7. 进入 harness

English | 中文

本章会向 harne`tools` 服务注册一个可由模型调用的工具，通过 harness 工具流水线执行它，并观察结果事件。整个示例无需密钥，也不会调用模型。

工具插件

[`greet-tool.ts` [`tmp/cordis-tutorial` 中:

```
import type { Context } from '@deepseek-ai/cordis'
import { brandString } from '@deepseek-ai/dsh-brand'
import { defineTool } from '@deepseek-ai/dsh-tools'
import type { ToolCallId } from '@deepseek-ai/dsh-llm'

export const name = 'greet-tool'
export const inject = ['tools']

export function apply(ctx: Context) {
  ctx.tools.register(defineTool({
    name: 'greet',
    description: 'Greet the named person.',
    parameters: {
      name: { type: 'string', required: true, description: 'Who to greet' },
    },
    output: {
      schema: { type: 'string' },
      render: (_args, value) => [{ type: 'text', text: value }],
    },
    async execute(args) {
      return `Hello, ${args.name}!`
    },
  }))

  // Drive one call through the real execution pipeline, standing in for
```

```
  // the model. ToolCallId brands the correlation id a provider would issue.
  void (async () => {
    const result = await ctx.tools.execute({
      callId: brandString<ToolCallId>('demo-1'),
      name: 'greet',
      arguments: { name: 'Cordis' },
      signal: new AbortController().signal,
    })
    console.log('tool replied:', JSON.stringify(result.content))
  })()
}
```

`inject: ['tools']` [illegible] 3 `ctx.tools.register(...)` [illegible] dispose [illegible] 2 `defineTool` [illegible] `parameters` [illegible] JSO `args` h[illegible] `execute` [illegible] `output.schema` `output.render` [illegible] Native renderer[illegible]

[illegible]

[illegible] `tool-logger.ts` [illegible] `tools/result` [illegible]

```
import type { Context } from '@deepseek-ai/cordis'
import type {} from '@deepseek-ai/dsh-tools'

export const name = 'tool-logger'
export const inject = ['tools']

export function apply(ctx: Context) {
  ctx.on('tools/result', (exec, result) => {
    const text = result.content
      .map(block => (block.type === 'text' ? block.text : ''))
      .join('')
    console.log(`[tool-logger] ${exec.name} -> ${text}`)
  })
}
```

`import type {} from '@deepseek-ai/dsh-tools'` [illegible] `'tools/result'` [illegible] payload [illegible] `stats.ts` 4[illegible]

[illegible]

```
- name: '@deepseek-ai/dsh-system-prompt'
- name: '@deepseek-ai/dsh-tools'
- name: './tool-logger.ts'
- name: './greet-tool.ts'
```

`@deepseek-ai/dsh-tools` [`systemPrompt` 服务，因为工具需要[illegible] schema，所以组合中也要列出该服务的提供方。缺少提供方时，工具插件会像

```
node --import tsx ../../vendor/cordis/bin.js
```

```
[tool-logger] greet -> Hello, Cordis!
tool replied: [{"type":"text","text":"Hello, Cordis!"}]
```

logger `tools/result` [`execute` [illegible]回的 promise 兑现之前。两个插件都不知道另一个插件存在，它们由注册表服务和事件连接。

从这里走向完整 agent（智能体）

真实 agent 就是这套组合再加上更多插件：[illegible] profile [illegible] agent [illegible] `--patch` [illegible] `greet-tool.ts` 即可。

后续可以阅读：

- [`defineTool`，包括呈现和更丰富的 schema。
- 三层 harness 如何组织可替换能力。
- [`cordis-surface` 区块：可以注入和监听的所有内容，各在其所属页面上。
- 架构：这些插件所处的系统地图。

Image not found or type unknown

版本号 #1
由一说春秋 创建于 3 九月 2026 17:00:06
由一说春秋 更新于 3 九月 2026 17:02:14