

基于 AI 的下一代 UI 测试工具

Midscene.js
重新定义测试体验

周晓

@FEDAY 2024.12



自我介绍和项目背景

Bytedance Web infra 团队成员

Module Federation 2.0 作者

Midscene.js 核心开发者

Github 地址: <https://github.com/zhoushaw>



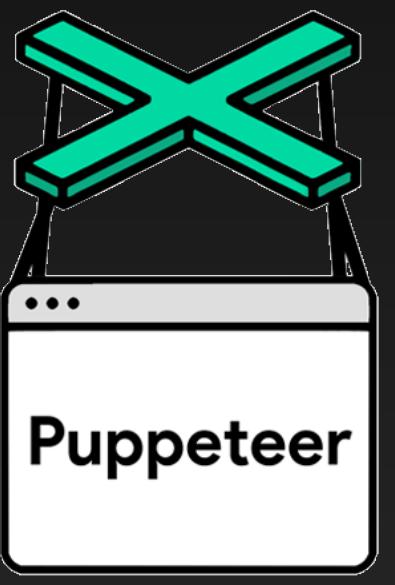
01 传统 UI 自动化测试的局限性

02 Midscene.js 基于多模态 AI 的测试框架

03 AI 在前端团队落地的实践经验

01 传统 UI 自动化测试的局限性

01 传统 UI 自动化测试的局限性



01 传统 UI 自动化测试的局限性-测试脚本编写

```
● ● ●  
it('我参与的 Tab 验证', async ({ page }) => {  
  // 定位并点击“我参与的”Tab  
  const favoriteTab = page.element({  
    attr: 'class',  
    attrValue: /^title/,  
    text: '我参与的',  
  })  
  
  await favoriteTab.click() // 点击“我参与的”Tab  
  
  // 假设 页面返回的客户别名数据  
  const expectedCustomerNames = ['客户A', '客户B', '客户C']  
  
  // 从页面获取客户名  
  const items = page.elements(  
    '[aria-labelledby="m-tabs-1-3"] .infinite-scroll-component div[class^="title"]'  
  )  
  const actualCustomerNames = await Promise.all(  
    items.map(async (item) => await item.getText())  
  )  
  
  // 验证页面展示的客户名与假设的结果一致  
  expect(actualCustomerNames).deep.equal(expectedCustomerNames)  
  
  console.log('客户名验证成功! ')  
})
```

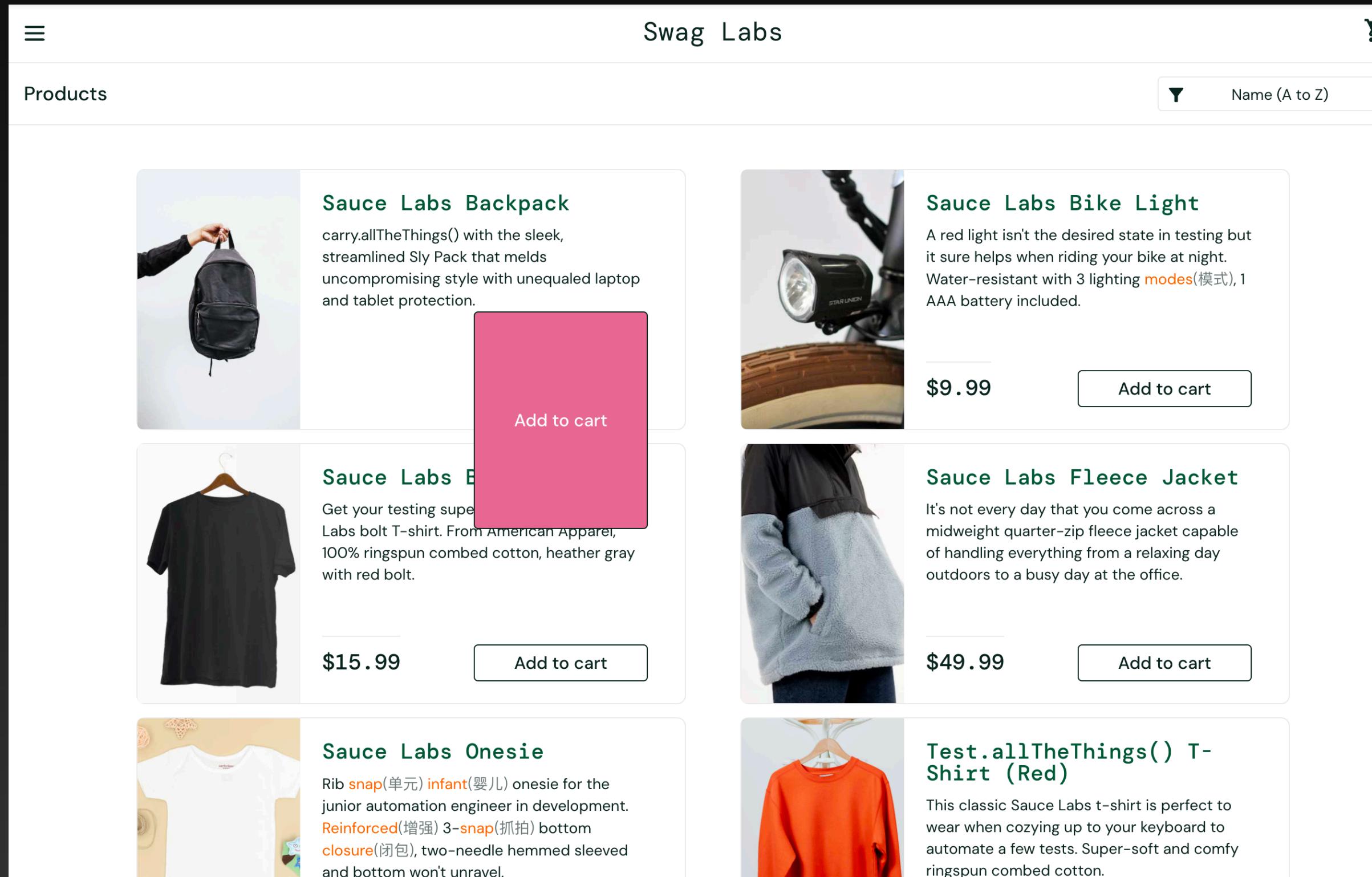
测试脚本编写与维护难点

1. 学习特定框架事件语法
难以对抗页面变化
2. Selector 选择器问题
需要侵入业务代码
3. 页面信息提取
4. 测试脚本可读性/可维护性差
5. 报告可读性差



Midscene.js

01 传统 UI 自动化测试的局限性-视觉检查



视觉检查

1. 白屏检测困难

监测 dom 渲染变化不稳定

2. 样式错乱

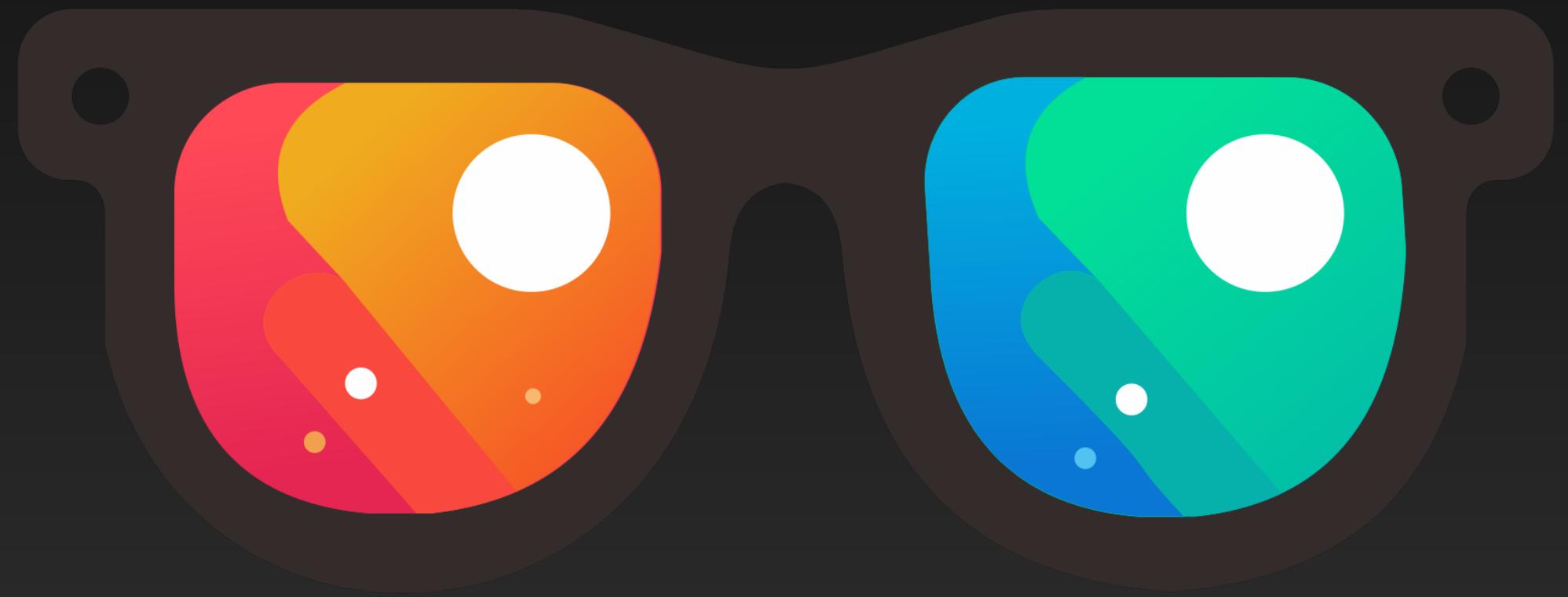
依赖人工视觉检查

02 Midscene.js 基于多模态 AI 的测试框架

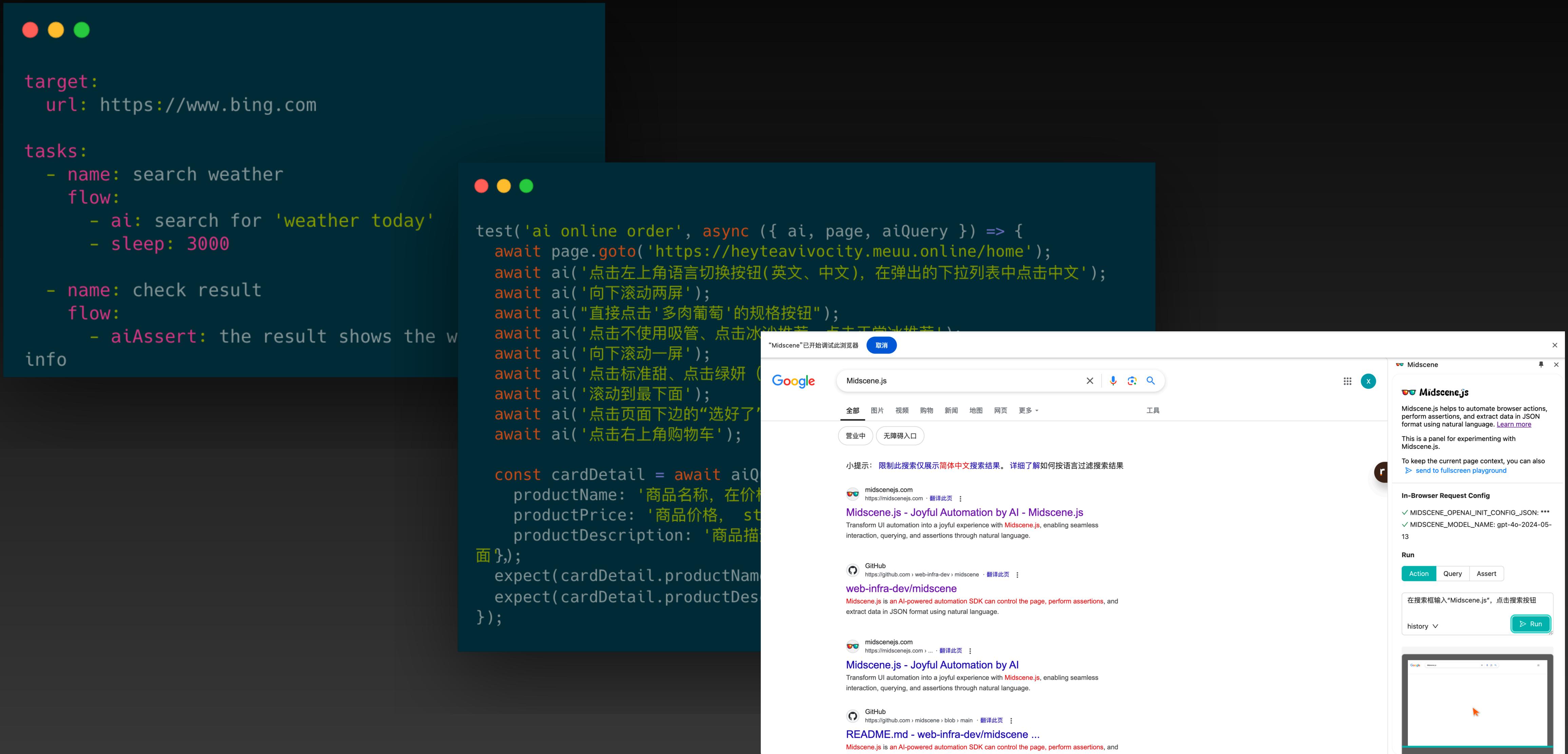
02 Midscene.js 基于多模态 AI 的测试框架

Midscene.js 是一款由多模态 AI
驱动 UI 的自动化工具。

1. 让 UI 自动化测试脚本变得更
容易编写、更易维护
2. 它将对传统 E2E 工具和人工
QA 带来变革！



02 Midscene.js 基于多模态 AI 的测试框架



```
target:
  url: https://www.bing.com

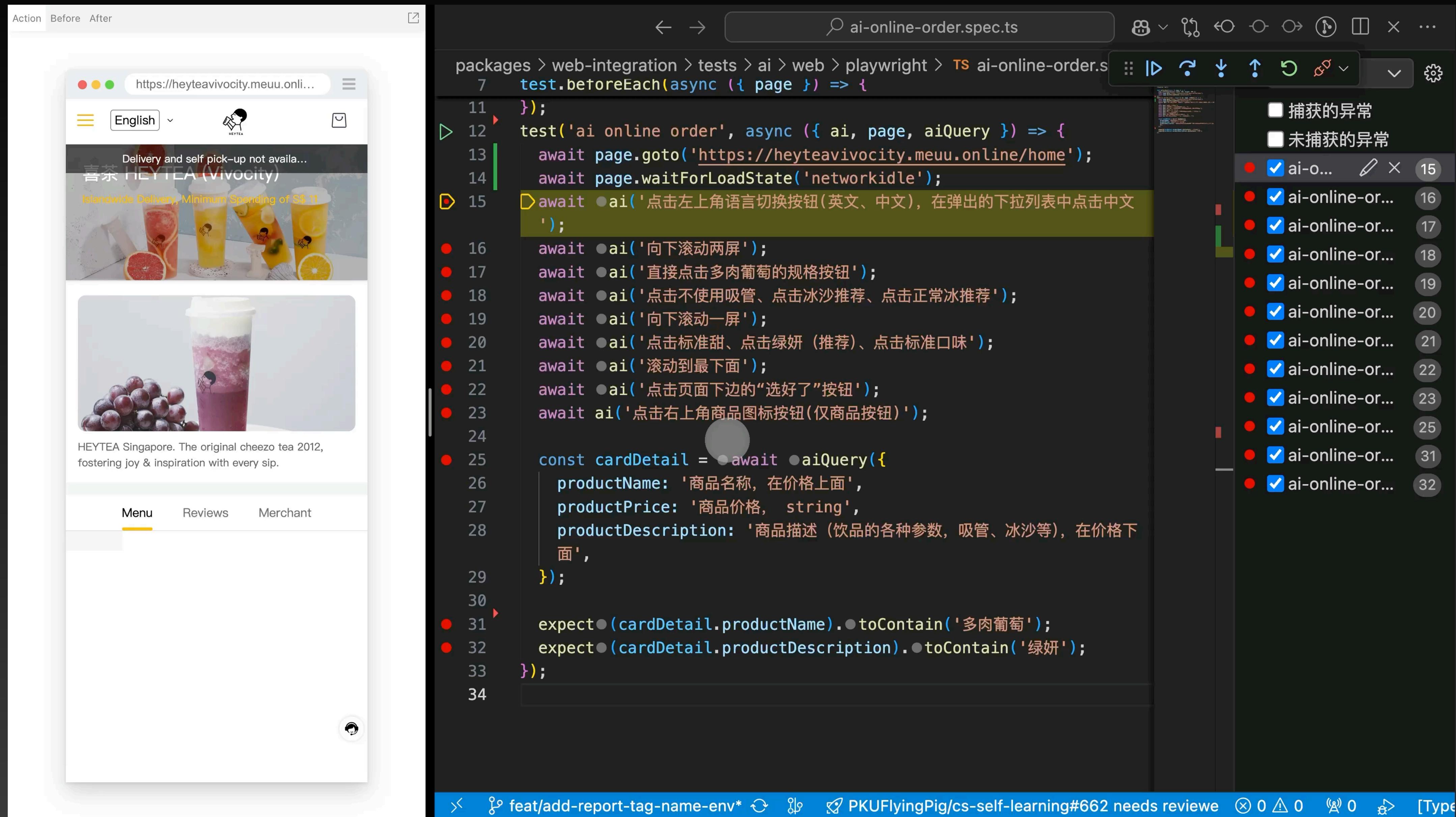
tasks:
- name: search weather
  flow:
    - ai: search for 'weather today'
    - sleep: 3000

- name: check result
  flow:
    - aiAssert: the result shows the w
info
```

```
test('ai online order', async ({ ai, page, aiQuery }) => {
  await page.goto('https://heyteavivocity.meuu.online/home');
  await ai('点击左上角语言切换按钮(英文、中文), 在弹出的下拉列表中点击中文');
  await ai('向下滚动两屏');
  await ai("直接点击'多肉葡萄'的规格按钮");
  await ai('点击不使用吸管、点击冰沙推荐');
  await ai('向下滚动一屏');
  await ai('点击标准甜、点击绿妍');
  await ai('滚动到底部');
  await ai('点击页面下边的“选好了”');
  await ai('点击右上角购物车');

  const cardDetail = await aiQuery('商品名称, 在价格下方');
  expect(cardDetail.productName).toBe('商品名称, 在价格下方');
  expect(cardDetail.productPrice).toBe('商品价格, 在下方');
  expect(cardDetail.productDescription).toBe('商品描述, 在下方');
});
```

02 Midscene.js 基于多模态 AI 的测试框架



The image shows a dual-pane interface. On the left is a browser window displaying the HeyTea Vivocity website. The page features a top banner with a delivery note, a large image of a drink, and a caption about HEYTEA Singapore. Below is a menu bar with 'Menu' (highlighted in yellow), 'Reviews', and 'Merchant'. On the right is a code editor showing a TypeScript test file named 'ai-online-order.spec.ts'. The code uses Playwright's multi-modal AI feature to interact with the HeyTea site. A specific test step at line 15 is highlighted with a green background and a tooltip: 'await ai('点击左上角语言切换按钮(英文、中文)，在弹出的下拉列表中点击中文')';. The code editor also includes a sidebar for managing test cases and a status bar at the bottom.

```
packages > web-integration > tests > ai > web > playwright > TS ai-online-order.s
7  test.beforeEach(async ({ page }) => {
11    });
12  test('ai online order', async ({ ai, page, aiQuery }) => {
13    await page.goto('https://heyteavivocity.meuu.online/home');
14    await page.waitForLoadState('networkidle');
15  await ai('点击左上角语言切换按钮(英文、中文)，在弹出的下拉列表中点击中文');
16    await ai('向下滚动两屏');
17    await ai('直接点击多肉葡萄的规格按钮');
18    await ai('点击不使用吸管、点击冰沙推荐、点击正常冰推荐');
19    await ai('向下滚动一屏');
20    await ai('点击标准甜、点击绿妍（推荐）、点击标准口味');
21    await ai('滚动到最下面');
22    await ai('点击页面下边的“选好了”按钮');
23    await ai('点击右上角商品图标按钮(仅商品按钮)');
24
25  const cardDetail = await aiQuery({
26    productName: '商品名称，在价格上面',
27    productPrice: '商品价格， string',
28    productDescription: '商品描述（饮品的各种参数，吸管、冰沙等），在价格下面',
29  );
30
31  expect(cardDetail.productName).toContain('多肉葡萄');
32  expect(cardDetail.productDescription).toContain('绿妍');
33
34
```

Action Before After

https://heyteavivocity.meuu.onli...

English HEYTEA

Delivery and self pick-up not availa...

喜茶 HEYTEA (Vivocity)

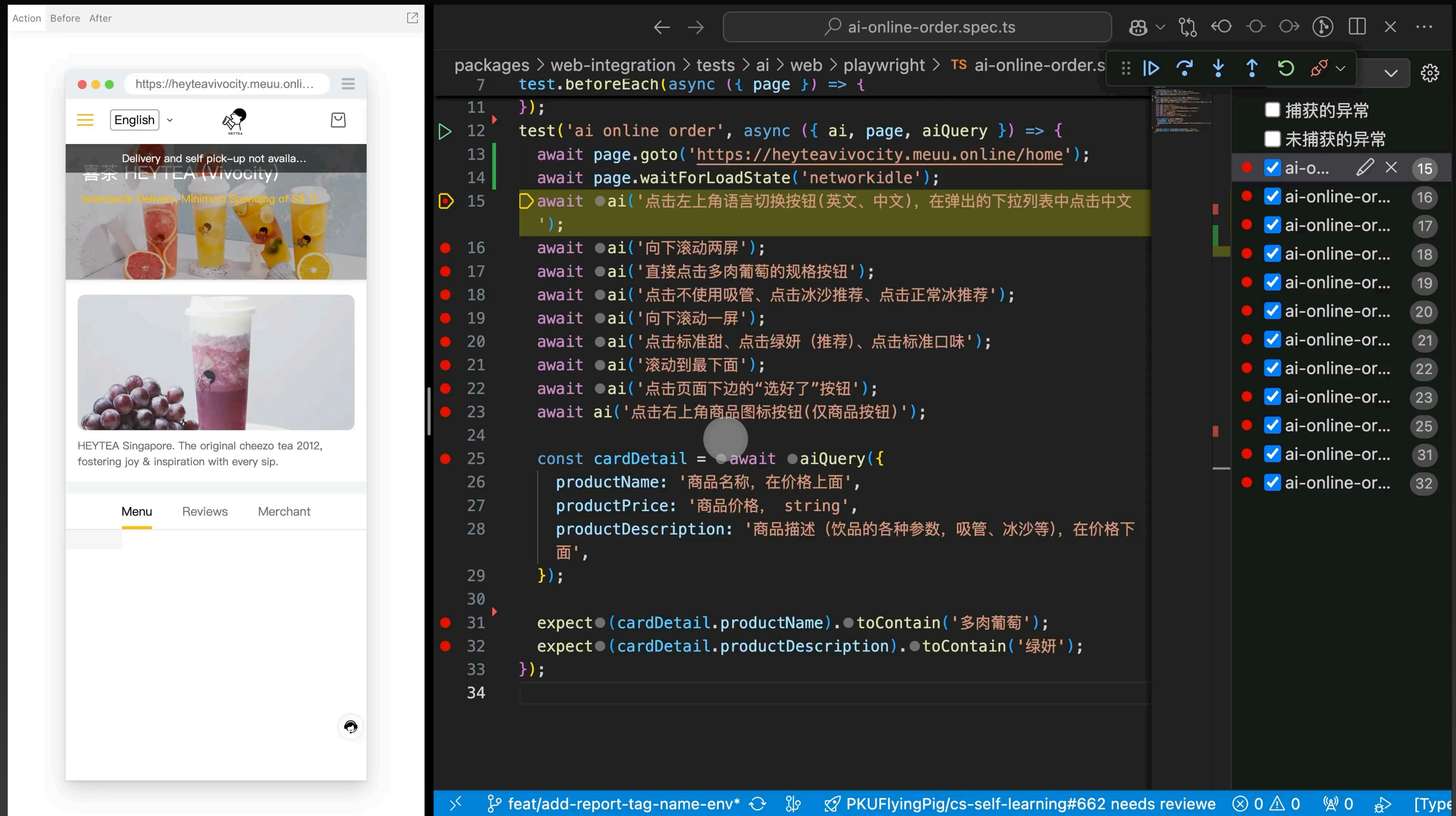
Islandwide Delivery. Minimum Spending of S\$ 11

HEYTEA Singapore. The original cheeze tea 2012, fostering joy & inspiration with every sip.

Menu Reviews Merchant

feat/add-report-tag-name-env* PKUFlyingPig/cs-self-learning#662 needs reviewe 0 △ 0 0 [Type]

02 Midscene.js 基于多模态 AI 的测试框架



The image shows a dual-pane interface. On the left is a browser window displaying the HeyTea Vivocity website. The page features a top banner with a delivery note, a large image of a drink, and a caption about HEYTEA Singapore. Below is a menu bar with 'Menu' (highlighted in yellow), 'Reviews', and 'Merchant'. On the right is a code editor showing a TypeScript test file named `ai-online-order.spec.ts`. The code uses Playwright's `ai` module to interact with the website. A specific test case is highlighted, showing steps to change language, scroll, and select items. The sidebar on the right lists multiple test runs, all of which have passed (indicated by green checkmarks). The bottom status bar shows the current file path and some GitHub repository information.

```
packages > web-integration > tests > ai > web > playwright > TS ai-online-order.s
7  test.beforeEach(async ({ page }) => {
11    });
12  test('ai online order', async ({ ai, page, aiQuery }) => {
13    await page.goto('https://heyteavivocity.meuu.online/home');
14    await page.waitForLoadState('networkidle');
15  // Dashed box highlights this block
16    await ai('点击左上角语言切换按钮(英文、中文)，在弹出的下拉列表中点击中文');
17    await ai('向下滚动两屏');
18    await ai('直接点击多肉葡萄的规格按钮');
19    await ai('点击不使用吸管、点击冰沙推荐、点击正常冰推荐');
20    await ai('向下滚动一屏');
21    await ai('点击标准甜、点击绿妍（推荐）、点击标准口味');
22    await ai('滚动到最下面');
23    await ai('点击页面下边的“选好了”按钮');
24    await ai('点击右上角商品图标按钮(仅商品按钮)');
25  const cardDetail = await aiQuery({
26    productName: '商品名称，在价格上面',
27    productPrice: '商品价格， string',
28    productDescription: '商品描述（饮品的各种参数，吸管、冰沙等），在价格下面',
29  });
30
31  expect(cardDetail.productName).toContain('多肉葡萄');
32  expect(cardDetail.productDescription).toContain('绿妍');
33});
34
```

Action Before After

https://heyteavivocity.meuu.onli...

English HEYTEA

Delivery and self pick-up not availa...

喜茶 HEYTEA (Vivocity)

Islandwide Delivery. Minimum Spending of S\$ 11

HEYTEA Singapore. The original cheeze tea 2012, fostering joy & inspiration with every sip.

Menu Reviews Merchant

feat/add-report-tag-name-env* PKUFlyingPig/cs-self-learning#662 needs reviewe 0 △ 0 0 [Type]

02 Midscene.js 基于多模态 AI 的测试框架-报告

Midscene.js Replay All ai online order - tests/ai/web/playwright/ai-online-order.spec.ts

ai online order

- 点击左上角语言切换按钮(英文、中文), 在弹出的下拉列表中点击中文
 - Planning 10.73ms
 - ✓ Insight / Locate 357ms
 - ✓ Action / Tap 375ms
 - ✓ Insight / Locate 3.29s
 - ✓ Action / Tap 629ms
- 向下滚动两屏
 - ✓ Planning 8.06s
 - ✓ Action / Scroll 376ms
 - ✓ Action / Scroll 387ms
- 直接点击多肉葡萄的规格按钮
 - ✓ Planning 19.01s
 - ✓ Insight / Locate 324ms
 - ✓ Action / Tap 435ms
- 点击不使用吸管、点击冰沙推荐、点击正常冰推荐
 - ✓ Planning 20.10s
 - ✓ Insight / Locate 390ms
 - ✓ Action / Tap 351ms
 - ✓ Insight / Locate 295ms
 - ✓ Action / Tap 345ms
 - ✓ Insight / Locate 296ms
 - ✓ Action / Tap 390ms
- 向下滚动一屏
 - ✓ Planning 20.95s
 - ✓ Action / Scroll 403ms
- 点击标准甜、点击绿妍 (推荐)、点击标准口味
 - ✓ Planning 7.01s
 - ✓ Insight / Locate 387ms
 - ✓ Action / Tap 341ms
 - ✓ Insight / Locate 269ms
 - ✓ Action / Tap 338ms
 - ✓ Insight / Locate 367ms
 - ✓ Action / Tap 408ms
- 滚动到最下面

9000ms 18000ms 27000ms 36000ms 45000ms 54000ms 63000ms 72000ms 81000ms 90000ms 99000ms 108000ms 117000ms 126000ms 135000ms

Insight Screenshots JSON View Open in Playground

Task Meta

status	finished
start	2024-11-20 15:57:12
end	2024-11-20 15:57:13
time cost	357ms
cache	{"hit":false}

Param

type	Insight / Locate
param	The language switch button with the text 'English'
quick answer	{ "reason": "It is located near the top left, is a text node. According to the screenshot, it is a language switch button with the text 'English', \"text\": \"English\", \"id\": \"877481\" } }

Output

English

```
page {  
  "underlyingPage": {  
    "_type": "Page",  
    "_guid": "  
    "page@b06d3153f705e3e17321137299032bf9"  
  },  
  "viewportSize": {  
    "width": 400,  
    "height": 905,  
    "dpr": 1  
  },  
  "pageType": "playwright"  
}  
id 877481  
indexId 1  
attributes {  
  "nodeType": "TEXT Node"  
}
```



Marker Elements

02 Midscene.js 基于多模态 AI 的测试框架-报告

Midscene.js Replay All ai online order - tests/ai/web/playwright/ai-online-order.spec.ts

ai online order

- 点击左上角语言切换按钮(英文、中文), 在弹出的下拉列表中点击中文
 - Planning 10.73s
 - ✓ Insight / Locate 357ms
 - ✓ Action / Tap 375ms
 - ✓ Insight / Locate 3.29s
 - ✓ Action / Tap 629ms
- 向下滚动两屏
 - ✓ Planning 8.06s
 - ✓ Action / Scroll 376ms
 - ✓ Action / Scroll 387ms
- 直接点击多肉葡萄的规格按钮
 - ✓ Planning 19.01s
 - ✓ Insight / Locate 324ms
 - ✓ Action / Tap 435ms
- 点击不使用吸管、点击冰沙推荐、点击正常冰推荐
 - ✓ Planning 20.10s
 - ✓ Insight / Locate 390ms
 - ✓ Action / Tap 351ms
 - ✓ Insight / Locate 295ms
 - ✓ Action / Tap 345ms
 - ✓ Insight / Locate 296ms
 - ✓ Action / Tap 390ms
- 向下滚动一屏
 - ✓ Planning 20.95s
 - ✓ Action / Scroll 403ms
- 点击标准甜、点击绿妍 (推荐)、点击标准口味
 - ✓ Planning 7.01s
 - ✓ Insight / Locate 387ms
 - ✓ Action / Tap 341ms
 - ✓ Insight / Locate 269ms
 - ✓ Action / Tap 338ms
 - ✓ Insight / Locate 367ms
 - ✓ Action / Tap 408ms
- 滚动到最下面

9000ms 18000ms 27000ms 36000ms 45000ms 54000ms 63000ms 72000ms 81000ms 90000ms 99000ms 108000ms 117000ms 126000ms 135000ms

Insight Screenshots JSON View Open in Playground

Task Meta

status	finished
start	2024-11-20 15:57:12
end	2024-11-20 15:57:13
time cost	357ms
cache	{"hit":false}

Param

type	Insight / Locate
param	The language switch button with the text 'English'
quick answer	{ "reason": "It is located near the top left, is a text node. According to the screenshot, it is a language switch button with the text 'English', \"text\": \"English\", \"id\": \"877481\" } }

Output

English

```
page {  
  "underlyingPage": {  
    "_type": "Page",  
    "_guid": "  
    "page@b06d3153f705e3e17321137299032bf9"  
  },  
  "viewportSize": {  
    "width": 400,  
    "height": 905,  
    "dpr": 1  
  },  
  "pageType": "playwright"  
}  
id 877481  
indexId 1  
attributes {  
  "nodeType": "TEXT Node"  
}
```



Marker Elements

02 Midscene.js 基于多模态 AI 的测试框架-报告

Midscene.js

Playground

Server Status

✓ Connected

[Switch to In-Browser Mode](#)

UI Context

您的购物车 0

添加的菜品 3

多肉葡萄 (首创) 5 S\$9.32

不使用吸管, 冰沙 (推荐), 正常冰 (推荐),
标准甜 (推荐), 绿妍 (推荐), 标准口味 (推荐)

关闭最低订购金额 SGD 11.00

商家暂未开通外卖和自取业务

烤黑糖波波牛乳 (首创) S\$ 5.92

轻波波牛乳 首创烤黑糖波波牛乳【轻盈版】，依然... S\$ 4.04

烤黑糖波波牛乳茶 选用定制级红茶底与冷藏牛乳灵感特调... S\$ 6.47

芋泥牛乳茶

{ "饮品信息": [{ "名称": "多肉葡萄 (首创)", "价格": "S\$9.32", "数量": "1", "选项": ["不使用吸管", "冰沙 (推荐)", "正常冰 (推荐)", "标准甜 (推荐)", "绿妍 (推荐)", "标准口味 (推荐)"] }] }

Run

Action Query Assert

提取饮品信息

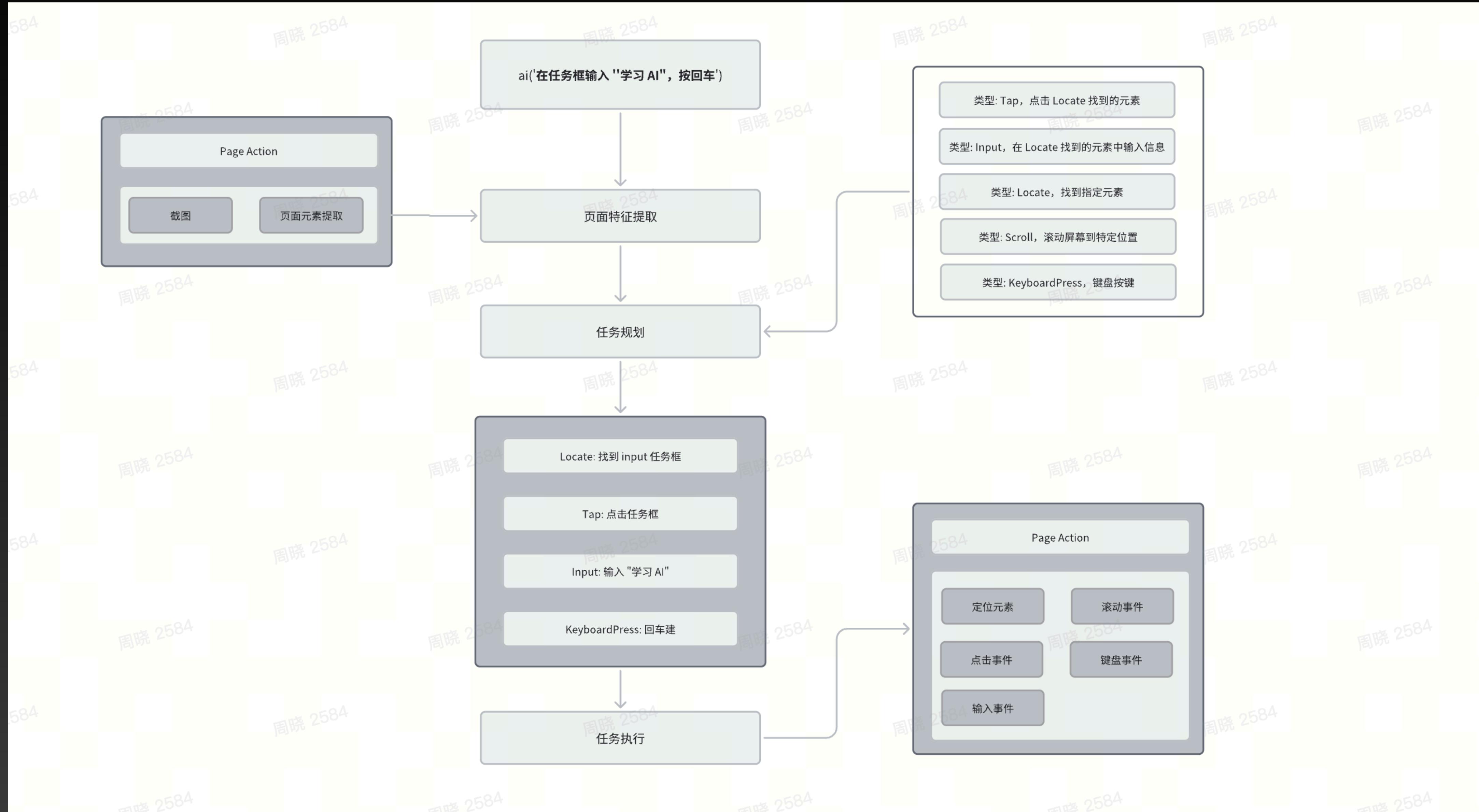
history ▾

switch: Command + Up /
Unl

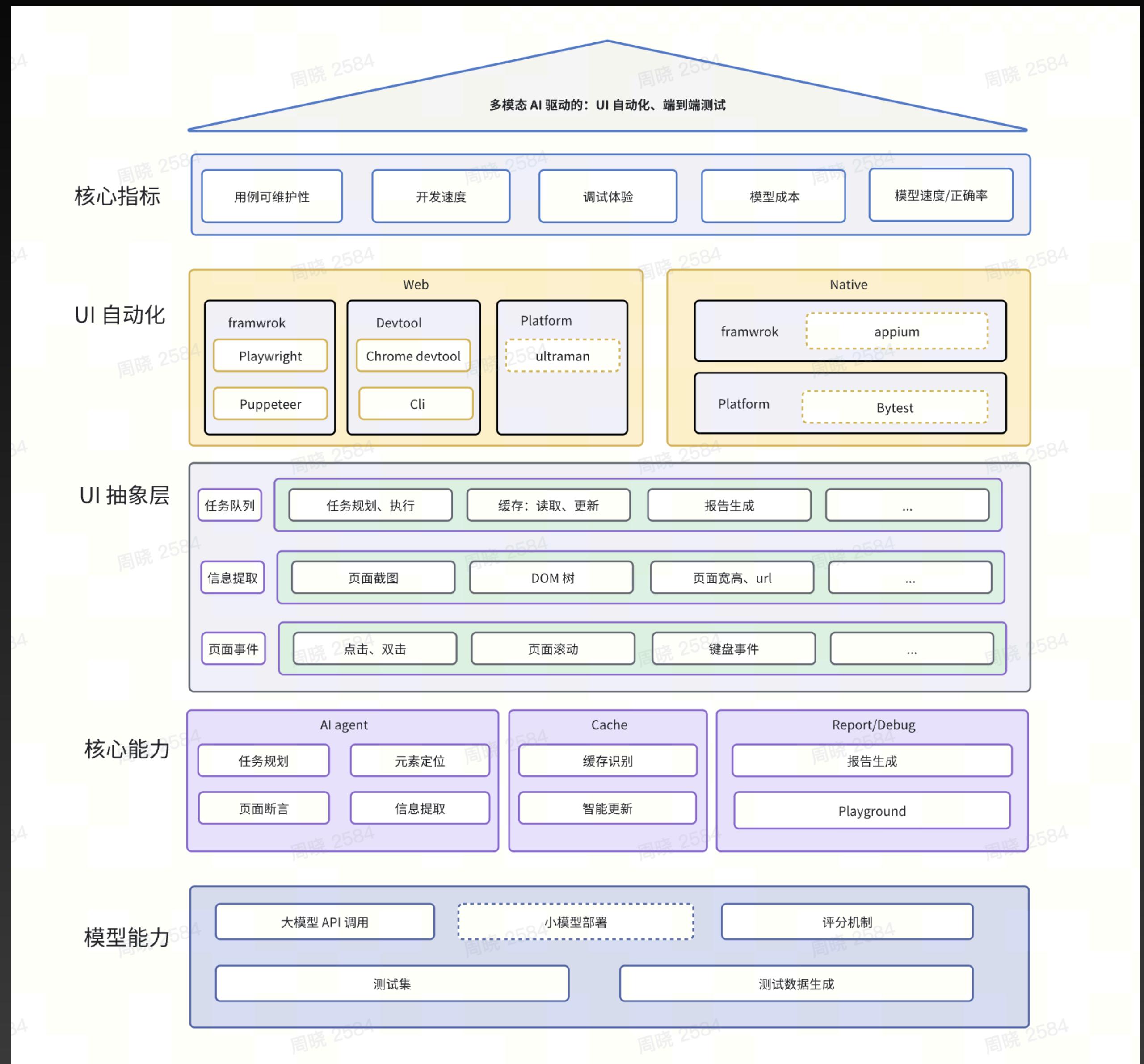
02 Midscene.js 基于多模态 AI 的测试框架

	传统 UI 自动化测试	多模态 AI 自动化测试
可维护性	★	★★★★★
可读性	★★	★★★★★
开发成本	★★★★★	★
开发体验	★	★★★★★
视觉检查能力弱	★	★★★★★

02 Midscene.js 基于多模态 AI 的测试框架-设计思路



02 Midscene.js 基于多模态 AI 的测试框架-整体架构设计



02 Midscene.js 基于多模态 AI 的测试框架-评测

The image displays a composite view of a mobile application's user interface and a side-by-side comparison of two sets of test prompts.

Left Side (Top): A dark-themed video player interface. At the top left is a circular close button with an 'X'. Next to it is a search bar containing the text "四妹". To the right of the search bar is a magnifying glass icon labeled "搜索". On the far right of the screen are vertical scroll and zoom controls. Below the video player, there is a dark overlay with white text: "@王洛同学 · 8月31日" and "第4集：究竟是谁控制了盘丝洞？《黑神话：悟空》第四回的剧情故事 #黑神话悟空 #单机游戏 #steam游戏". Underneath this is a button labeled "合集：《黑神话：悟空》 | 更新至第8集". At the bottom is a playback bar showing a red play/pause button, the time "00:00 / 00:00", and a progress bar.

Left Side (Bottom): A screenshot of a tea shop's product listing. The top part shows a banner for "喜茶 HEYTEA (Vivocity)" with images of various drinks. Below the banner is a large image of a drink with grapes. Text below the image reads: "喜茶为芝士现泡茶的原创者。自创立以来，喜茶专注于呈现来自世界各地的优质茶香，让茶饮这一古老文化焕发出新的生命力。". The main list shows categories like "时令鲜果" and "经典不踩雷". Under "时令鲜果" is a product named "多肉大杯 (首创)" with a price of "6.82". Under "经典不踩雷" is a product named "轻芒芒甘露" with a price of "6.54". Each product entry includes a small image, a price box, and a "选规格" (Select Specification) button.

Right Side (Top): A list of four test prompts. Each prompt is numbered 1 through 4 and accompanied by a dropdown arrow. The prompts are: 1. 左下角播放或暂停按钮, 2. 点赞按钮, 3. 评论按钮, 4. 书签按钮.

Right Side (Bottom): A list of six test prompts. Each prompt is numbered 1 through 6 and accompanied by a dropdown arrow. The prompts are: 1. Top left menu bar icon, 2. Toggle language text button(Could be: 中文、english text), 3. Top right shopping cart, 4. The price number on the right of the drink picture, 5. 选择规格按钮, 6. Bottom right Customer service button.

02 Midscene.js 基于多模态 AI 的测试框架-评测



模型	任务名称	平均分数	平均耗时	成功率	详细
gpt-4o-2024-05-13	todo	100.00	6.84s	100.00%	▼
gpt-4o-2024-05-13	online_order	100.00	7.11s	100.00%	▼
gpt-4o-2024-05-13	online_order_list	100.00	4.43s	100.00%	▼
gpt-4o-2024-05-13	taobao	100.00	8.58s	100.00%	▼
gpt-4o-2024-05-13	aweme_login	100.00	3.67s	100.00%	▼
gpt-4o-2024-05-13	aweme_play	100.00	8.47s	100.00%	▼
aws_claude35_sdk_sonnet	todo	100.00	7.22s	100.00%	▼
aws_claude35_sdk_sonnet	online_order	100.00	7.33s	100.00%	▼
aws_claude35_sdk_sonnet	online_order_list	91.67	9.06s	91.67%	▼
aws_claude35_sdk_sonnet	aweme_login	100.00	6.83s	100.00%	▼
aws_claude35_sdk_sonnet	aweme_play	87.50	9.83s	87.50%	▼
aws_claude35_sdk_sonnet	taobao	73.33	11.80s	73.33%	▼
gpt-4o-2024-08-06	todo	100.00	4.97s	100.00%	▼
gpt-4o-2024-08-06	online_order	100.00	5.72s	100.00%	▼
gpt-4o-2024-08-06	online_order_list	100.00	7.55s	100.00%	▼
gpt-4o-2024-08-06	taobao	95.00	6.98s	95.00%	▼
gpt-4o-2024-08-06	aweme_login	100.00	5.47s	100.00%	▼
gpt-4o-2024-08-06	aweme_play	100.00	6.82s	100.00%	▼

02 Midscene.js 基于多模态 AI 的测试框架-总结

传统 UI 自动化的局限性

- 脚本难以维护
- 视觉检查能力弱
- 无法抵抗变更

造成了开发者不愿意编写 UI 自动化脚本

Midscene.js

- 使用自然语言代替自动化脚本
- 具备视觉检查能力
- 消耗 Token

让自动化测试变成一项愉悦的工作

03 AI 在前端团队落地实践经验

03 AI 与前端技术的融合-了解 LLM 能力边界

了解 AI (LLM) 在特定场景下的能力表现

1. 文本理解
2. 数学能力
3. 图片理解
4. 图片生成

途径

1. 论文
2. benchmark

03 AI 与前端技术的融合-深入业务

明确业务需求

1. 问题定义：深入了解业务痛点，确定AI应用的具体场景和目标。
2. 可行性分析：评估AI技术在特定场景下的适用性和潜在价值

模型选择和合规问题

1. 是否有足够的数据和资源来自己训练或微调模型
2. 第三方模型需要考虑数据安全、隐私、法律等各类问题

03 AI 与前端技术的融合-建立评测机制

建立评价体系

能够通过一些指标来判断产品变得更好还是更差了

评价指标

对于确定性任务（数学计算、文本提取、函数运算...）可以采用单测的方式，对比输入和输出是否一致，来确定目前产品的评分

对于不确定任务（文本生成、图片生成...）结合业务场景，定义指标，引入人工或 AI 评测机制来进行评测

最后

Midscene.js 开源地址

<https://github.com/web-infra-dev/midscene>



shaw

浙江 杭州



扫一扫上面的二维码图案，加我为朋友。



Midscene.js