

Cypress

An introduction

- Introduction
- Front end testing tool
- Easy to set up
- (Relatively) easy to use

● Overview

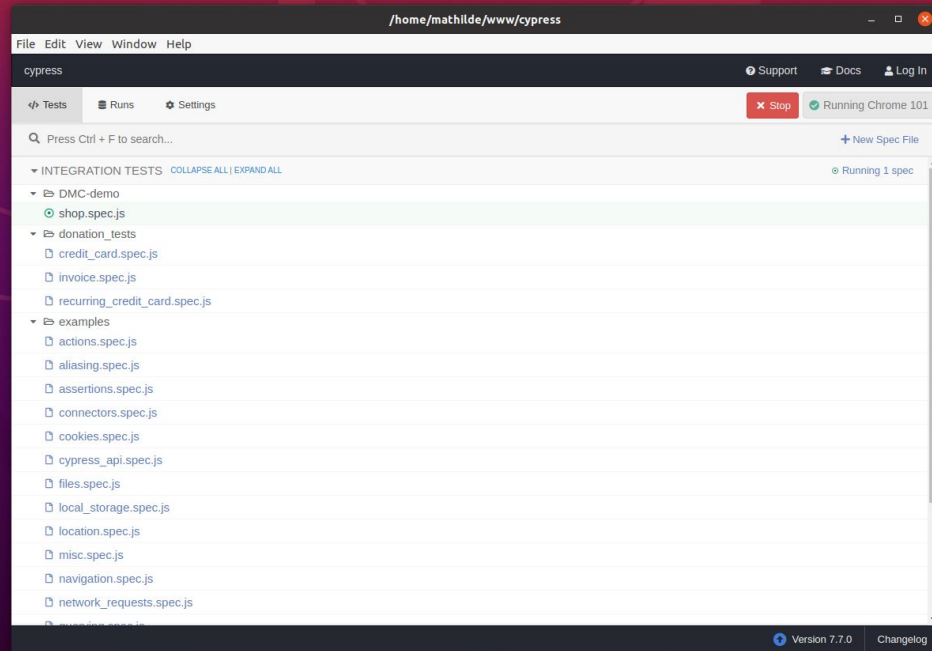
- Easy to install
- Easy to use: An example
- Writing tests and interacting with Drupal
- More

● Easy to install

- `npm install cypress`
`npx cypress open`

- Also available as a docker image, fully working

Easy to use



Easy to use

The image shows a Cypress test runner interface with three main sections:

- Test steps:** A list of test actions such as `visit`, `contains`, `invoke`, `get`, `type`, `click`, and `assert`. It includes error messages like `TypeError: Drupal.toolbar.ToolbarModel is not a construc...` and `ReferenceError: Backbone is not defined`.
- Live display of the test:** A browser window showing a table of orders. The table has columns for ID, date, shop, merchandise, user, status, price, and actions. The data includes entries for 'Shop Merchandise' with various dates and prices.
- Your beloved inspector:** The Chrome DevTools 'Elements' panel showing the DOM structure of the application, including `<div id="app">`, `<div class="runner-container">`, and `<div class="size-container">`.

● Writing tests

- Tests are javascript based

```
context('Credit card Shopping', () => {  
  beforeEach(() => {  
    cy.visit('/shop')  
    cy.get('.eu-cookie-compliance-secondary-button')  
      .click()  
  })  
  
  it('buys 2 t-shirts', () => {  
    cy.get('h3:contains("TShirts")')  
      .click()  
    cy.get('.content-item__summary:contains("DrupalCamp T-Shirt")')  
      .click()  
    cy.get('.product__description')  
      .should('contain', 'Dieses T-Shirt besteht aus 100% Baumwolle und respektiert die Umwelt.')  
    cy.getByLabel('Size')  
      .select('L')  
  })  
})
```

- jQuery syntax is supported

Interacting with Drupal

- Through a custom API

```
Cypress.Commands.add('getOrder', (order_id) => {  
  let url = '/cypress/script/getOrder';  
  if(Cypress.config('CYPRESS_XDEBUG')) {  
    url = url.concat('?XDEBUG_SESSION_START=phpstorm');  
  }  
  let drupal_api_key = Cypress.env('drupal_api_key');  
  return cy.request('POST', url, {  
    drupal_api_key,  
    order_id,  
  }).then(response => {  
    return response.body;  
  });  
});
```

Custom Cypress command

```
cy.getCurrentOrderId().then(($order_id) => {  
  cy.getOrder($order_id).as('order')  
  cy.get('@order')  
    .its('item_0').as('order_item_0')  
  cy.get('@order')  
    .its('total_price')  
    .should('eq', '100CHF')  
});
```

Custom module with API in the website

```
/**  
 * Execute arbitrary PHP scripts from cypress.  
 */  
class ScriptController extends ControllerBase {  
  
  /**  
   * Returns the details of the order, given the order id.  
   */  
  public function getOrder() {  
    $content = Json::decode(\Drupal::request()->getContent());  
    if (!$content) {  
      new Response('Request body has to be JSON.', status: 400);  
    }  
  }  
}
```

```
cypress.getorder:  
  path: '/cypress/script/getOrder'  
  methods: [POST]  
  defaults:  
    _controller: '\Drupal\primer_cypress\Controller\ScriptController::getOrder'  
  requirements:  
    _custom_access: '\Drupal\primer_cypress\Controller\ScriptController::checkAPIKey'
```

Call in the test

● More

- Extensive and intuitive documentation
- Can be integrated in the CI
- (Kind of) supports visual testing.
- Code coverage report



Thank you for your attention

Any questions?