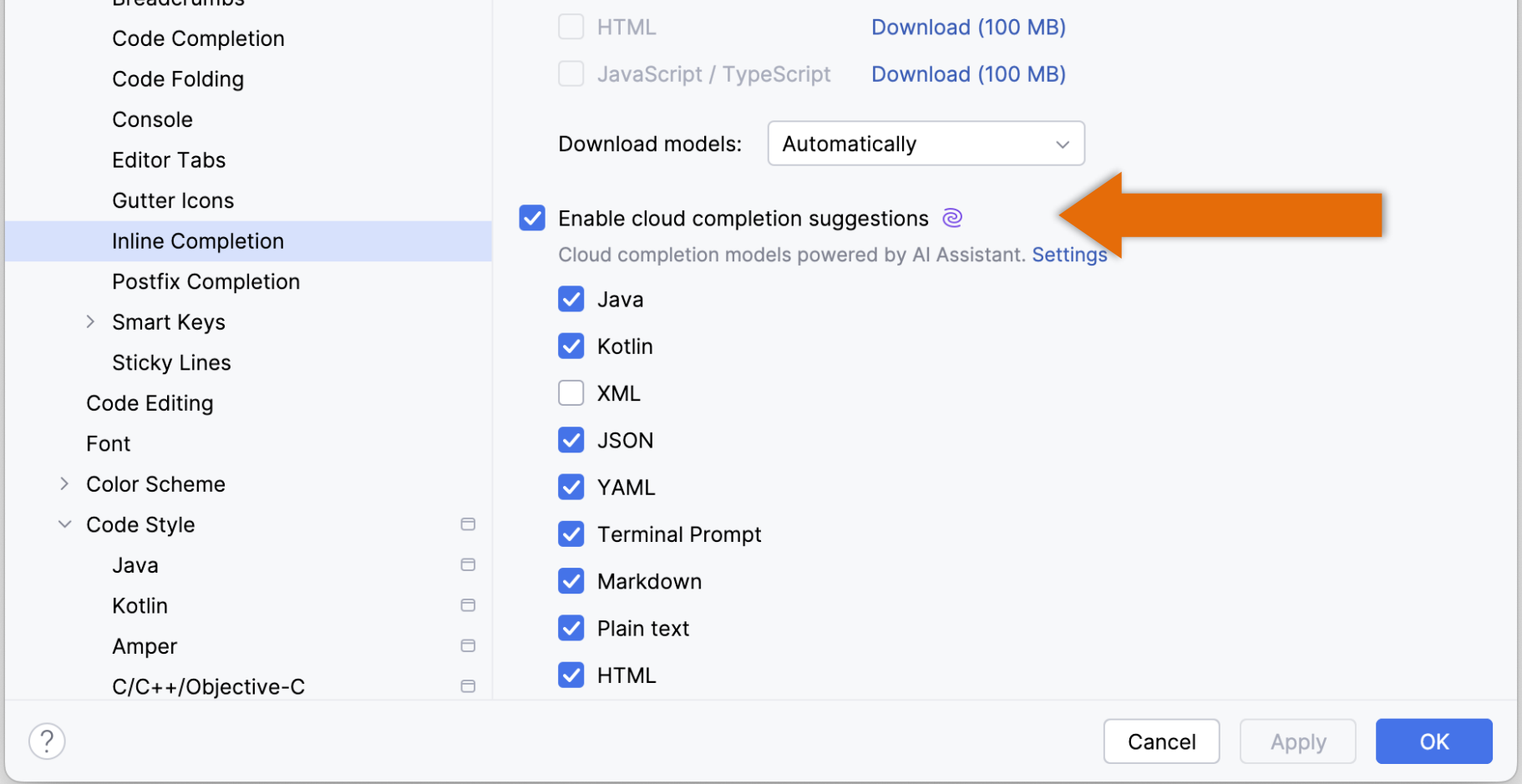
  - [x] Java
  - [x] Kotlin
  - [ ] CSS-like — Download (100 MB)
  - [ ] HTML — Download (100 MB)
  - [ ] JavaScript / TypeScript — Download (100 MB)

  Download models: Automatically

- [x] Enable cloud completion suggestions

  Cloud completion models powered by AI Assistant. Settings

  - [x] Java
  - [x] Kotlin
  - [ ] XML
  - [x] JSON


Breadcrumbs
Code Completion
Code Folding
Console
Editor Tabs
Gutter Icons
Inline Completion
Postfix Completion
Smart Keys
Sticky Lines
Code Editing
Font
Color Scheme
Code Style
Java
Kotlin
Amper
C/C++/Objective-C
HTML
Download (100 MB)
JavaScript / TypeScript
Download (100 MB)
Download models:
Automatically
Enable cloud completion suggestions
Cloud completion models powered by AI Assistant. Settings
Java
Kotlin
XML
JSON
YAML
Terminal Prompt
Markdown
Plain text
HTML
Cancel
Apply
OK

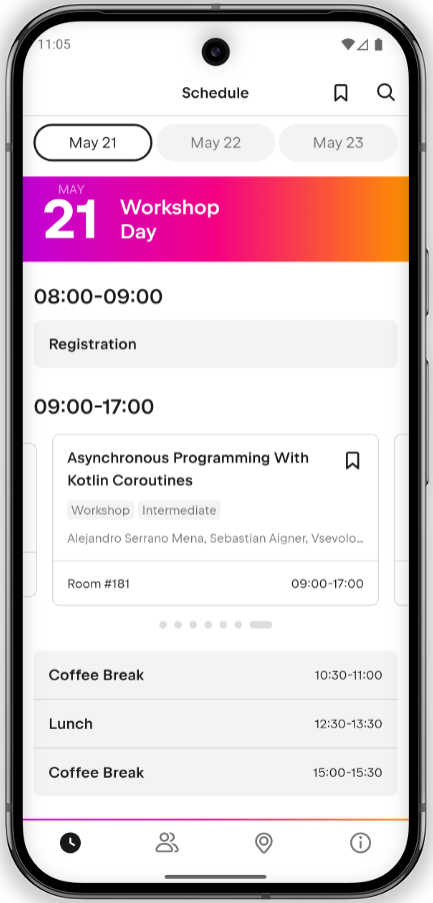
KotlinConfApp main shared [jvm]

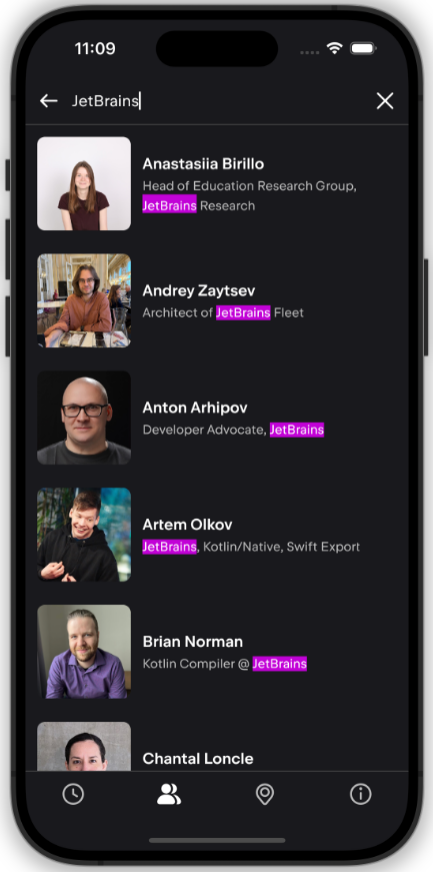
Button.kt

```kotlin
fun Button(
    val backgroundColor by animateColorAsState(
        if (primary) KotlinConfTheme.colors.primaryBackground
        else Color.Transparent
    )
    val borderColor by animateColorAsState(
        if (primary) Color.Transparent
        else KotlinConfTheme.colors.strokeHalf
    )

    val alpha by animateFloatAsState(if (enabled) 1f else 0.5f)

    Box(
        modifier = modifier
            .alpha(alpha)
            .heightIn(min = 56.dp)
            .border(width = 1.dp, color = borderColor, shape = ButtonShape)
            .clip(ButtonShape)
            .then(if (enabled) Modifier.clickable(onClick = onClick) else Modifier)
            .semantics { role = Role.Button }
            .background(backgroundColor)
            .padding(horizontal = 32.dp),
```

45:6 LF UTF-8

github.com/JetBrains/kotlinconf-app





Nintendo
ENTERTAINMENT SYSTEM
POWER
RESET
Nintendo
SELECT
START
B
A

Next

Edit

Suggestions

Settings

Tools › AI Assistant Revert changes

- Appearance & Behavior
- Keymap
- Editor
- Plugins
- Version Control
- Build, Execution, Deployment
- Languages & Frameworks
- Tools
  - Actions on Save
  - AI Assistant
  - Code With Me
  - CSV Formats
  - Database
  - Database Versioning
  - Diagrams
  - Diff & Merge
  - External Tools
  - Features Suggester
  - Features Trainer
  - HTTP Client
  - IntelliJ API Watcher
  - JPA Entity Declaration
  - JPA Reverse Engineering
  - Junie

Features

- [ ] Enable cloud completion suggestions
  Go to Inline Completion settings page to fine-tune completion suggestions
- [x] Enable next edit suggestions Beta
  AI Assistant will suggest code modifications as you type

  Languages
  - [x] Java
  - [x] Kotlin
  - [ ] JavaScript / TypeScript Alpha
- [ ] Instant Preview mode Alpha
  Show suggestions instantly, with fewer confirmation prompts
- [x] Chain suggestions Alpha
  Automatically request another suggestion after accepting one
- [x] Code Insight actions Alpha
  Suggest code insight actions (e.g., refactorings)
- [ ] Allow attaching database schemas to AI Assistant chat
  AI Assistant gets access to the structure of the attached database schemas. It improves the quality of generated database queries. Note: this leads to faster consumption of your AI Assistant volume.
- [x] Provide AI-generated name suggestions
- [ ] Suggest converting pasted code to the language of the target file
- [x] Generate a title for the shelved changelist

Cancel Apply OK

KotlinConfApp main shared [jvm]

MapScreen.kt

```
192 private fun Map(
204     val scope = rememberCoroutineScope()
205
206     val validOffsetX = (-svg.width * 0.25f) ≤ .. ≤ (svg.width * 0.25f)
207     val validOffsetY = (-svg.height * 0.25f) ≤ .. ≤ (svg.height * 0.25f)
208
209     val interactiveModifiers = if (!interactive) {
210         Modifier
211     } else {
212         Modifier
213             .transformable(rememberTransformableState { zoomChange, panChange, _ →
214                 if (!interactive) return@rememberTransformableState
215
216                 scope.launch { this: CoroutineScope
217                     scale.snapTo((scale.value * zoomChange).coerceIn(zoomRange))
218                     offsetX.snapTo((offsetX.value + panChange.x / scale.value).coerceIn(validOffsetX))
219                     offsetY.snapTo((offsetY.value + panChange.y / scale.value).coerceIn(validOffsetY))
220                 }
221             })
222             .pointerInput(Unit) { this: PointerInputScope
223                 if (!interactive) return@pointerInput
224
```

5 3

206:41 LF UTF-8

KotlinConfApp main shared [jvm]

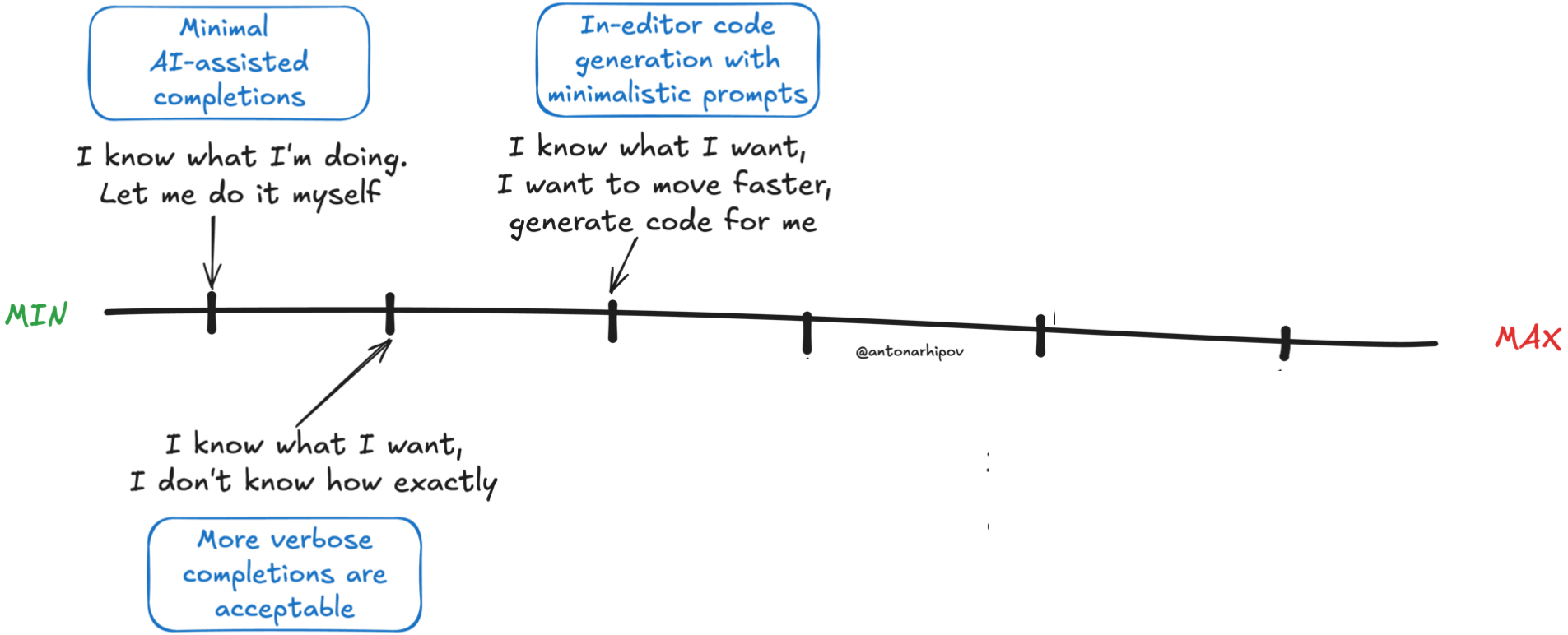
MapScreen.kt

```
192 private fun Map(
204     val scope = rememberCoroutineScope()
205
206     val xRange = (-svg.width * 0.5f) ≤ .. ≤ (svg.width * 0.5f)
207     val yRange = (-svg.height * 0.5f) ≤ .. ≤ (svg.height * 0.5f)
208
209     val interactiveModifiers = if (!interactive) {
210         Modifier
211     } else {
212         Modifier
213             .transformable(rememberTransformableState { zoomChange, panChange, _ →
214                 if (!interactive) return@rememberTransformableState
215
216                 scope.launch { this: CoroutineScope
217                     scale.snapTo((scale.value * zoomChange).coerceIn(zoomRange))
218                     offsetX.snapTo((offsetX.value + panChange.x / scale.value).coerceIn(xRange))
219                     offsetY.snapTo((offsetY.value + panChange.y / scale.value).coerceIn(yRange))
220                 }
221             })
222             .pointerInput(Unit) { this: PointerInputScope
223                 if (!interactive) return@pointerInput
224
```

2 5 3

219:95 LF UTF-8

Levels of AI-assisted programming



KotlinConfApp main shared [jvm]

ConferenceService.kt

```kotlin
class ConferenceService(
    suspend fun loadConferenceData() {
        val newData = client.downloadConferenceData() ?: return
        val oldData = storage.getConferenceCache().first()

        storage.setConferenceCache(newData)


    }

    fun isOnboardingComplete(): Flow<Boolean> {
        return storage.isOnboardingComplete()
    }

    suspend fun completeOnboarding() {
        storage.setOnboardingComplete(true)
    }

    suspend fun acceptPrivacyNotice(): Boolean {
        val userId = storage.getUserId().first()
        if (userId != null) return true
```

139:1 LF UTF-8

```kotlin
package org.jetbrains.kotlinconf.utils

import ...

private val diacriticsRegex = "\\p{Mn}+".toRegex()

fun String.containsDiacritics(): Boolean = normalizeNfd().contains(diacriticsRegex)  Márton Braun

fun String.removeDiacritics() = normalizeNfd().replace(diacriticsRegex, "")  Márton Braun

private fun String.no                                    ng = normalize(Form.NFD)  Márton Braun
```

AI Actions

- Explain Code
- Suggest Refactoring
- Find Problems
- New Chat Using Selection
- Write Documentation
- Generate Unit Tests
- Generate Code... ⌘\

Add your prompts...

KotlinConfApp | kc-522-no-bookmark-items | shared [jvm]

Commit | Shelf | Stash

Changes 2 files

- ScheduleScreen.kt shared/src/commonMain/kotlin/org/jetbrains/kotlinconf/screens
- ScheduleViewModel.kt shared/src/commonMain/kotlin/org/jetbrains/kotlinconf/screens

Unversioned Files 3 files

Amend

2 modified

Commit Message

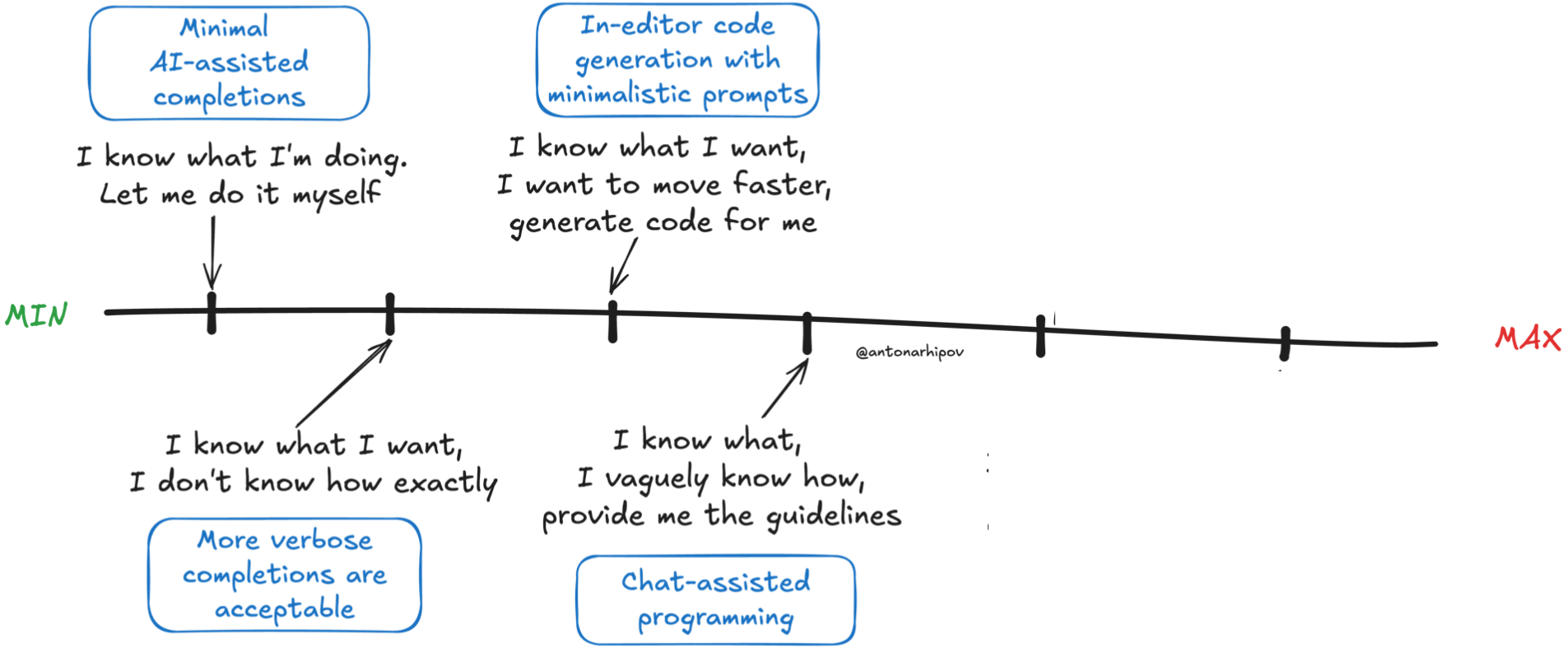
Commit | Commit and Push...

ScheduleViewModel.kt

```
38  data class TimeSlotTitleItem(val
     value: TimeSlot) : ScheduleListItem
39
40  data class NoBookmarksItem(val id:
     String) : ScheduleListItem
41
42  data class SessionItem(
43      val value: SessionCardView,
44      val tagMatches: List<String> =
         emptyList(),
45      val titleHighlights:
         List<IntRange> = emptyList(),
46      val speakerHighlights:
         List<IntRange> = emptyList(),
47  ) : ScheduleListItem
48
49  data class ServiceEventItem(
50      val value: SessionCardView,
51  ) : ScheduleListItem
52
53  data class ServiceEventGroupItem(
```

41:1 LF UTF-8

Levels of AI-assisted programming



KotlinConfApp main shared [jvm]

APIClient.kt

```kotlin
class APIClient(

    var userId: String? = null

    private val client = HttpClient {
        install(ContentNegotiation) {
            json()
        }

        expectSuccess = true

        install(DefaultRequest) {
            url.takeFrom(apiUrl)
        }
    }

    /**
     * @return status of request.
     */
    suspend fun sign(userId :String): Boolean {
        return safeApiCall {
```

AI Chat

New Chat

Multiline code completion ⌥ ⇧ \

Code generation in the editor ⌘ \

AI actions in the editor's context menu

All features

I want to follow best practices for Ktor client setup, do I need to fix anything?

Codebase APIClient.kt Current

Chat Claude 3.7 Sonnet

Share feedback

47:38 LF UTF-8

KotlinConfApp main shared [jvm]

APIClient.kt

Accept All Discard All 0/1

```kotlin
class APIClient(
    private val client = HttpClient {

        install(HttpTimeout) {
            requestTimeoutMillis = 15000  // 15 seconds
            connectTimeoutMillis = 15000
            socketTimeoutMillis = 15000
        }

        install(HttpRequestRetry) {
            retryOnServerErrors(maxRetries = 3)
            exponentialDelay()
        }

        install(Logging) {
            logger = object : KtorLogger {
                override fun log(message: String) {
                    appLogger.log(LOG_TAG) {
                     message }
```

AI Chat

+ New Chat

Kotlin Report Feedback Apply β

```kotlin
private val client = HttpClient {
    install(ContentNegotiation) {
        json()
    }

    install(H
        reque
        conne
        socke
    }

    install(H
        retry
        expon
    }
```

Model

- Claude 3.7 Sonnet
- Claude 4 Sonnet
- Gemini 2.5 Flash
- Gemini 2.5 Pro Beta
- GPT-4o
- GPT-5 New
- GPT-5 mini New
- GPT-5 nano New
- GPT-4.1
- o3
- More Models

Local Models

Set Up Local Models...

Ask AI Assistant, use @

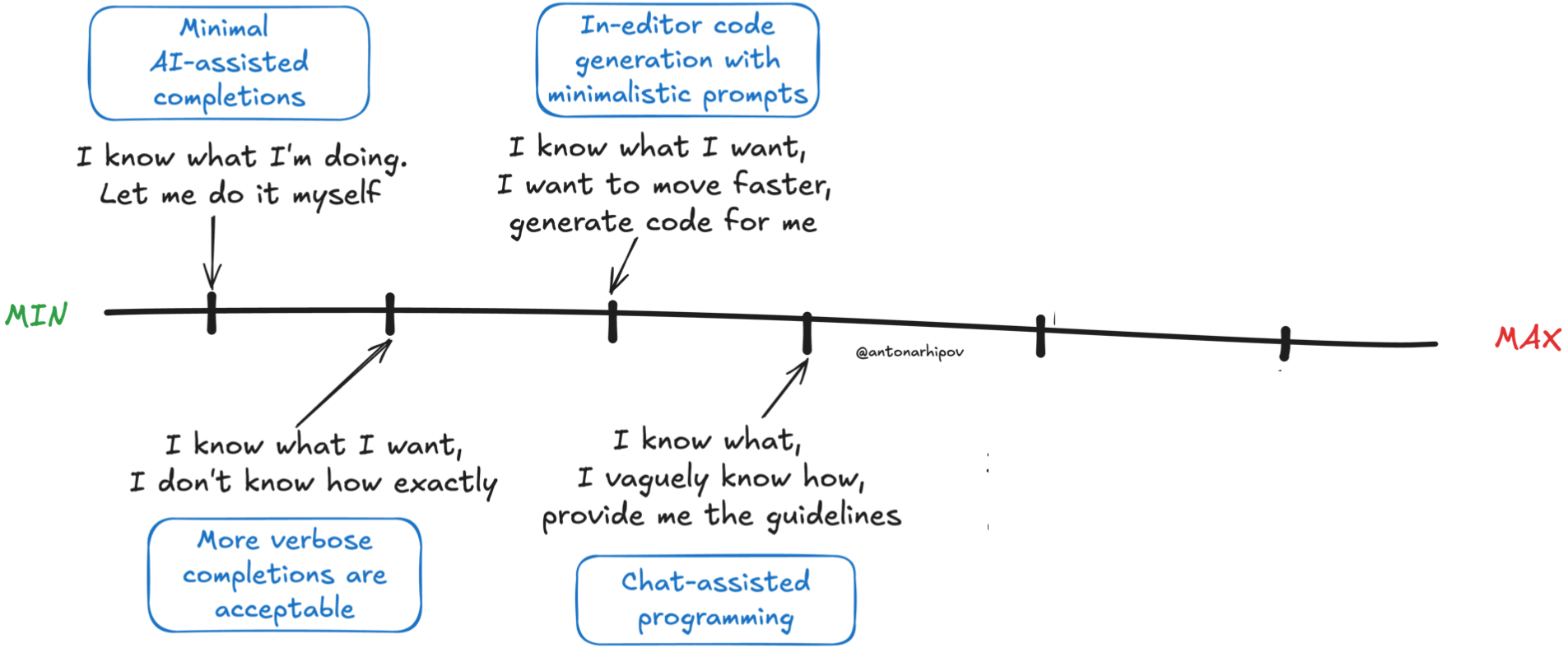
Codebase

Chat Claude 3.7 Sonnet

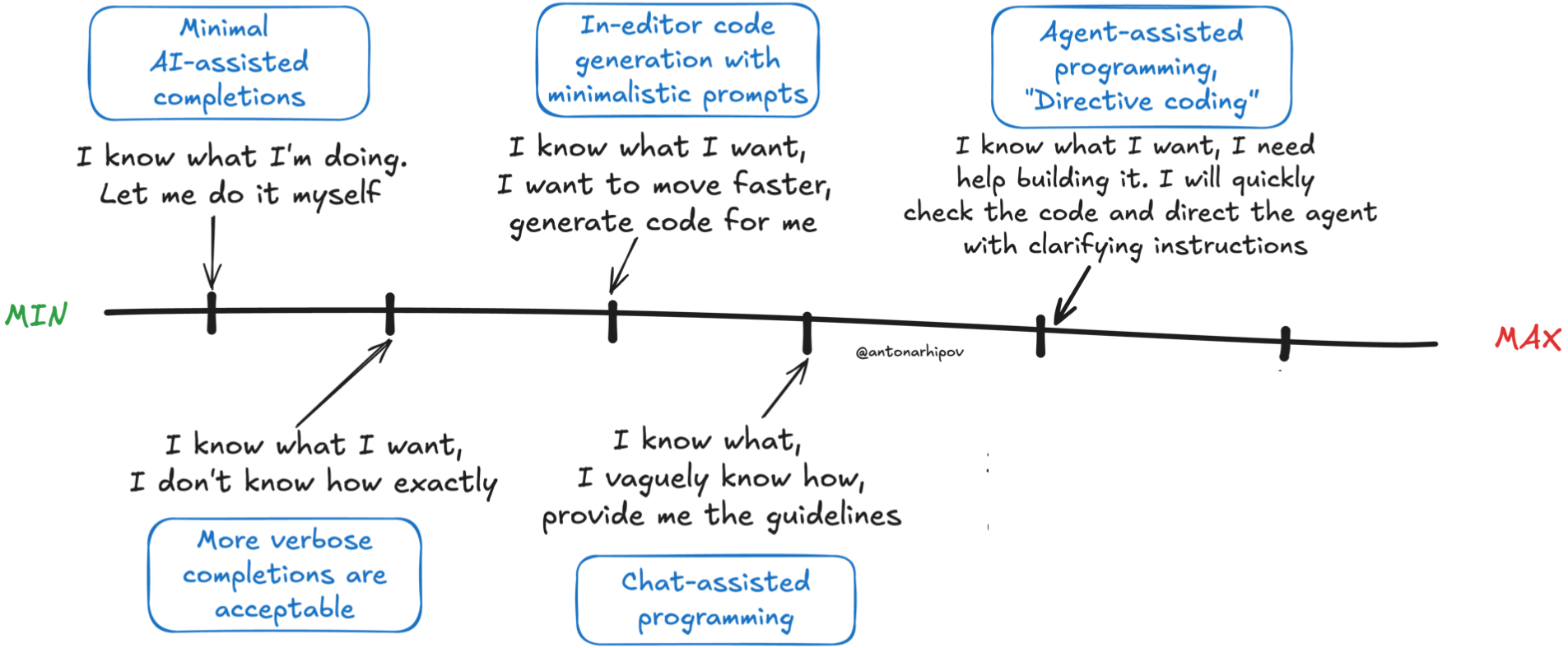
Share feedback ↗

47:38 LF UTF-8

Levels of AI-assisted programming



Levels of AI-assisted programming



Junie
by JETBRAINS

KotlinConfApp main shared [hotRunJvm]

AboutAppScreen.kt FlagsManager.kt

```kotlin
import ...

class FlagsManager(
    private val platformFlags: Flags,
    private val storage: ApplicationStorage,
    private val scope: CoroutineScope,
) {

    val flags = storage.getFlags()
        .map { it ?: platformFlags }
        .stateIn(scope, SharingStarted.Eagerly, platformFlags)

    init {
        scope.launch {
            val storedFlags = storage.getFlags().first()
            if (storedFlags == null) {
                storage.setFlags(platformFlags)
            }
        }
    }

```

11:20 LF UTF-8

KotlinConfApp main shared [hotRunJvm]

AboutAppScreen.kt FlagsManage

```
44  @Composable
45  fun AboutAppScreen(
46      onBack: () → Unit,
47      onGitHubRepo: () → Unit,
48      onRateApp: () → Unit,
49      onPrivacyNotice: () → Unit,
50      onTermsOfUse: () → Unit,
51      onLicenses: () → Unit,
52      onJunie: () → Unit,
53  ) {
54      ScreenWithTitle(
55          title = stringResource(Res.string.about_app_titl e),
```

Junie EAP

Junie

Junie is a Coding Agent.
It understands the context of the whole project.
This makes it autonomous, and gives it ability to solve big coding tasks.

Implement the new developer settings screen.

* This should be a new screen in the app
* The screen should have a list of toggles where the user can change the values of Flags using FlagsManager
* Use existing UI components where possible
* Add the version number of the app to the bottom of the About screen
* Tapping the version number 5 times should navigate to the developer menu

Code Brave Smarter

Ask Junie to:

Generate project-specific guidelines

Mastering Junie

Junie Getting Started Guide
Coding Guidelines for Your AI Agents

45:20 LF UTF-8

KotlinConfApp main shared [hotRunJvm]

3x speed

AboutAppScreen.kt FlagsManage

```
@Composable
fun AboutAppScreen(
    onBack: () → Unit,
    onGitHubRepo: () → Unit,
    onRateApp: () → Unit,
    onPrivacyNotice: () → Unit,
    onTermsOfUse: () → Unit,
    onLicenses: () → Unit,
    onJunie: () → Unit,
) {
    ScreenWithTitle(
        title = stringResource(Res.string.about_app_title),
```

Junie EAP

Junie

Junie is a Coding Agent.
It understands the context of the whole project.
This makes it autonomous, and gives it ability to solve big coding tasks.

Implement the new developer settings screen.

* This should be a new screen in the app
* The screen should have a list of toggles where the user can change the values of Flags using FlagsManager
* Use existing UI components where possible
* Add the version number of the app to the bottom of the About screen
* Tapping the version number 5 times should navigate to the developer menu

Code Brave Smarter

Ask Junie to:

Analyse requirements and create an improvement plan

Mastering Junie

Junie Getting Started Guide
Coding Guidelines for Your AI Agents

45:20 LF UTF-8

KotlinConfApp main shared [hotRunJvm]

AboutAppScreen.kt FlagsManage

```
44  @Composable
45  fun AboutAppScreen(
46      onBack: () → Unit,
47      onGitHubRepo: () →
          Unit,
48      onRateApp: () →
         Unit,
49      onPrivacyNotice: ()
          → Unit,
50      onTermsOfUse: () →
          Unit,
51      onLicenses: () →
         Unit,
52      onJunie: () → Unit,
53
    onOpenDeveloperSetting
    s: () → Unit,
54  ) {
55      ScreenWithTitle(
56          title =
             stringResource
```

Junie EAP

Develop Developer Settings Screen with Toggles and Version Navigation

Done Start again Decline

components.
4. Implement a new DeveloperSettingsScreen composable listing all Flags with SettingsItem toggles, wired to FlagsManager.
5. Add a new route DeveloperSettingsScreen to Routes.kt and register a composable destination in KotlinConfNavHost.kt, navigating on back and from About screen.
6. Update AboutAppScreen to show app version at the bottom and implement "tap 5 times" to navigate by passing a new onOpenDeveloperSettings callback and

Done

Changed Files

AboutAppScreen.kt +19
KotlinConfNavHost.kt +5
Routes.kt +4

New Files Add all to Git

DeveloperSettingsScreen.kt +85

A new Developer Settings screen was successfully implemented, featuring toggles for various flags using FlagsManager. Additionally, the About screen was updated to display the app version at the bottom, and a tap gesture was added to navigate to the Developer menu. No errors were encountered during the implementation process.

Rollback

Start New Task

Type your task here, press ⏎

+ Code Ask Brave Mode Think More

Run selected configuration 45:20 LF UTF-8

KotlinConfApp main shared [hotRunJvm]

AboutAppScreen.kt FlagsManager.kt

```kotlin
@Composable
fun AboutAppScreen(
    onBack: () -> Unit,
    onGitHubRepo: () -> Unit,
    onRateApp: () -> Unit,
    onPrivacyNotice: () -> Unit,
    onTermsOfUse: () -> Unit,
    onLicenses: () -> Unit,
    onJunie: () -> Unit,
    onOpenDeveloperSettings: () -> Unit,
) {
    ScreenWithTitle(
        title = stringResource(Res.string.about_app_title),
        onBack = onBack,
    ) {
        Column(verticalArrangement = Arrangement.spacedBy(8.dp)) {
            Text(
                stringResource(Res.string.about_app_description),
                color = KotlinConfTheme.colors.longText,
                modifier = Modifier.padding(vertical = 24.dp),
            )
```

Junie EAP

Develop Developer Settings Screen with Toggles and Version Navigation

Done Start ... D

Done

Changed Files

AboutAppScreen.kt +19
KotlinConfNavHost.kt +5
Routes.kt +4

New Files Add all to Git

DeveloperSettingsScreen… +85

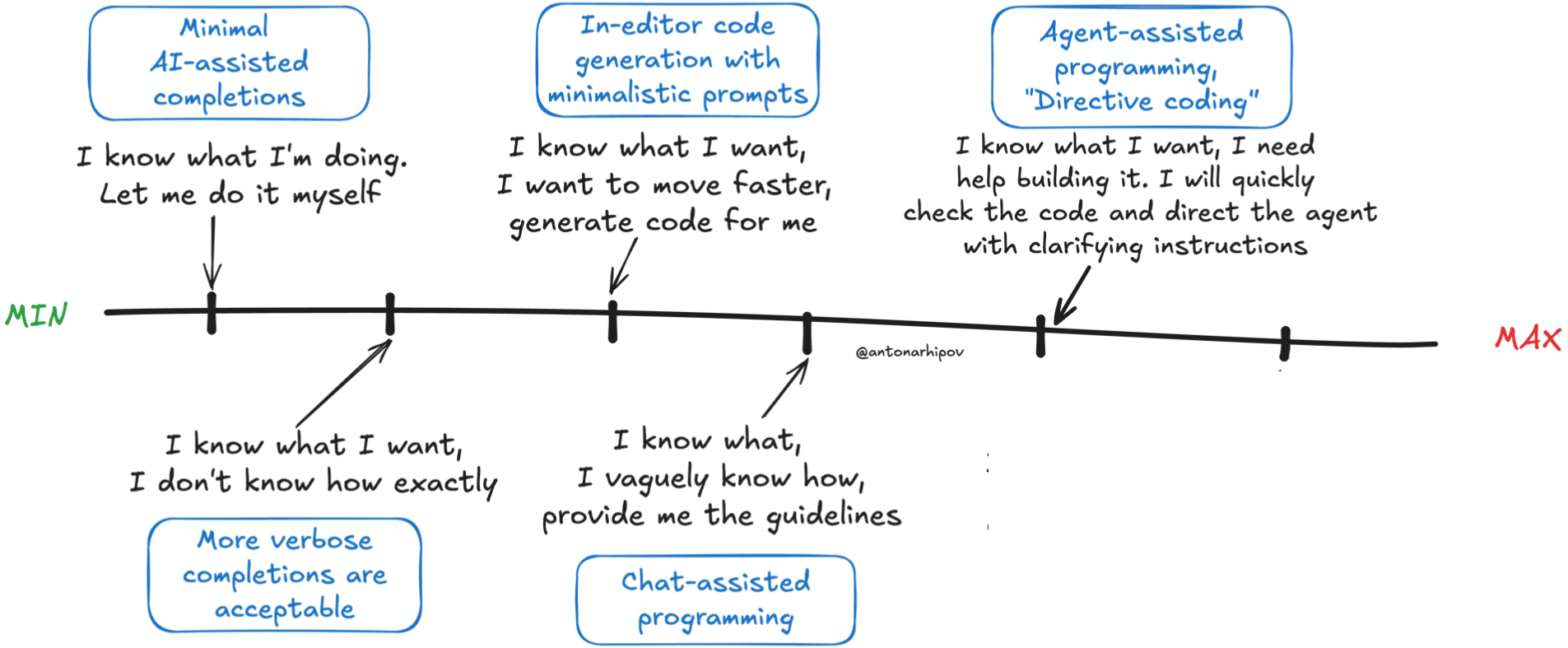
A new Developer Settings screen was successfully implemented, featuring toggles for various flags using FlagsManager. Additionally,

Type your task here, press ⏎

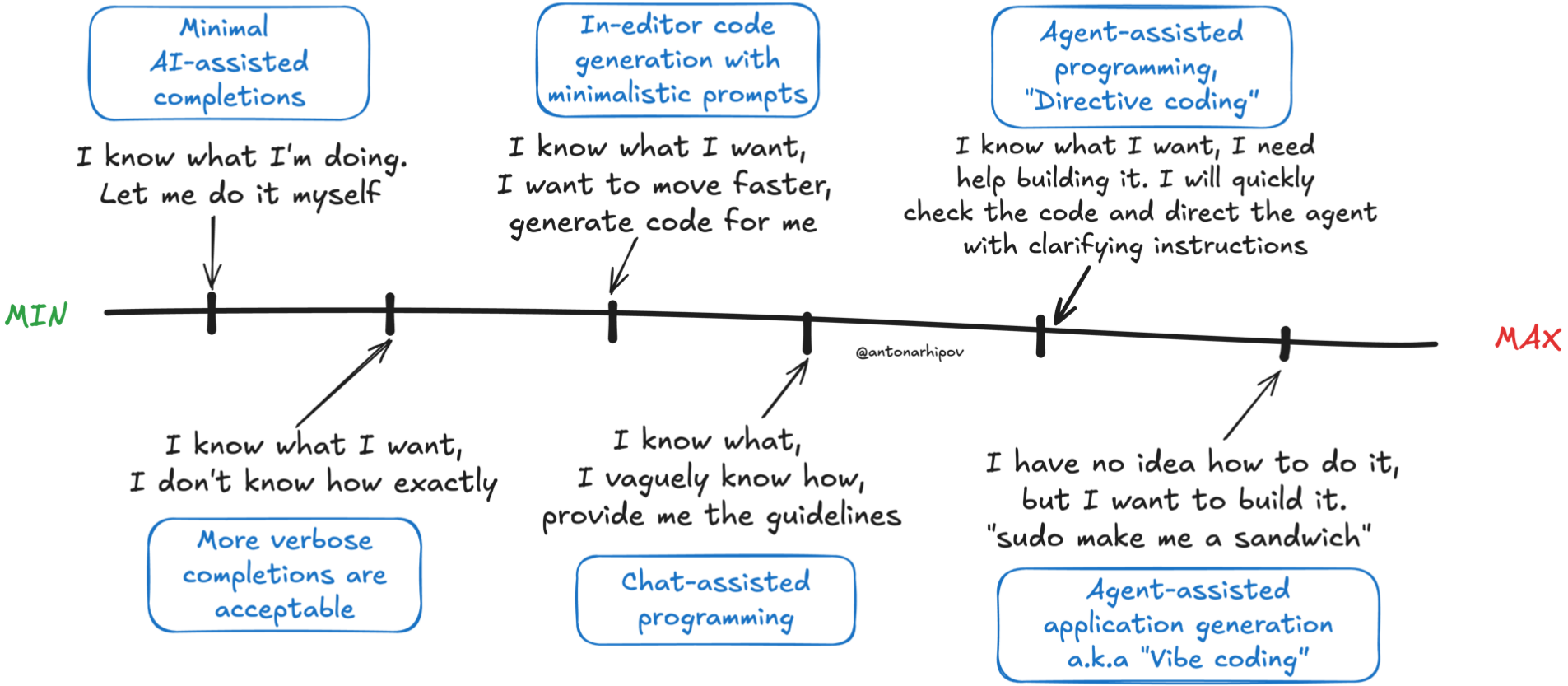
Code Brave Smart

45:20 LF UTF-8

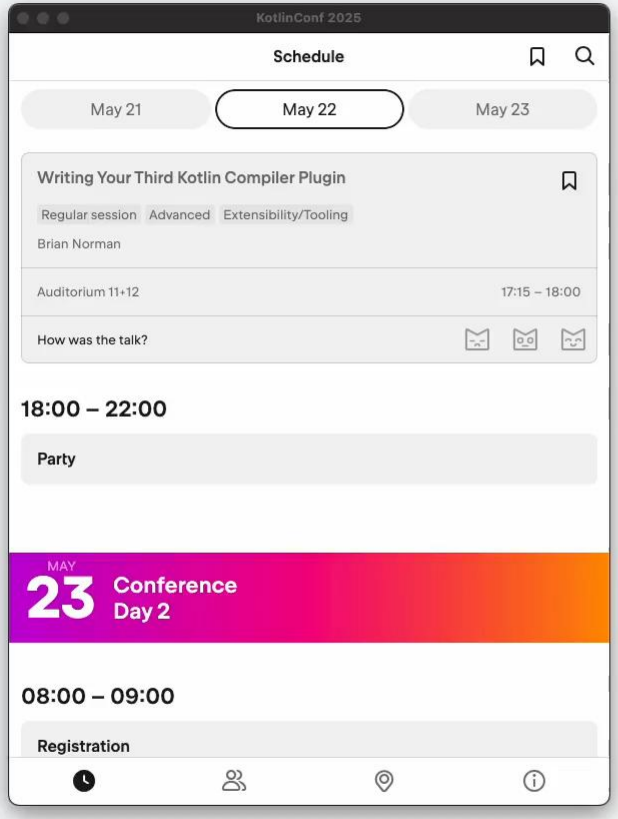
Levels of AI-assisted programming



Levels of AI-assisted programming



KotlinConfApp main shared [hotRunJvm]



Search Everywhere Double ⇧

Project View ⌘1

Go to File ⇧⌘O

Recent Files ⌘E

Navigation Bar ⌘↑

Drop files here to open them

Junie EAP

Junie is a Coding Agent.
It understands the context of the whole project.
This makes it autonomous, and gives it ability to solve big coding tasks.

Add a fun animation to the party item in the schedule with Kodee

Code Brave Smarter

Tasks Mastering Junie

Recent

- [ ] Develop Developer Settings Screen with Toggles and...

KotlinConfApp main

4x speed

shared [hotRunJvm]

KotlinConf 2025

Schedule

May 21 | May 22 | May 23

Writing Your Third Kotlin Compiler Plugin

Regular session Advanced Extensibility/Tooling

Brian Norman

Auditorium 11+12 17:15 – 18:00

How was the talk?

18:00 – 22:00

Party

MAY 23 Conference Day 2

08:00 – 09:00

Registration

Search Everywhere Double ⇧

Project View ⌘1

Go to File ⇧⌘O

Recent Files ⌘E

Navigation Bar ⌘↑

Drop files here to open them

Junie EAP

Junie is a Coding Agent.
It understands the context of the whole project.
This makes it autonomous, and gives it ability to solve big coding tasks.

Add a fun animation to the party item in the schedule with Kodee

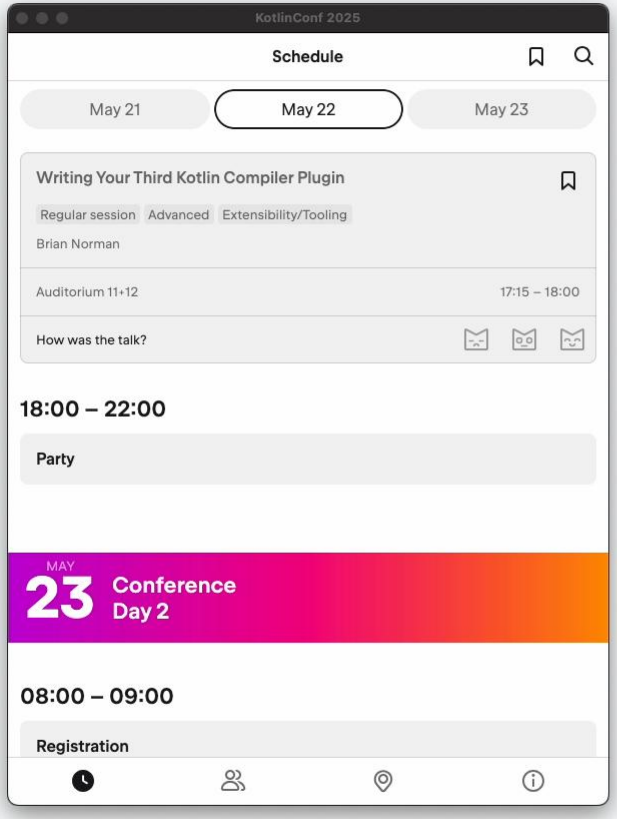
Code Brave Smarter

Tasks Mastering Junie

Recent

Develop Developer Settings Screen with Toggles and...

KotlinConfApp main shared [hotRunJvm]



Search Everywhere Double ⇧

Project View ⌘1

Go to File ⇧⌘O

Recent Files ⌘E

Navigation Bar ⌘↑

Drop files here to open them

Junie EAP

"Animate Party Item with Kodee"

Stop

Show Plan

The search for service event implementations found relevant components, indicating a potential for integrating the missing party item with Kodee. Further investigation is needed.

Open ScheduleScreen.kt

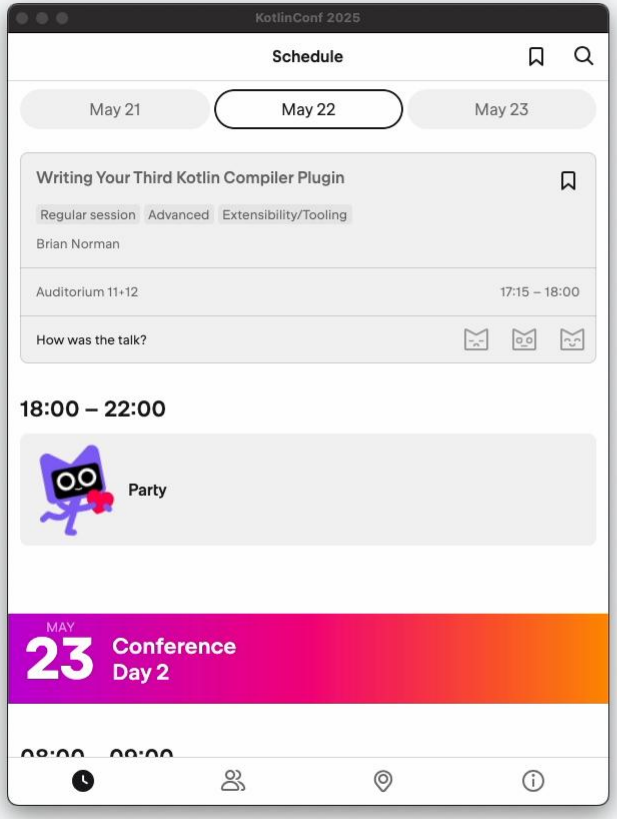
The ScheduleScreen.kt file was examined to understand how service events are rendered. The next steps involve identifying where to incorporate Kodee animation for the party item in the rendering process.

Edit ServiceEvents.kt +8

The ServiceEvent component has been modified to include Kodee animation for event titles containing "Party." The update enhances the presentation of service events related to parties.

Working

KotlinConfApp main shared [hotRunJvm]



Search Everywhere Double ⇧

Project View ⌘1

Go to File ⇧⌘O

Recent Files ⌘E

Navigation Bar ⌘↑

Drop files here to open them

Junie EAP

"Animate Party Item with Kodee"

Done Start again Decline

Done

Changed Files

ServiceEvents.kt +40

A fun animation featuring Kodee was successfully added to the party item in the schedule, incorporating a bouncing scale effect and rotation. The animation activates when the event title mentions "Party," enhancing the visual appeal of the schedule. No errors or issues were reported during the implementation.

Rollback

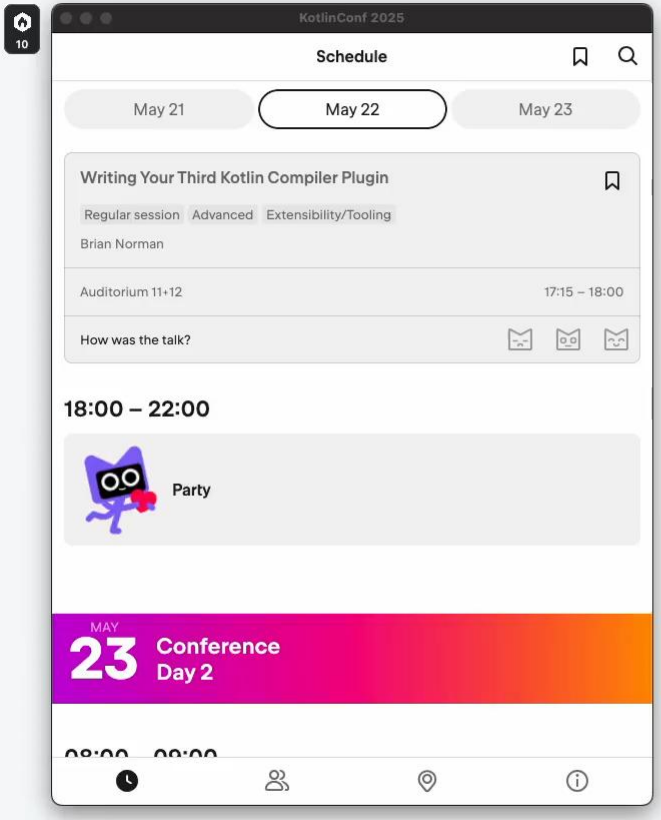
Start New Task

Type your task here, press ⏎

Code Brave Smarter

KotlinConfApp main shared [hotRunJvm]

3x speed



Search Everywhere Double ⇧

Project View ⌘1

Go to File ⇧⌘O

Recent Files ⌘E

Navigation Bar ⌘↑

Drop files here to open them

Junie EAP

"Animate Party Item with Kodee"

Stop

Show Plan

infinite transition, scale, and rotate animations has been successfully added.

Done

Changed Files

ServiceEvents.kt +40

A fun animation featuring Kodee was successfully added to the party item in the schedule, incorporating a bouncing scale effect and rotation. The animation activates when the event title mentions "Party," enhancing the visual appeal of the schedule. No errors or issues were reported during the implementation.

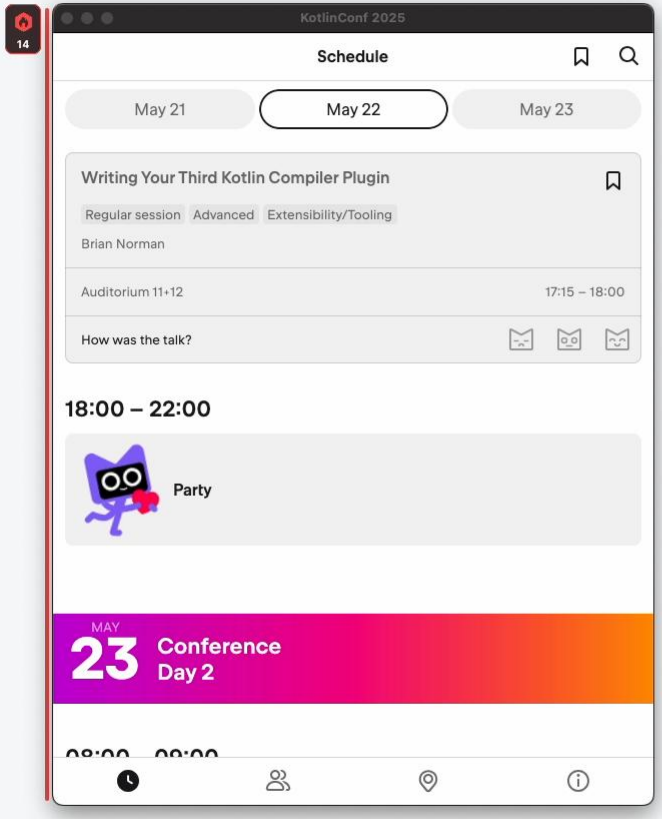
Add confetti

Terminal

Open

Initializing Terminal Session

KotlinConfApp main shared [hotRunJvm]



Search Everywhere Double ⇧

Project View ⌘1

Go to File ⇧⌘O

Recent Files ⌘E

Navigation Bar ⌘↑

Drop files here to open them

Junie EAP

"Animate Party Item with Kodee"

Stop

Show Plan

Edit `ServiceEvents.kt` +13 -5

The confetti animation has been added to the party event item within the composable. However, there are syntax errors and semantic warnings that need to be addressed.

Edit `ServiceEvents.kt` +2

The confetti animation has been integrated into the ServiceEventItem, but syntax errors have been introduced due to a missing closing brace. The next step is to correct the syntax error to ensure proper functionality.

Edit `ServiceEvents.kt` -2

The syntax error caused by excessive closing braces has been corrected by undoing the edit and properly fixing the file structure. The ServiceEventItem structure is now intact and error-free.

Sending LLM request

KotlinConfApp main shared [hotRunJvm]

Project

- kotlinconf-app [KotlinConfApp]
  - .devcontainer
  - .github
  - .gradle
  - .idea
  - .junie
    - guidelines.md
  - .kotlin
  - androidApp
  - backend
  - build
  - docs
  - gradle
  - iosApp
  - kotlin-js-store
  - news
  - shared
  - ui-components
  - ui-components-gallery
  - .editorconfig
  - .gitignore
  - app.json
  - build.gradle.kts
  - cleanup.sh
  - CODE_OF_CONDUCT.md

guidelines.md

```
KotlinConf App - Development Guidelines

1. Build and Configuration

- Repository layout (selected modules)
  - shared: KMP business logic + UI; targets Android, JVM (desktop), iOS (arm64/x64/simulator), JS and WasmJS
  - androidApp: Android application consuming shared
  - ui-components: design system and reusable Compose components consumed by shared
  - backend: Ktor server consuming shared

- Typical build commands
  - Android: use the `androidApp` run configuration
  - iOS: use `KotlinConfAppScheme`
  - Desktop: `./gradlew :shared:jvmRunHot -DmainClass=org.jetbrains.kotlinconf.MainKt`
  - Web: `./gradlew :shared:wasmJsBrowserDevelopmentRun `
  - Backend: `./gradlew :backend:run`

- iOS
  - Frameworks are produced for iosX64/iosArm64/iosSimulatorArm64 with
```

1:40 LF UTF-8

Settings

Tools > Junie > MCP Settings

- Tools
  - Junie
    - Action Allowlist
    - MCP Settings
    - Models
    - Project Settings
  - Jupyter
  - Kotlin Notebook
  - MCP Server
  - Qodana
  - Remote SSH Externa
  - Rsync
  - Shared Indexes
  - Space
  - SSH Configuration
  - SSH Terminal
  - Startup Tasks
  - Tasks

MCP Servers

Add custom MCP configuration here. Configure project specific MCP in project-root/.junie/mcp Learn more

List of servers

	Name	Status	Level
☑	deepl	✓	Global
☑	bluesky-mcp	✓	Global
☑	context7	✓	Global
☑	github	✓	Global

Settings

Tools > Junie > MCP Settings

Tools
- Junie
  - Action Allow
  - MCP Setting
  - Models
  - Project Setti
- Jupyter
- Kotlin Notebo
- MCP Server
- Qodana
- Remote SSH E
- Rsync
- Shared Indexe
- Space
- SSH Configuration
- SSH Terminal
- Startup Tasks
- Tasks

MCP Servers

e/mcp Learn more

Status	Level
	Global
	Global
	Global
	Global

Edit MCP Configuration

```
{
  "mcpServers": {
    "deepl": {
      "command": "npx",
      "args": ["deepl-mcp-server"],
      "env": {
        "DEEPL_API_KEY":
```

Current location is /Users/zsmb/.junie/mcp/mcp.json

Cancel OK

Junie EAP

Japanese Translation for UI Components Strings.

Stop

Show Plan

A Japanese translation has been initiated for the UI component strings, based on the provided English content. The base English strings have been successfully extracted in preparation for translation.

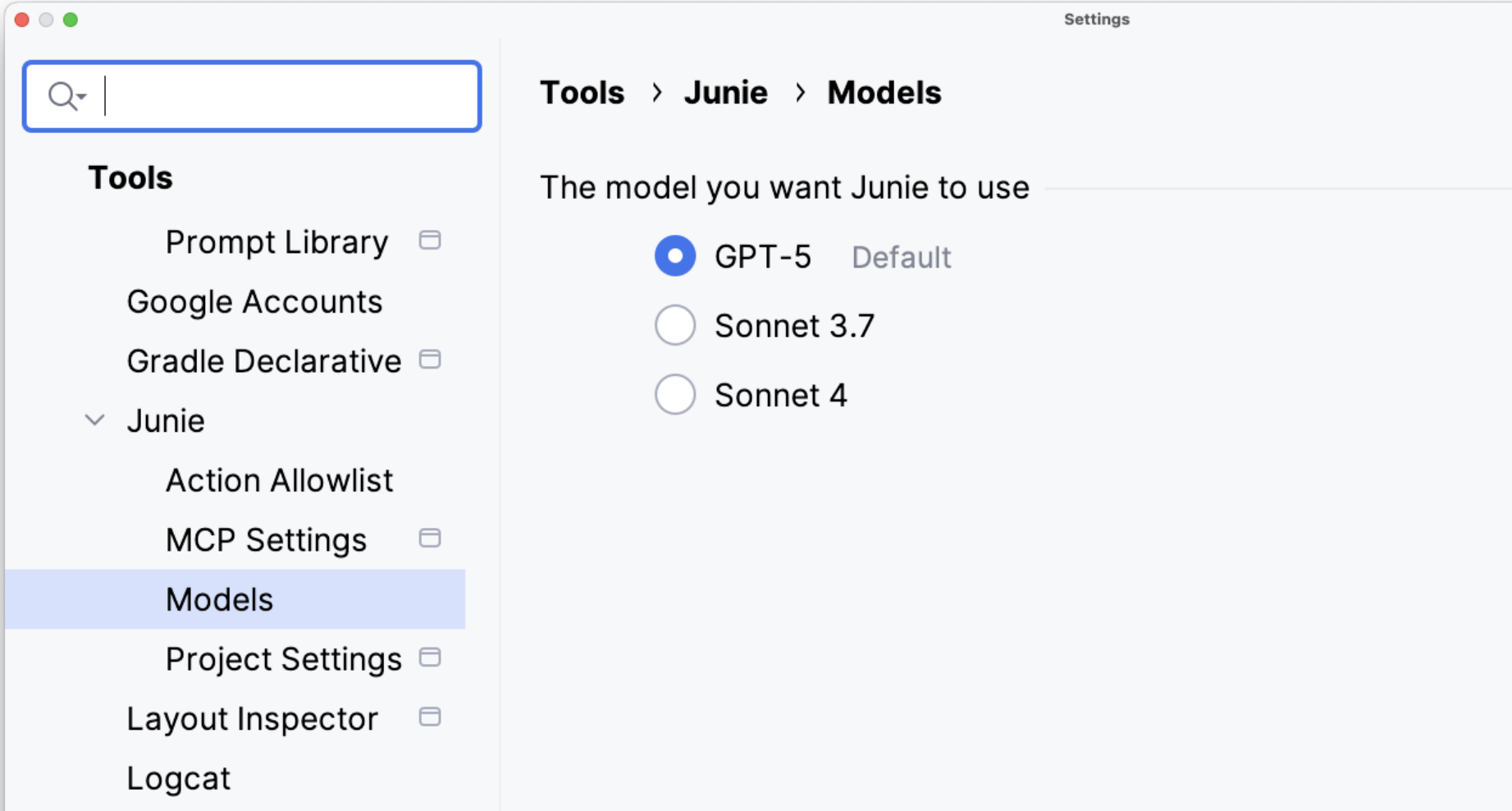
MCP

```
deepl/translate-text
{
   "text": "Translate the following Android
string resources from English to Japanese.
\nRules:\n- Keep keys on the left of '=' exactly
unchanged.\n- Only translate the text on the
right of '='.\n- Preserve placeholders like %1$s
and %1$d exactly.\n- Do not add quotes. Do not
wrap lines. One key=value per line. \n- Use
```

Expand

The English string resources have been successfully translated into Japanese, preserving the original key-value structure and placeholders. The translations maintain a natural and concise UI expression.

Working


Settings
Tools
Prompt Library
Google Accounts
Gradle Declarative
Junie
Action Allowlist
MCP Settings
Models
Project Settings
Layout Inspector
Logcat
Tools › Junie › Models
The model you want Junie to use
GPT-5
Default
Sonnet 3.7
Sonnet 4

Junie
by JETBRAINS

EAP

jb.gg/junie-github

Add a German translation #517

Closed #519

zsmb13 opened 3 weeks ago · edited by zsmb13 Edits Member

Add German translations for the strings of the ui-components and shared modules. These should be placed in `values-de/strings.xml` files.

Create sub-issue

zsmb13 3 weeks ago Member Author

zsmb13 opened 3 weeks ago · edited by zsmb13 Edits Member

Add German translations for the strings of the ui-components and shared modules. These should be placed in `values-de/strings.xml` files.

Create sub-issue

zsmb13 3 weeks ago Member Author

@jetbrains-junie Please work on this issue

👍 1

jetbrains-junie bot 3 weeks ago Contributor

Hey, it's Junie by JetBrains! I started processing your request 🚀

Create sub-issue

zsmb13 3 weeks ago Member Author

@jetbrains-junie Please work on this issue

👍 1

jetbrains-junie bot 3 weeks ago Contributor

Hey, it's Junie by JetBrains! I started processing your request 🚀

jetbrains-junie mentioned this 3 weeks ago

[Junie]: feat(i18n): add German translations for all modules #519

[Junie]: feat(i18n): add German translations for all modules #519

Edit | Code

Merged **zsmb13** merged 5 commits into `main` from `jetbrains-junie/issue-517-run-2cfec4bd-750f-41e1-987c-140a274f07b1` 3 weeks ago

Conversation 13 | Commits 5 | Checks 5 | Files changed 2 | +210 −0

jetbrains-junie bot commented 3 weeks ago — Contributor

📌 Hey! This PR was made for you with Junie, the coding agent by JetBrains Early Access Preview

It's still learning, developing, and might make mistakes. Please make sure you review the changes before you accept them.
We'd love your feedback — join our Discord to share bugs, ideas: here.

- 🔗 **Issue:** Fixes: Add a German translation #517
- ⚙️ **Trigger:** Issue

📊 Junie Summary

German translations have been successfully added for all necessary strings across multiple modules, including ui-components, shared, androidApp, and ui-components-gallery. Each module now contains a dedicated `values-de/strings.xml` file with appropriate translations. No errors or omissions were noted during the implementation.

Reviewers
- ashdavies
- SebastianAigner

Assignees
No one—assign yourself

Labels
junie

Projects
None yet

Milestone
No milestone

Development

Merged

[Junie]: feat(i18n): add German translations for all modules #519

zsmb13 merged 5 commits into `main` from `jetbrains-junie/issue-5…` 3 weeks ago

zsmb13 commented 3 weeks ago — Member

Please remove the translations for app name, those are not required

zsmb13 commented 3 weeks ago — Member

@jetbrains-junie ^^

👍 1

jetbrains-junie bot commented 3 weeks ago — Contributor Author

Hey, it's Junie by JetBrains! I started processing your request 🚀

`[issue-519] refactor: remove app name translations from German files` … `655cd70`

Notifications Customize

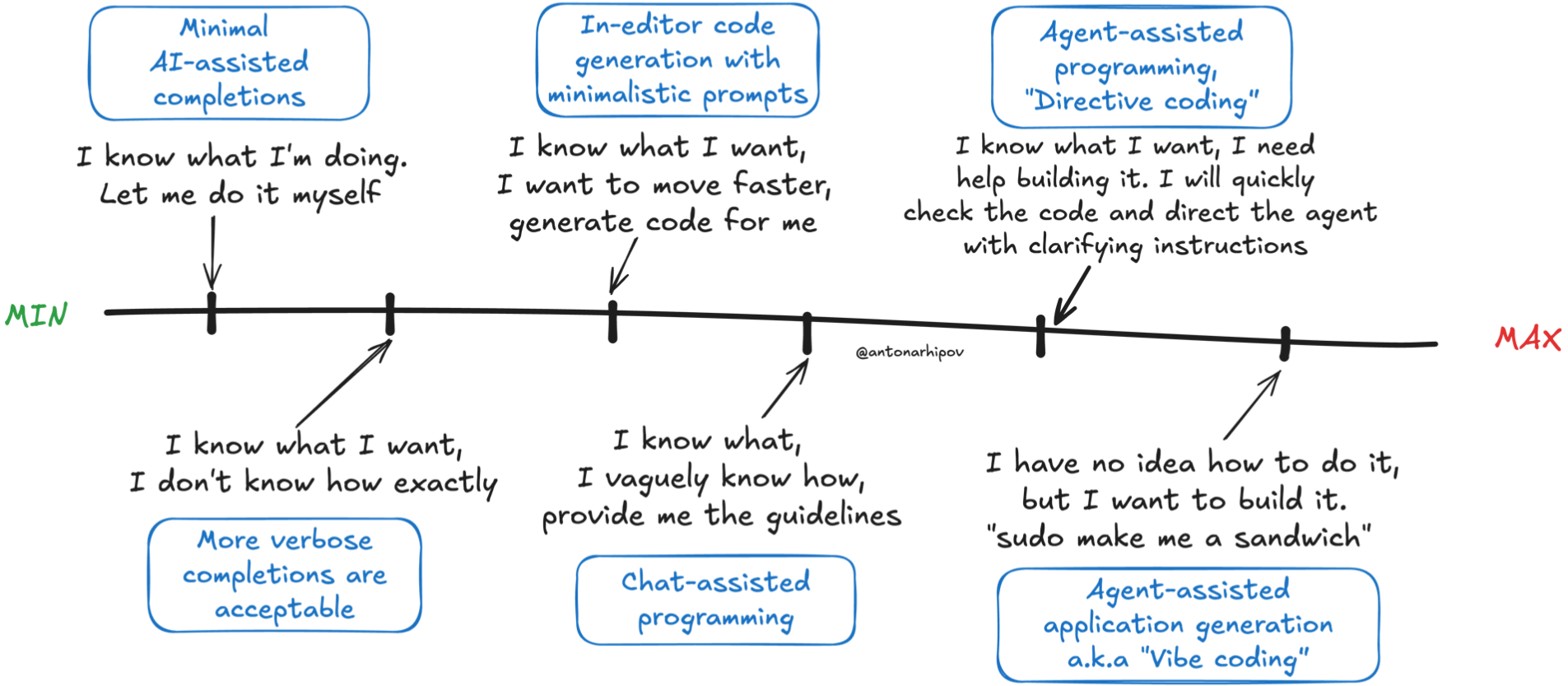
Unsubscribe

You're receiving notifications because you're watching this repository.

3 participants

Lock conversation

Levels of AI-assisted programming



jb.gg/ai

JetBrains AI

Optimize your workflow. With AI built for you.

We build smarter, more efficient tools to make your development workflow faster, more productive, and enjoyable. Choose the AI support that suits you best: from AI-powered code assistance to full automation of coding tasks right within our IDEs. Powered by our proprietary models and best-in-class AI, our solutions are responsibly designed to ensure transparency, protect your privacy, and keep you in control.

AI that fits your development needs

AI in IDEs

Less routine, more coding joy. At JetBrains, we design AI to empower developers, not replace them. From unlimited code completion to Junie, the coding agent – all refined JetBrains AI features are now available right in your IDE, for free.

Get Started. It's free

JETBRAINS
AI
Developer Tools
Team Tools
Education
Solutions
Support
Store
Upcoming Livestream:
Building Smarter AI Agents With Koog
Monday, August 18, 2025 17:00 – 18:00 local time
Register
JetBrains AI
Optimize your workflow. With AI built for you.
We build smarter, more efficient tools to make your development workflow faster, more productive, and enjoyable. Choose the AI support that suits you best: from AI-powered code assistance to full automation of coding tasks right within our IDEs. Powered by our proprietary models and best-in-class AI, our solutions are responsibly designed to ensure transparency, protect your privacy, and keep you in control.

Part 1

AI-assisted coding in Kotlin

Part 2

Building AI-powered apps in Kotlin

Kotlin ♡ AI on every layer

- Model Provider SDKs
- Core AI toolkit
- Agentic Frameworks

Model Provider SDKs

OpenAI Java API Library

maven-central v2.20.1 javadoc 2.20.1

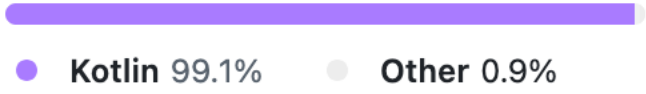
The OpenAI Java SDK provides convenient access to the OpenAI REST API from applications written in Java.

The REST API documentation can be found on platform.openai.com. Javadocs are available on javadoc.io.

Installation

Try `openai-java-spring-boot-starter` if you're using Spring Boot!

Languages



Model Provider SDKs

Languages

Anthropic Java API Library

maven-central v2.3.1 javadoc 2.3.1

The Anthropic Java SDK provides convenient access to the Anthropic REST API from applications written in Java.

The REST API documentation can be found on docs.anthropic.com. Javadocs are available on javadoc.io.

Installation

Gradle

```
implementation("com.anthropic:anthropic-java:2.3.1")
```

ML Kit from Google

ML Kit Guides Reference More Search / English

Machine learning for mobile developers

ML Kit brings Google's machine learning expertise to mobile developers in a powerful and easy-to-use package. Make your iOS and Android apps more engaging, personalized, and helpful with solutions that are optimized to run on device.

Get started

Optimized for mobile

ML Kit's processing happens on-device. This makes it fast and unlocks real-time use cases like processing of camera input. It also works while offline and can be used for processing images and text that need to remain on the device.

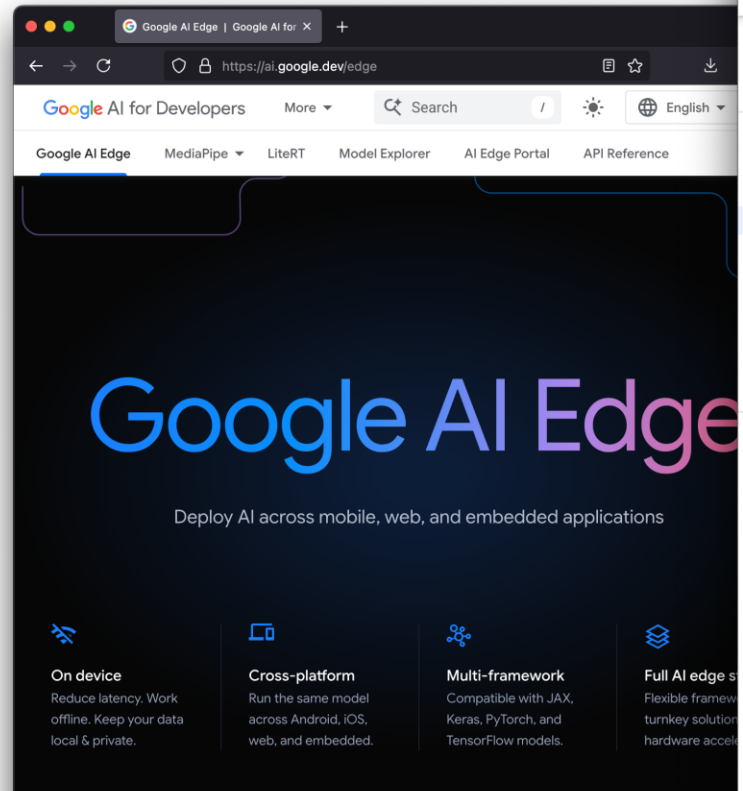
Built with Google expertise

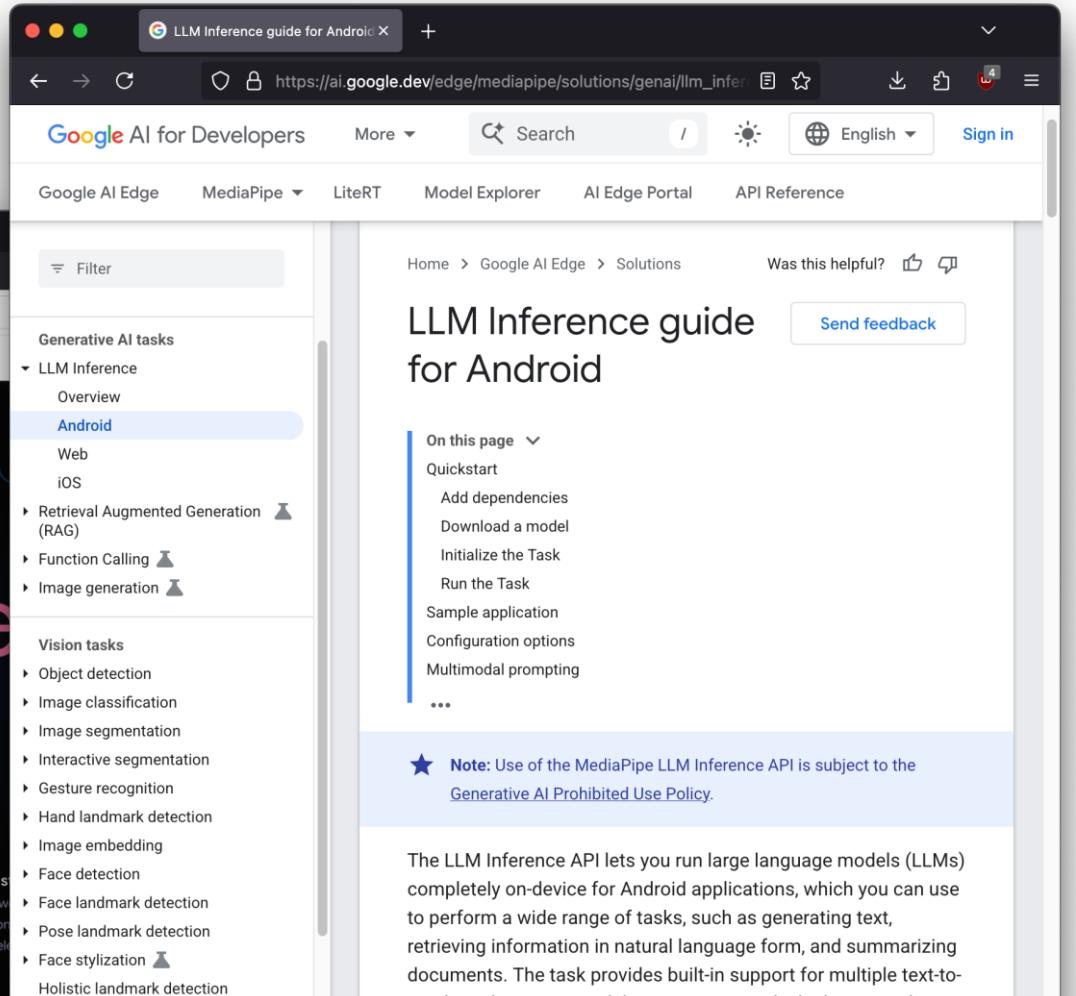
Take advantage of the machine learning technologies that power Google's own experiences on mobile.

Easy to use

We combine best-in-class machine learning models with advanced processing pipelines and offer these through easy-to-use APIs to enable powerful use cases in your apps.

AI Edge & MediaPipe





Introducing **Koog**

Kotlin AI Agent Framework

What is Koog?

- Kotlin framework for **AI agents**
  - LLM-enabled applications that make decisions and can influence other systems
- Manages LLM interactions
- Provides DSLs to express complex workflows via graphs

Platforms supported by Koog

- Android
- JVM (Desktop / Server-Side)
- Web (Wasm / JS)
- iOS

Truly multiplatform!

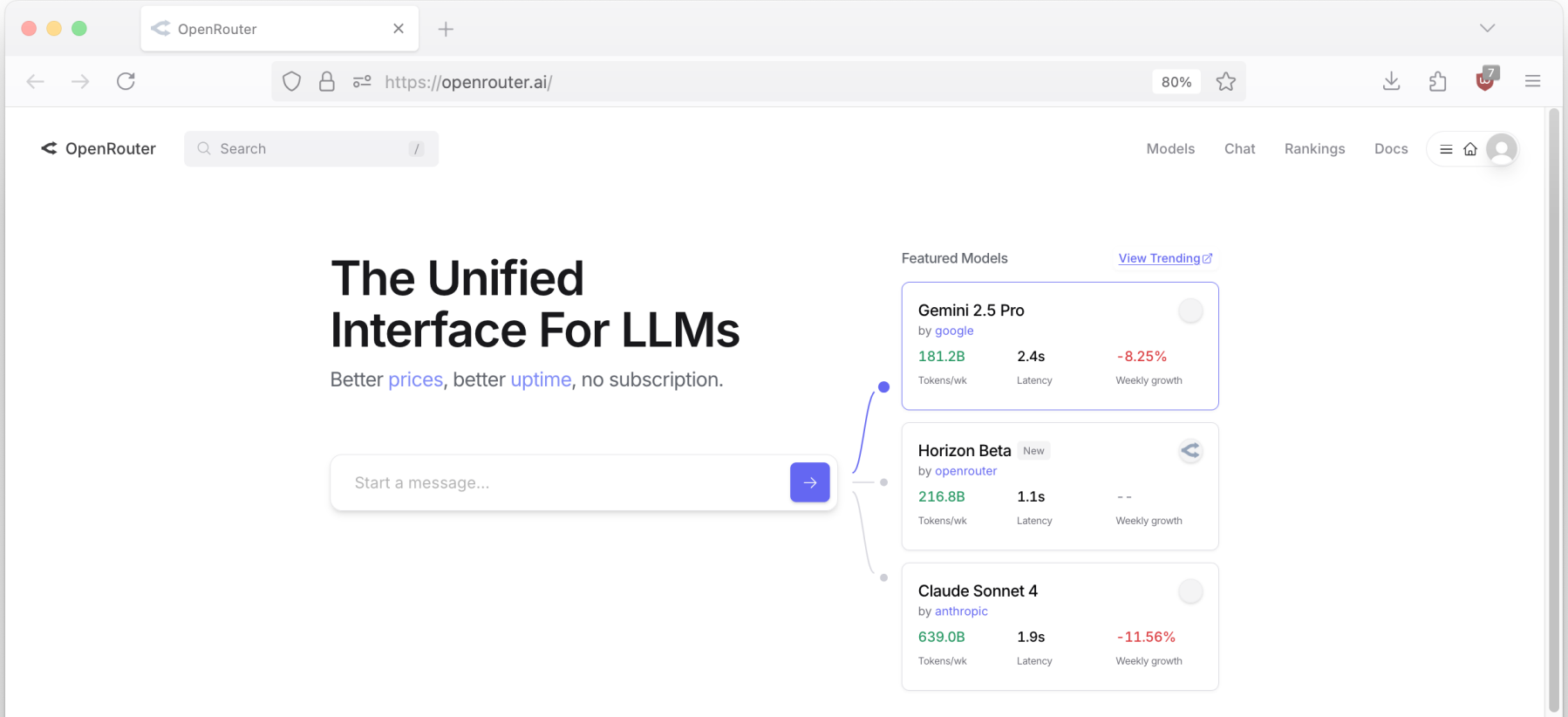
Let's cook with Koog!

Step 0: Get LLM access!

Koog supports major vendors

- Google
- OpenAI
- Anthropic
- OpenRouter
- Ollama (on-device LLMs!)

Personal recommendation: OpenRouter



Personal recommendation: OpenRouter

1 Signup

Create an account to get started. You can set up an org for your team later.

2 Buy credits

Credits can be used with any model or provider.

Apr 1 $99

Mar 30 $10

3 Get your API key

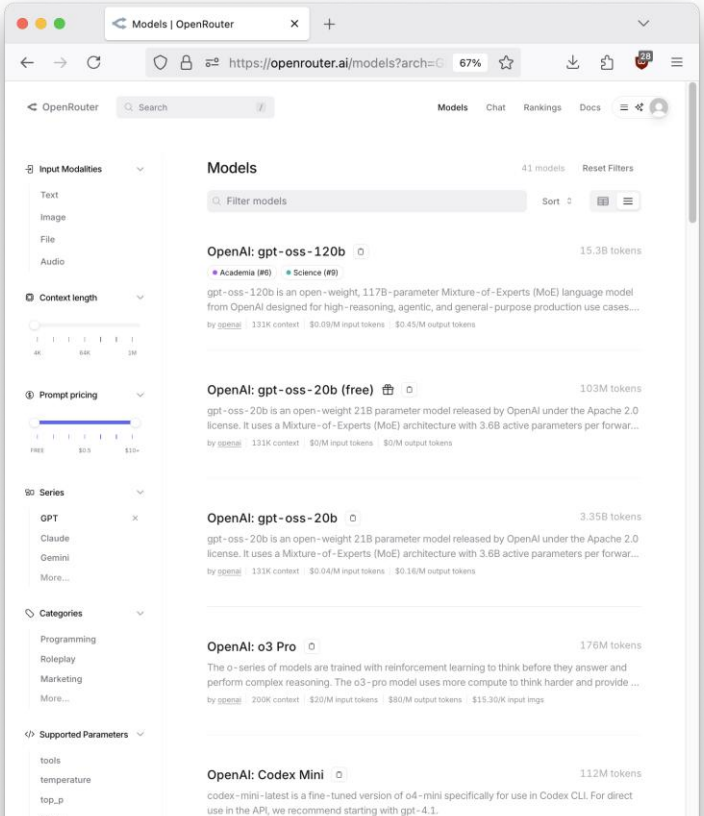
Create an API key and start making requests. Fully OpenAI compatible.

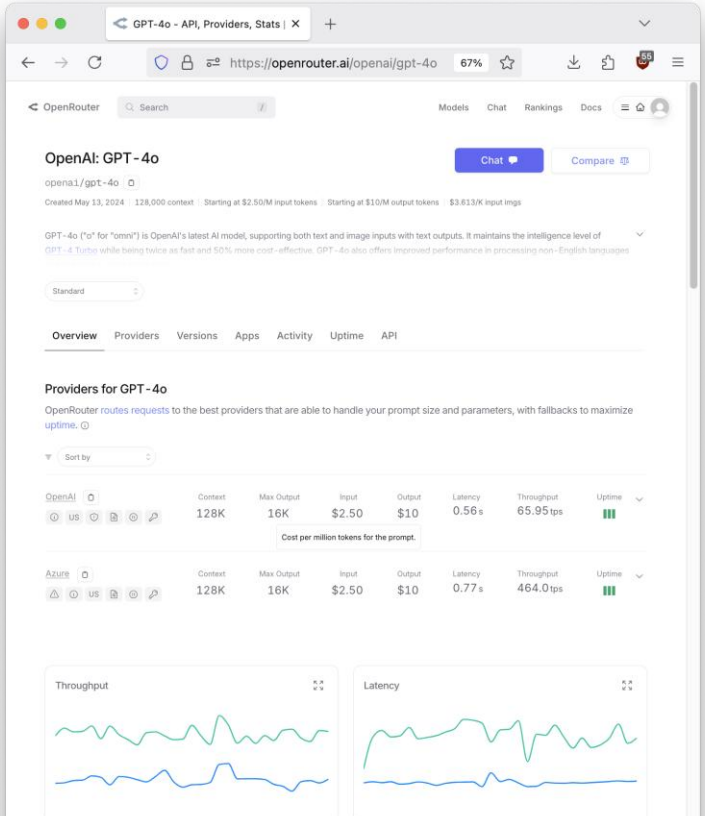
OPENROUTER_API_KEY

••••••••••••••••

prepaid = no scary billing!
(and $5 gets you** a lot **of tokens!)

You can choose and try different models





A very simple agent

```
val agent: Agent<String, String> = AIAgent(
    executor = simpleOpenRouterExecutor(ApiKeys.openRouter),
    systemPrompt = "You are a helpful assistant. Answer concisely.",
    llmModel = OpenAIModels.Chat.GPT5,
)

suspend fun main() {
    val result: String =
        agent.run("What is the capital of Bavaria?")
    println(result)
}
```

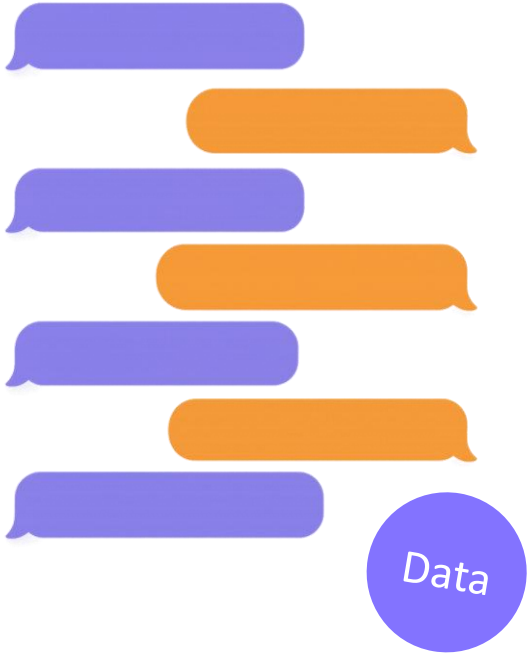
The capital of Bavaria is Munich.

We need more… much more!

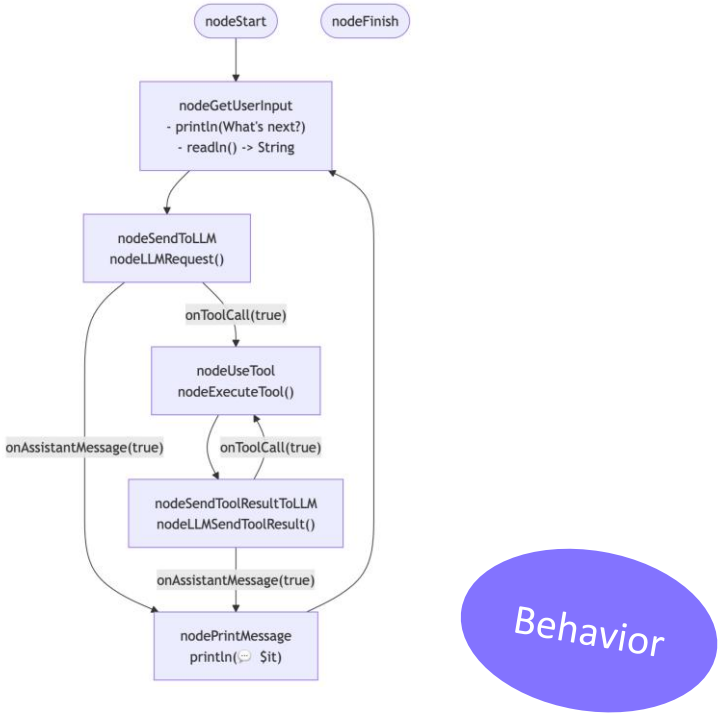
- Exchanging more than one message 💬
- A way to *steer* the agent 🕹️
- Working with tools 🛠️
- A way to observe the agent 🔍

Koog gives you everything to build **production-ready agents.**

Koog's fundamental structures



LLM message history

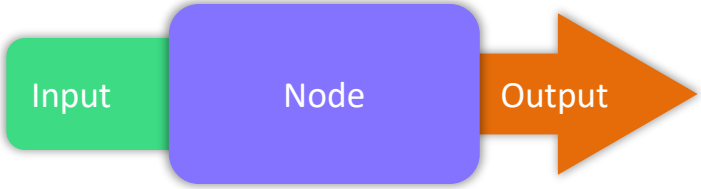


Graph(s): Made from nodes and edges

LLM message history

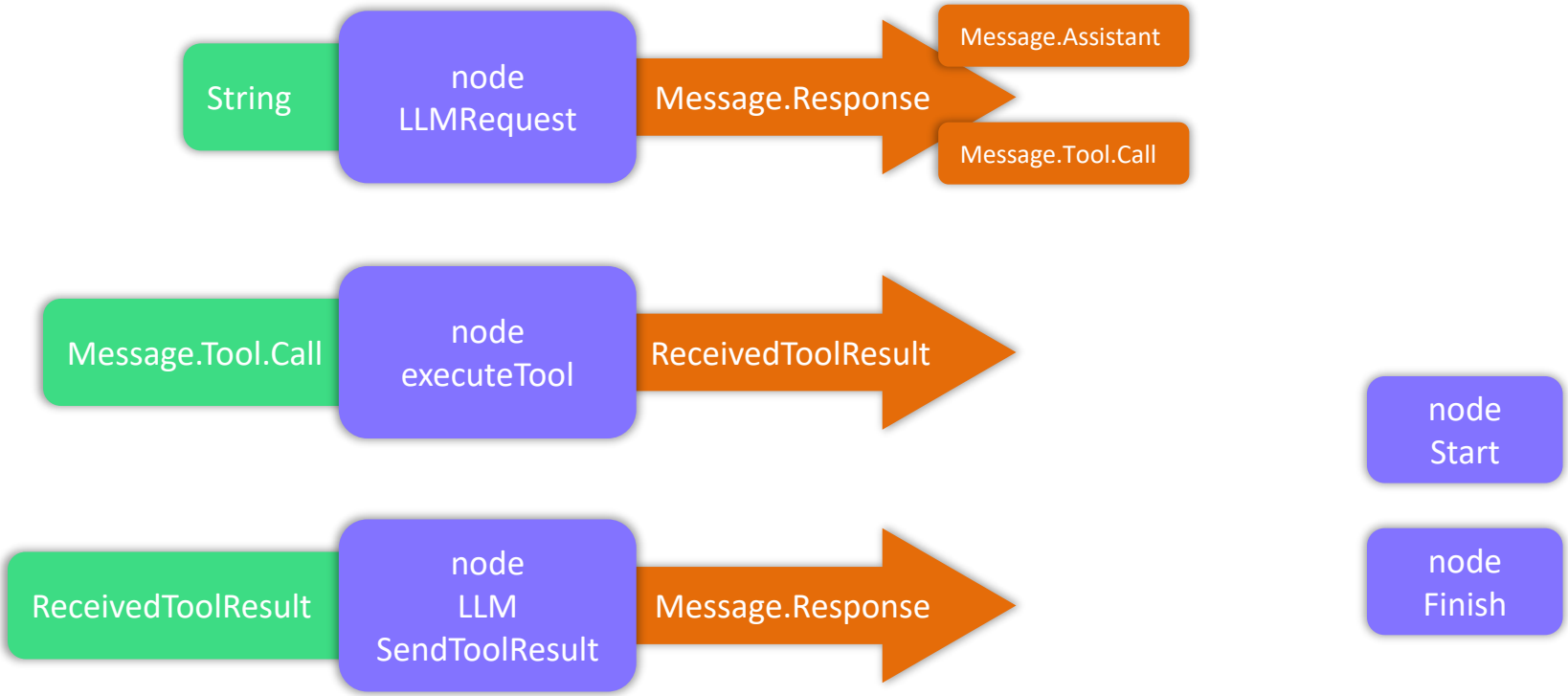
- Automatically managed by Koog
- Doesn't have to be manually changed in normal operation
  - Ability to make changes is still exposed
- Advanced features for management (like history compression)

Introducing graphs and nodes

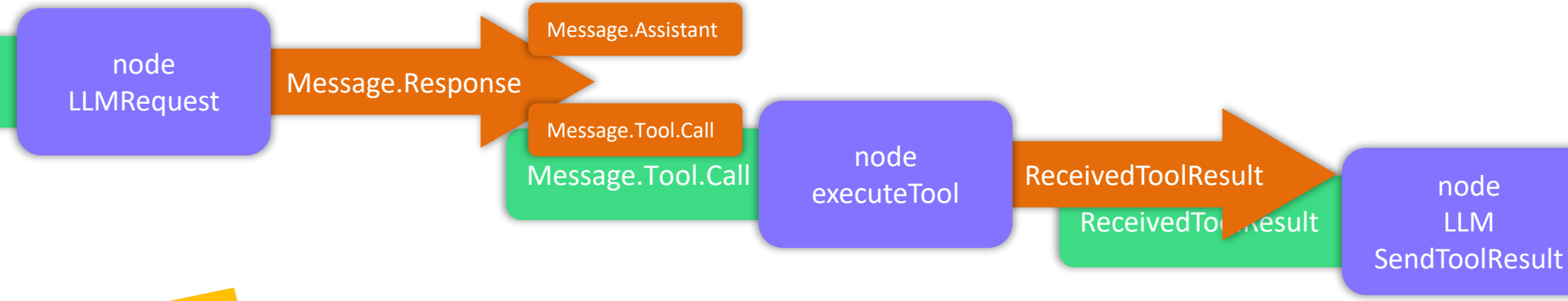


Some available nodes

node inputs/outputs
≠ LLM history!

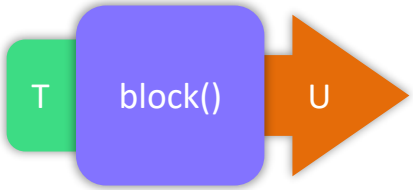


Nodes snap together with edges



we can parametrize edges with conditionals, too!

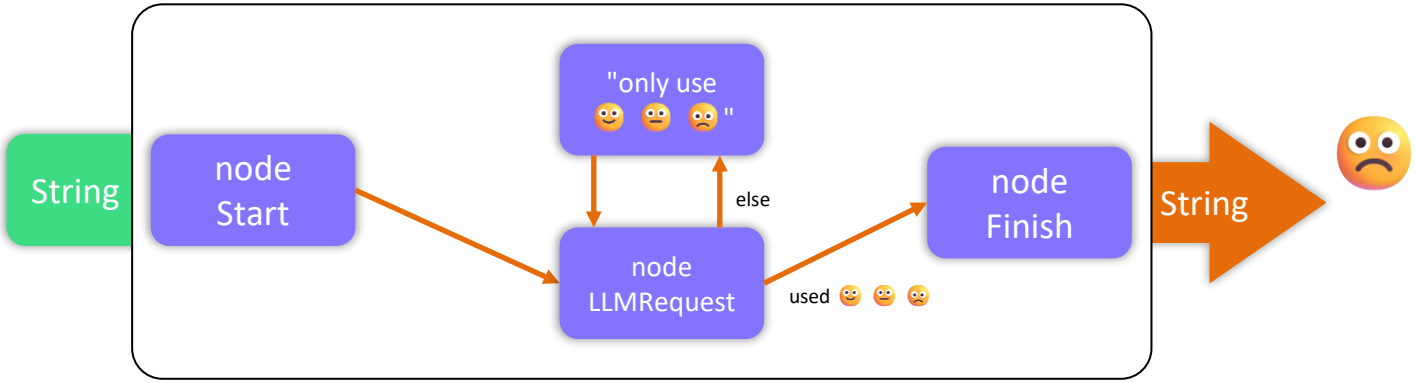
Custom node




```
val nodeCustom by node<String, Int> { str ->
    println(str)
    42
}
```

Creating graphs with nodes and edges

"I'm *blue*. What emotion am I feeling?"




```
private val classificationStrategy = strategy<String, String>("Emotion classification") {
  val nodeSendToLLM by nodeLLMRequest()
  val nodeSayEmojiOnlyPlease by node<Any, String> {
    "Please only react using the following emojis: 🙂 😐 ☹️"
  }
  edge(nodeStart forwardTo nodeSendToLLM)
  edge(nodeSendToLLM forwardTo nodeFinish onAssistantMessage { msg ->
    listOf("🙂", "😐", "☹️").any { emoji -> emoji in msg.content }
  })
  edge(nodeSendToLLM forwardTo nodeSayEmojiOnlyPlease)
  edge(nodeSayEmojiOnlyPlease forwardTo nodeSendToLLM)
}
```

...didn't touch the message history at all!

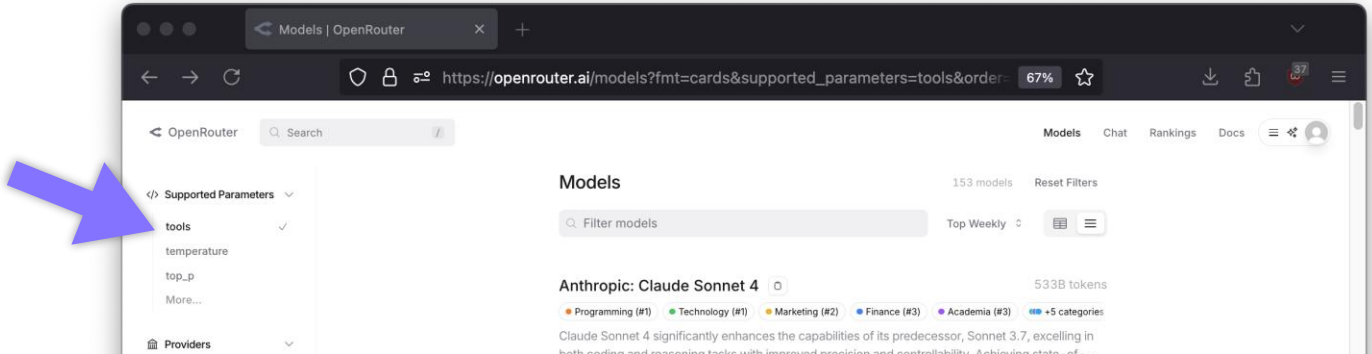
make them as complex (and nested) as you need!

Plugging it in

```
AIAgent(
    executor = simpleOpenRouterExecutor(ApiKeys.openRouter),
    systemPrompt = "Answer the user's question.",
    llmModel = OpenAIModels.Chat.GPT5,
    strategy = classificationStrategy,
)
```

Integrating with tools

- We want our agent to affect the world around it!
- We need to provide it with (deterministic) integrations for the agent to:
  - Obtain task-specific information (e.g. "Read this GitHub issue")
  - Perform task-specific actions (e.g. "Transfer 💰 to ")
- LLMs are specifically trained for this!



Declaring our own tool (almost)

```
suspend fun getGitHubIssueContents(
    owner: String,
    repo: String,
    issueNumber: Int,
    token: String? = null,
): String =
    GitHubIssueClient.getGitHubIssueContents(
        owner,
        repo,
        issueNumber,
        token
    )
```

Creating a toolset with LLM descriptions

```
@LLMDescription("Tools for interacting with GitHub Issues API. Can fetch JSON of a GitHub issue.")
class GitHubIssueToolset : ToolSet {
  @Tool("get_github_issue_contents")
  @LLMDescription("Fetch the JSON of a GitHub issue and return it as a raw string.")
  suspend fun getGitHubIssueContents(
    @LLMDescription("Repository owner or organization, e.g., 'octocat'.")
    owner: String,
    @LLMDescription("Repository name, e.g., 'hello-world'.")
    repo: String,
    @LLMDescription("Issue number in the repository.")
    issueNumber: Int,
    @LLMDescription("Optional GitHub token. If omitted, GITHUB_TOKEN env variable will be used.")
    token: String? = null,
  ): String =
    GitHubIssueClient.getGitHubIssueContents(
      owner,
      repo,
      issueNumber,
      token
    )
}
```

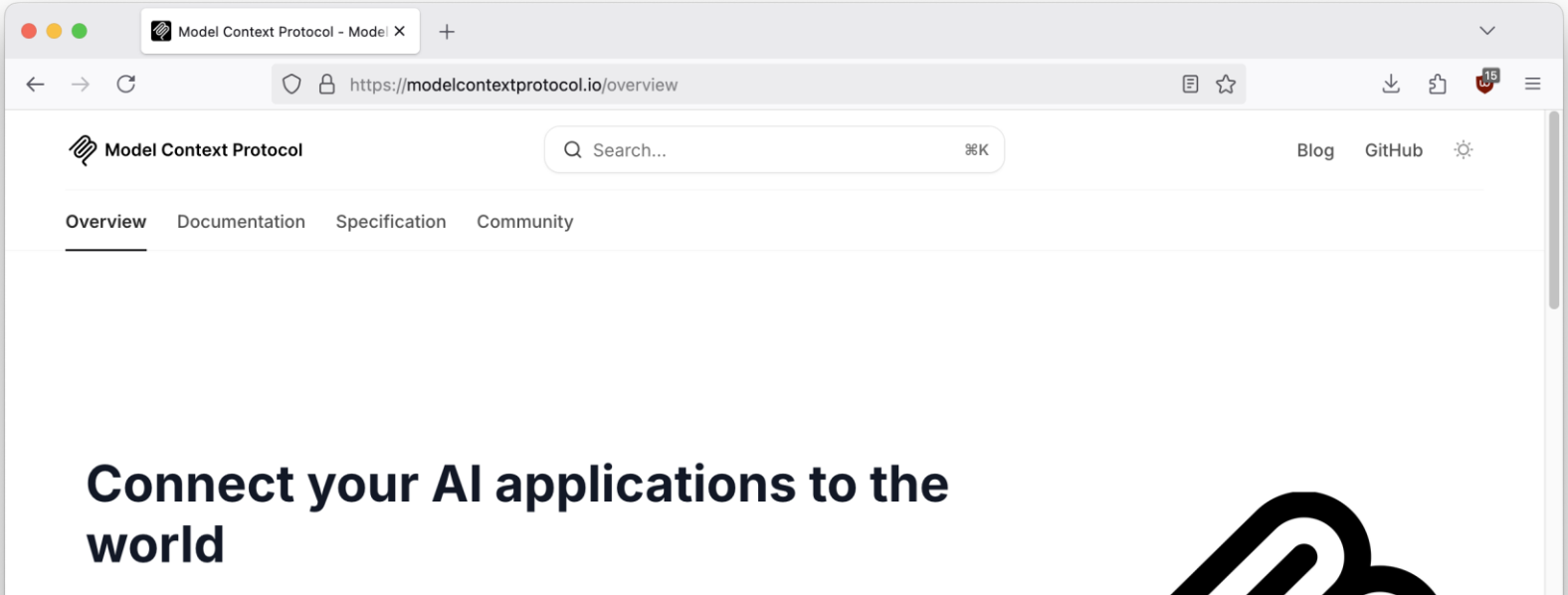
same code, just more annotations!

Plugging it in

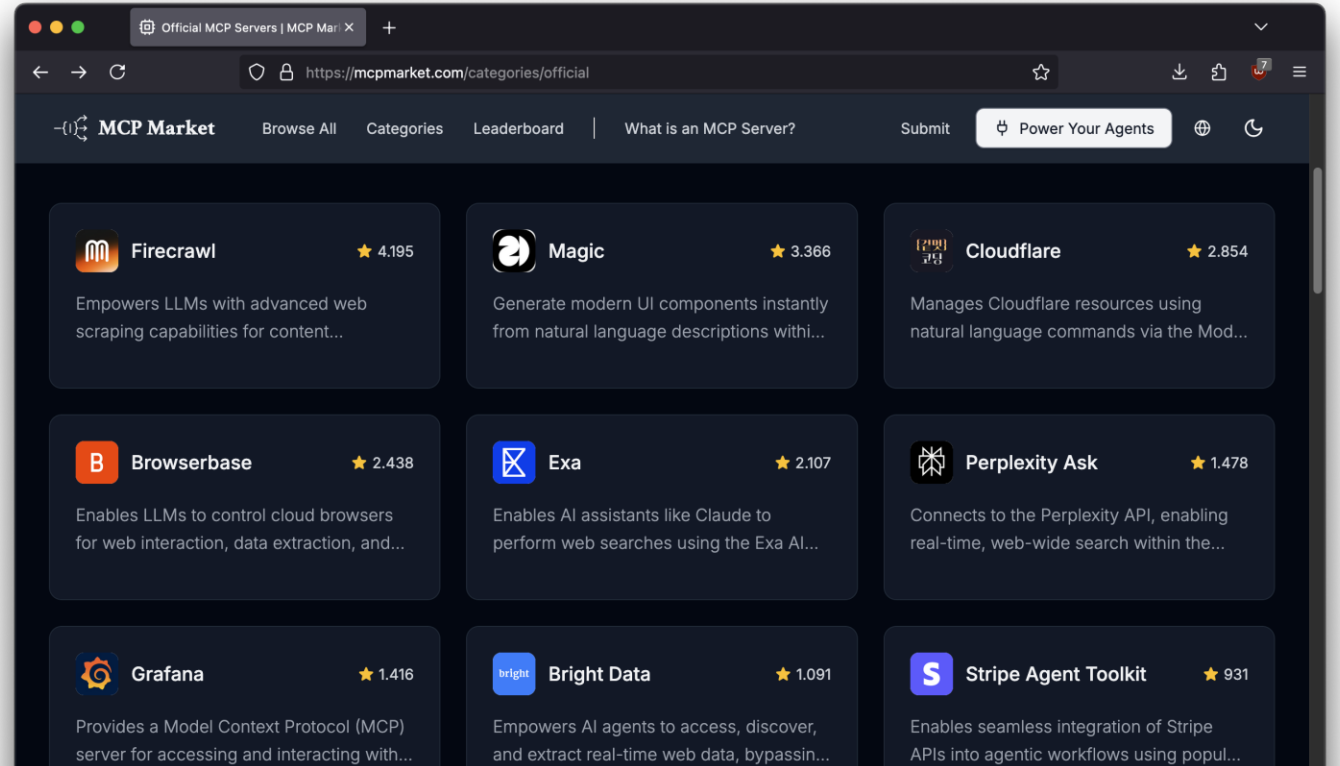
```
AIAgent(
    executor = simpleOpenRouterExecutor(ApiKeys.openRouter),
    systemPrompt = """Answer the user's question
        |about a specific GitHub issue using tools.""".trimMargin(),
    llmModel = OpenAIModels.Chat.GPT5,
    toolRegistry = ToolRegistry {
        tools(GitHubIssueToolset())
    },
    strategy = oneShotStrategy,
)
```

External tools in Koog via MCP

- AI ⟷ Tool communication protocol
- stdio or server-sent events



MCP server inspiration



Adding MCP

Pick one

```
val process = ProcessBuilder("path/to/mcp/server").start()
val stdioTransport =
    McpToolRegistryProvider.defaultStdioTransport(process)


val sseTransport =
    McpToolRegistryProvider.defaultSseTransport(
        "http://localhost:8931"
    )


val toolRegistry = McpToolRegistryProvider.fromTransport(
    transport = yourTransport,
    name = "my-client",
    version = "1.0.0"
)
```

Koog also covers advanced use cases!

Agents as tools

```
private val myAgent : AIAgent<String, String> = AIAgent(   1 Usage
```

takes String

returns String

```
val tool = myAgent.asTool(/*...*/)
```

Not just String → String!

```
val typedStrategy = strategy<String, Int>("Calculation") {
    val nodeToNumber by node<String, Int> { it.split(" ").sumOf { it.toInt() } }
    edge(nodeStart forwardTo nodeToNumber)
    edge(nodeToNumber forwardTo nodeFinish)
}

val typedAgent = AIAgent(
    inputType = typeOf<String>(),
    outputType = typeOf<Int>(),
    promptExecutor = simpleOpenRouterExecutor(ApiKeys.openRouter),
    strategy = typedStrategy,
    agentConfig = AIAgentConfig.withSystemPrompt(prompt = "You are..."),
)

val result: Int = typedAgent.run("5 7 10")
```

...this agent doesn't even use AI 🤔

Structured data processing

```
@Serializable
data class WeatherForecast(
    val temperature: Int,
    val conditions: String,
    val precipitation: Int
)
```

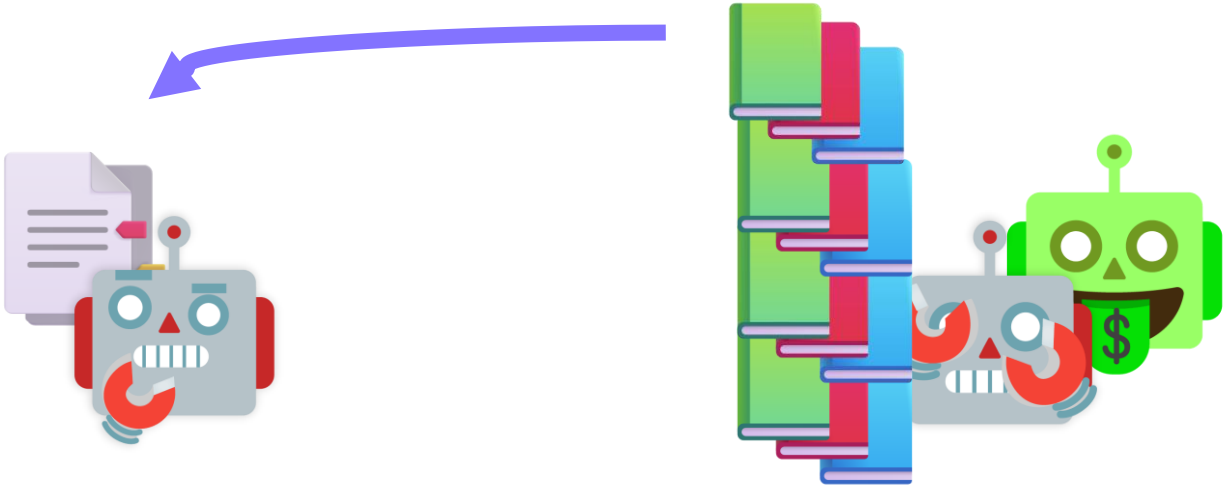
Structured data processing

```
@Serializable
@SerialName("WeatherForecast")
@LLMDescription("Weather forecast for a given location")
data class WeatherForecast(
    @property:LLMDescription("Temperature in Celsius")
    val temperature: Int,
    @property:LLMDescription("Weather conditions (e.g., sunny, cloudy, rainy)")
    val conditions: String,
    @property:LLMDescription("Chance of precipitation in percentage")
    val precipitation: Int
)
```

node
LLMRequest**Structured**

History compression

- LLM APIs are $/token
- LLMs have limited context windows

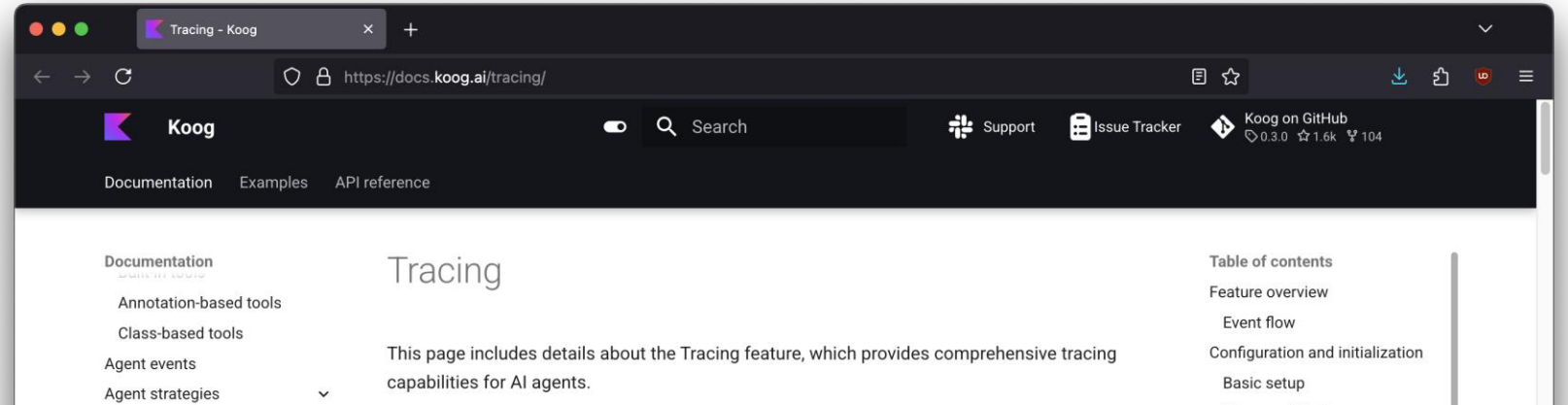


node
LLM
CompressHistory

- Different strategies to remove unnecessary information from LLM message history
  - WholeHistory
  - FromLastNMessages
  - Chunked

Observability / Tracing

- Insights into what's happening inside your agent via the tracing module
- Integration with OpenTelemetry



Supporting advanced use cases

- Memory: Concepts & Facts management
- Content moderation
- Ranked Document Storage for RAG
- Semantic meaning via Embeddings
- Testing & Mocking

Koog runs on phones and servers!

- You can write and run Koog agents to run on Android phones!
- Koog doesn't do inference by itself – it uses APIs for that
- Agents themselves aren't that computationally heavy!

koog.ai

youtube.com/kotlin

Concluding highlights

- AI Tooling helps you work with Kotlin!
  - IDE features like completion and next edit suggestions help write code
  - AI Assistant provides project insights and edits
  - Junie, the JetBrains AI agent, can write full features end-to-end
- Koog makes building AI agents idiomatic
  - It's superpower is composability via graphs!
  - It type-safely integrates into 3rd party systems and your own systems, via tools, MCP & co.
  - It's designed to scale, with features like history compression, observability integrations, and tracing.
  - Koog becomes more powerful day by day!

JETBRAINS Come visit the JetBrains booth!

Enjoy DroidKaigi!

Márton Braun
@zsmb13

Sebastian Aigner
@sebi_io