

## Assignments Cypress



A training without assignments is like a ship model in a box. The model looks beautiful on the box in the store.

You can take the package home with you, but as long as you don't take the time to assemble the package, it will never be a three-master.

So I would like you to practise what you have learned during the training. The assignments below will help you with this.

First, I will give you a number of tips that increase the chance that your assignment can actually be carried out. Then I also give you background information about the importance of assignments. And of course the elaboration of the assignment is not lacking.

After completing these assignments, you will have developed a basis for further developing the test by testing other functions.

Good luck with the assignments!

*Rob Flier*

### Exercise 1:

*Visit the webpage and check if you are on the homepage*

To do this exercise you need the following commands:

- `cy.visit()`
- `cy.title()`
- `cy.viewport()`

url: <http://localhost:8080/parabank/index.htm?ConnType=JDBC>

Use: <https://docs.cypress.io/api/commands/visit> as a reference if you don't know how the function work.

When I want to cheat:

In your Cypress project folder you have a folder called "Integration". In this folder create a new test file "parabank.spec.js" and copy the code below:

```
/ <reference types="Cypress" / >
beforeEach("Parabank", () => {
  cy.visit("http://parabank.parasoft.com/parabank/index.htm?ConnType=JDBC");
  cy.title().should("include", "ParaBank");
  cy.viewport(1680, 1200);
});
```

## Exercise 2:

### *Log in with username and password*

To do this exercise you need the following commands:

- `cy.get()`
- `cy.click()`

Kopieer de onderstaande code:

```
it("should be possible to do my banking affairs", () => {  
  //  
  cy.log('voer hier je test in');  
  //  
});
```

You can log in with: User name: *john* & Password: *demo*

Tip: Use F12 (right mouse button: Inspect (CTRL-shift-I) to select elements from the web page (<https://www.youtube.com/watch?v=97NbmSA1zWo>).  
Commonly used term is "Dev Tools".

When I want to cheat:

```
// Log in with username and password  
cy.get('[class="input"]').first().type("john");  
cy.get('[class="input"]').last().type("demo");  
cy.get('[class="button"]').last().click();
```

Explanation: because both the username and password have the same class, you must indicate which field you mean. For example use "first" and "last" in your test.

### Exercise 3:

#### *Open a new account number (Open New Account)*

To do this exercise you need the following commands:

- `cy.get()`
- `cy.click()`
- `cy.contains()`
- `cy.wait()`

Open a new savings account number (SAVINGS) and check whether the new account exists and that it has an initial deposit.

Tip: in Dev Tools / Elements you can find a link (href) as well as an id in addition to a class. You can also test these elements.

When I want to cheat:

Also try to use the green comment lines in your test.

```
// Open New Account
cy.get('a[href*="/parabank/openaccount.htm"]').click();
// Try this
// Comment line 16 by putting 2 forward slashes '//' in front of the command and the line will be turned green
// Uncomment line 20 by deleting the forward slashes
// cy.contains("Open New Account").click(); // Alternative for line 16
cy.get("#type").select("SAVINGS");
cy.wait(500).get('[class="button"]').click();
cy.wait(500).get("#newAccount").click();
cy.get("#accountType").contains("SAVINGS");
cy.get("#balance").contains("$100.00");
```

## Exercise 4:

### *Apply for a loan (Request Loan)*

To do this exercise you need the following commands:

- `cy.get()`
- `cy.click()`
- `cy.contains()`
- `cy.wait()`

Apply for a loan of \$1,000 with a monthly payment of \$100 and verify that the new account exists and that it has an initial investment.

Tip: in Dev Tools / Elements you can also find a text after the (href). This time, use the text to make a test.

When I want to cheat:

You can also use forward slashes instead of quotes to avoid using uppercase and lowercase characters in your text. You can also use parts of the text field using forward slashes.

```
// Request loan
cy.contains("Request loan").click();
// cy.findbyText("Request loan").click(); // Alternative for above line
// cy.get('[href*="/parabank/requestloan.htm"]').click(); // Alternative for above line
cy.wait(500).get("#amount").type("100");
cy.get("#downpayment").type("100");
cy.get('[class="button"]', {contains: "Apply Now"}).click();
cy.wait(500).contains("Congratulations, your loan has been approved");
// Je kunt ook slashes gebruiken i.p.v. quotes om het gebruik van hoofd- en kleine letters in je tekst te omzeilen
cy.wait(500).contains("/congratulations, your loan has been approved/");
// Of je kunt m.b.v. slashes ook delen van de tekst gebruiken i.p.v. de hele tekst
cy.wait(500).contains("/congratulations/");
cy.wait(500).contains("/your loan has been approved/");
```

## Exercise 5:

### *Make a bank transfer (Transfer Funds)*

To do this exercise you need the following commands:

- `cy.get()`
- `cy.click()`
- `cy.contains()`
- `cy.wait()`
- `cy.clear()`
- `cy.within()`

Make a transfer from account number 13344 of, for example, \$ 1000 to your newly opened new savings account.

Make a conscious mistake in the overwrite field (Amount) and catch the error message in your test.

Tip: The Cypress documentation states exactly why and how you can use the within clause (<https://docs.cypress.io/api/commands/within.html#Usage>).

When I want to cheat:

```
// Transfer Funds
cy.contains('Transfer Funds/i).click();
cy.get("#rightPanel").within(() => {
  cy.wait(500);
  // First enter an invalid amount on purpose
  cy.get("#amount").type("$134.55");
  cy.get("#toAccountId").select("13455");
  cy.wait(500).click();
  // Then clear your input and fill in a valid amount
  cy.get("#amount").clear().type("134.55");
  cy.wait(500).contains('Please enter a valid amount./i);
  // Then clear your input and fill in a valid amount
  cy.wait(500).get('i[class="button"]').contains("Transfer").click();
});
```

## Exercise 6:

### *Check your debits and credits (Accounts Overview)*

To do this exercise you need the following commands:

- `cy.get()`
- `cy.click()`
- `cy.contains()`
- `cy.wait()`
- `cy.should()`
- `cy.eq()`

1. Check your most recent debit to account number 13344 and your transfer to your savings account.
2. Then also check whether your loan with the amount requested is visible.
3. Check for each account number whether it is a checking account, or a savings account or loan.

When I want to cheat:

```
// Accounts Overview
// Afboeking
cy.contains('Accounts Overview/i').click();
cy.contains('13344').click();
cy.get('#accountType').should('contain', 'CHECKING');
cy.wait(1500)
  .get('tbody>tr')
  .eq(-1)
  .contains('/134.55/i');
// Bijboeking
cy.contains('Accounts Overview/i').click();
cy.contains('13455').click();
cy.get('#accountType').should('contain', 'SAVINGS');
cy.wait(1500).get('tbody>tr').eq(-1).click();
cy.wait(500).contains('/134.55/i');
// Leenbedrag (al bijgeschreven?)
cy.contains('Accounts Overview/i').click();
cy.contains('13566').click();
cy.get('#accountType').should('contain', 'LOAN');
cy.wait(500).contains('/1000.00/i');
```