

测试报告：京东购物流程自动化测试

一、测试内容

- 打开京东首页
- 搜索指定商品 (Redmi Buds6)
- 应用商品筛选器 (京东物流)
- 点击商品进入详情页
- 将商品加入购物车

二、测试功能

1. 打开京东首页

设计思路：

- 使用 `driver.get()` 打开京东首页。
- 通过断言页面标题中包含“京东”来验证页面是否成功加载。
- 这是基础的 smoke test，用于判断京东首页是否正常访问。

目的：

验证网络连接和首页可用性，为后续操作做准备。

对应代码：

```
def open_jd_homepage(self):  
    """打开京东首页并断言标题包含‘京东’"""  
    self.driver.get("https://www.jd.com/")  
    assert "京东" in self.driver.title, "未成功打开京东首页"
```

测试截图：



小米耳机

猪猪侠报道 [领外卖神券](#) [3C数码国家补贴](#)

🏠 家用电器

📱 手机 / 运营商 / 数码

💻 电脑 / 办公 / 文具用品

🏠 家居 / 家具 / 家装 / 厨具

👕 男装 / 女装 / 童装 / 内衣

💄 美妆 / 个护清洁 / 宠物

👠 女鞋 / 箱包 / 钟表 / 珠宝

👟 男鞋 / 运动 / 户外

🚗 房产 / 汽车 / 汽车用品

👶 母婴 / 玩具乐器

🍷 食品 / 酒类 / 生鲜 / 特产

HOT 出口转内销

新品 购物车

五一放价季

每满200减30



购物车



我的



客服



反馈

2. 搜索指定商品

设计思路:

- 通过 ID 精确定位搜索框 (#key), 这是京东搜索框的唯一标识。
- 输入商品关键词 Redmi Buds6, 使用 clear() 避免历史输入影响。
- 使用 CSS 选择器 .button 定位搜索按钮并点击。

目的:

模拟用户在首页搜索商品的行为, 检验搜索功能是否可用。

对应代码:

```
def search_product(self, keyword):
    """搜索商品"""
    search_box = self.wait.until(EC.presence_of_element_located((By.ID, "key")))
    search_box.clear()
    search_box.send_keys(keyword)
    search_button = self.wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, ".button")))
    search_button.click()
```

测试截图：

Redmi Buds6白色

搜索

红米buds6 | 蓝牙耳机 | 迈从g98pro | 红米buds6pro | 键盘机械

全部商品分类

大牌奥莱

美妆馆

超市

生鲜

京东国际

拍卖

金融

全部结果 > "Redmi Buds6白色"

品牌:			赖特易耳	红米	唐古麦	ESCAS
	炫套		纪皓	原装	广齐信	小米锅巴
音质认证:	杜比全景声	XISEM audio	Hi-Res Audio	DTS		
佩戴方式:	眼镜式	头戴式	入耳式	开放式颈挂	挂耳式	耳夹式
适用型号:	左柚 FC02	左柚 FC01	小米 Redmi Buds 6青春版	小米 Redmi Buds 6s		
高级选项:	适用品牌	特色功能	场景	包装形式	类型	款式
					功能	振膜类型

商品精选

广告

【52dB】深海降噪 沉浸无打扰
CD级音质 入耳舒适

¥265.00

【2024年新款】52dB降噪耳机 沉浸音质 入耳舒适 6期白
已有10万+人评价

REDMI Buds 7S
32h超长续航

综合

销量

评论数

新品

价格

¥ - ¥

共120

配送至 山东青岛市

☐ 京东物流

☐ 国家补贴

☐ 货到付款

☒ 仅显示有货

☐ 京东国际

☐ 可配送

国家补贴

母亲节

3C数码

Redmi Buds 6 无线耳机

限时价格

¥99

精致轻巧半入耳

14.2mm超大动圈单元

内置5种降噪模式

84.15元

国家补贴 立减15%

¥99.00

国家补贴

母亲节

3C数码

Redmi Buds 6 无线耳机

限时价格

¥139

智慧 AI 通话降噪

42dB深度主动降噪

12.4mm 超大动圈单元

118.15元

国家补贴 立减15%

¥139.00

3. 应用商品筛选器

设计思路：

- 使用 XPath 定位筛选器中“京东物流”选项（第一个特征项）。
- 使用 element_to_be_clickable 等待元素加载完成。
- 点击后页面会自动刷新以应用过滤。

目的:

验证筛选器功能是否正常运作, 且页面是否能正确响应用户的筛选行为。

对应代码:

```
def apply_filter(self):
    """筛选商品"""
    filter_element = self.wait.until(EC.element_to_be_clickable(
        (By.XPATH, '//*[@id="J_feature"]/li[1]/i')
    ))
    filter_element.click()
```

测试截图:



4. 点击商品进入详情页

设计思路：

- 使用 CSS Selector 定位第一个商品的标题文字区域（`#J_goodsList .gl-item:nth-child(1) .p-name em`）。
- 加入显式等待，确保元素可点击。
- 使用 `assert` 判断元素是否存在，提升测试健壮性。
- 点击商品后会打开新页面。

目的：

确保点击商品后能够正确跳转到详情页。

对应代码：

```
def click_product(self):
    """点击第一个搜索到的商品"""
    product_name = self.wait.until(EC.element_to_be_clickable(
        (By.CSS_SELECTOR, "#J_goodsList .gl-item:nth-child(1) .p-name em")
    ))
    assert product_name is not None, "商品元素未找到"
    product_name.click()
```

测试截图：



5. 添加商品到购物车

设计思路：

- 首先调用 `switch_to_new_window()` 切换到商品详情页。
- 等待并点击“加入购物车”按钮（通过 ID `InitCartUrl` 定位）。
- 使用断言检查加入购物车按钮存在，作为添加成功的标志。

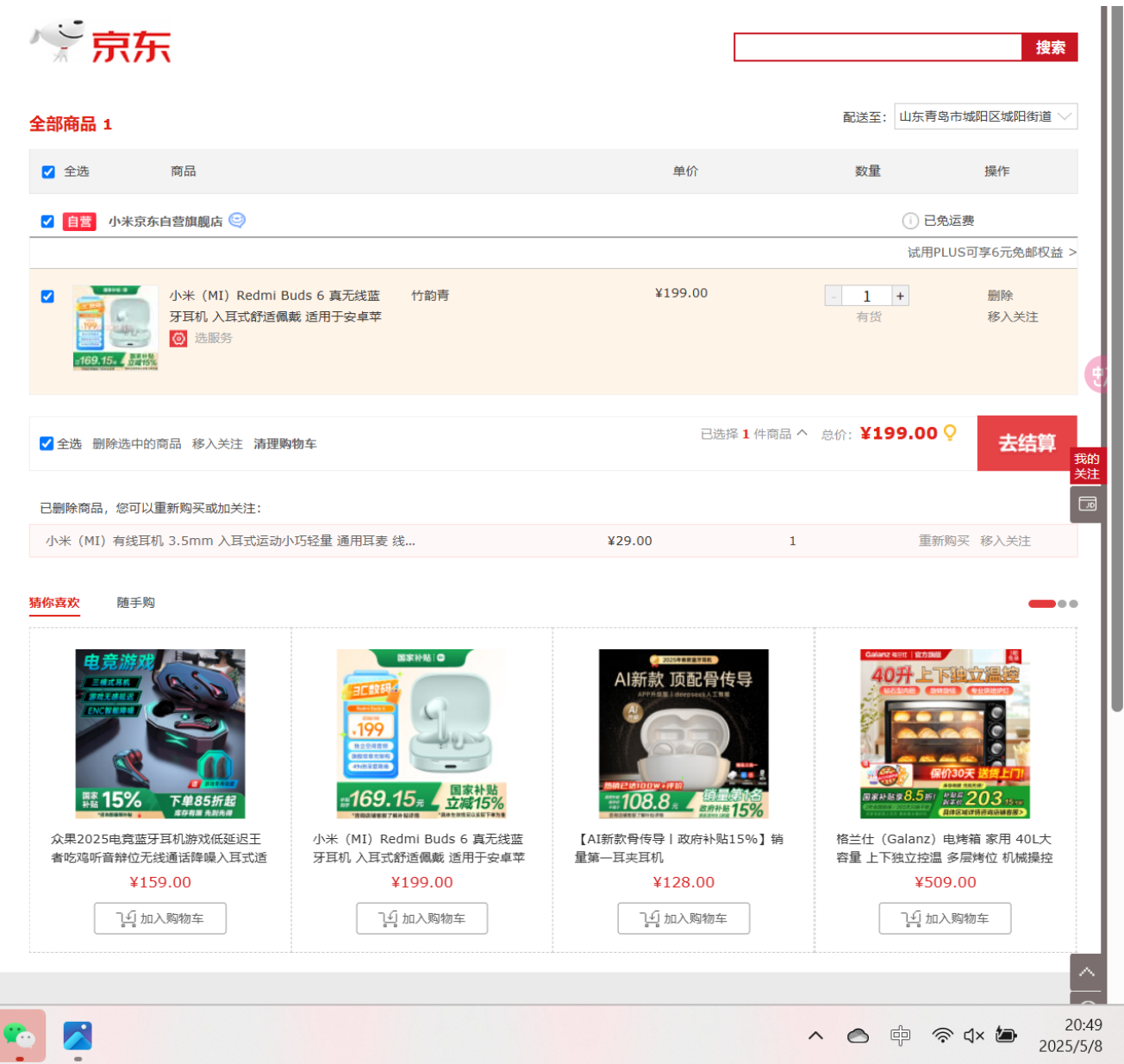
目的：

模拟购物操作的最后一步，验证加入购物车功能是否正常。

对应代码：

```
def add_to_cart(self):
    """添加商品到购物车"""
    self.switch_to_new_window()
    add_to_cart_button = self.wait.until(EC.element_to_be_clickable((By.ID, "InitCartUrl")))
    add_to_cart_button.click()
    assert self.driver.find_element(By.ID, "InitCartUrl"), "添加到购物车失败"
```

测试截图：



三、总代码

```
import pytest
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.firefox.options import Options
from selenium.webdriver.firefox.firefox_profile import FirefoxProfile

class TestJDShopping:
    def setup_method(self, method):
        # 初始化Firefox浏览器
        # profile_path =
r"C:\Users\95432\AppData\Roaming\Mozilla\Firefox\Profiles\q8uyn2xu.default-release"
        # profile = FirefoxProfile(profile_path)
        # options = Options()
        # self.driver = webdriver.Firefox(firefox_profile=profile, options=options)
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.wait = WebDriverWait(self.driver, 50)

    def teardown_method(self, method):
        # 测试结束关闭浏览器
        self.driver.quit()

    def open_jd_homepage(self):
        """打开京东首页并断言标题包含'京东'"""
        self.driver.get("https://www.jd.com/")
        assert "京东" in self.driver.title, "未成功打开京东首页"

    def search_product(self, keyword):
        """搜索商品"""
        search_box = self.wait.until(EC.presence_of_element_located((By.ID, "key"))) # 通过ID定位
        search_box.clear()
        search_box.send_keys(keyword)
        search_button = self.wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR,
".button")))) # 通过CSS定位
        search_button.click()

    def apply_filter(self):
        """筛选商品"""
        filter_element = self.wait.until(EC.element_to_be_clickable(
            (By.XPATH, '//*[@id="J_feature"]/li[1]/i') # 使用XPath定位
        ))
        filter_element.click()

    def click_product(self):
        """点击第一个搜索到的商品"""
        product_name = self.wait.until(EC.element_to_be_clickable(
            (By.CSS_SELECTOR, "#J_goodsList .gl-item:nth-child(1) .p-name em")
        ))
        assert product_name is not None, "商品元素未找到"
        product_name.click()

    def add_to_cart(self):
        """添加商品到购物车"""
        self.switch_to_new_window()
```

```
add_to_cart_button = self.wait.until(EC.element_to_be_clickable((By.ID, "InitCartUrl")))
add_to_cart_button.click()
# 断言购物车按钮存在
assert self.driver.find_element(By.ID, "InitCartUrl"), "添加到购物车失败"

def switch_to_new_window(self):
    """切换到新打开的窗口"""
    current_handles = self.driver.window_handles
    time.sleep(2) # 简单等待新页面打开
    new_handles = self.driver.window_handles
    for handle in new_handles:
        if handle not in current_handles:
            self.driver.switch_to.window(handle)
            break

@pytest.mark.run(order=1)
def test_jd_shopping_flow(self):
    """完整测试流程"""
    self.open_jd_homepage()
    self.search_product("Redmi Buds6")
    self.apply_filter()
    self.click_product()
    self.add_to_cart()
```