

The purpose of this challenge is to evaluate your testing automation skills. The challenge consists on two different exercises about both UI and API testing automation.

- Tests should be written in Gherkin (BDD).
- Any programming language can be used, although Java is a plus.
- Any framework/tool can be used.
- The use of *design patterns* is a plus. If so, please explain them.
- The solution must be platform-agnostic.
- Add a README.md file at the project root with instructions to build the project and run the tests from the command line. Other documentation can be included too, if necessary.
- All the software required to build and run the project must be included in the project itself.

Exercise 1 - UI automation

This exercise is about writing automated tests to validate the following features from a UI point of view.

Login feature, <https://eu.devo.com/login> :

1. Define the **three** most valuable test cases from your point of view. Please, justify your choice.
2. Automate **two** of them. Please, justify your choice.

Timezone feature: define test case and automate a test that validates the timezone feature within the web. Main considerations:

1. Timezone can be changed by clicking on the Timezone option in the main menu (below username and domain). Within the modal, select a timezone from any of the countries available in the dropdown.
2. The selected timezone should be applied only for the current session.

To log in to the application you should have received an invitation in your email to register in Devo. In case you didn't receive it, please contact us.

Exercise 2 - API automation

This exercise consists on writing automated tests to validate the Query API of Devo described on <https://docs.devo.com/confluence/ndt/v7.3.0/api-reference/query-api>. The API consists on several requests but just the following two have to be automated for this exercise.

1) **Query request:** get data from a query from the EU region.

a) Requirements:

- i) QueryId: "a41f6b48-9d9d-47e2-abda-b77451b09058"
- ii) Token: "3b27ee980b87284463da35a89cdb0374"
- iii) From: *1 hour before (relative date)*
- iv) To: *now (relative date)*
- v) Mode type: csv

b) Add all validation that you consider necessary in the response but at least:

- i) The response header Content-Type is
"text/csv; charset=utf-8; header=present"
- ii) eventdate values are between dates required
- iii) eventdate values should follow the pattern
"YYYY-MM-dd' 'HH:mm:ss.SSS";
- iv) eventdate, method, protocol columns are returned

2) **Job request:** remove a query from the EU region.

a) Requirements:

i) Existing running job J1:

- (1) QueryId: "a41f6b48-9d9d-47e2-abda-b77451b09058"
- (2) Token: "3b27ee980b87284463da35a89cdb0374"
- (3) From: *1 hour before (relative date)*
- (4) To: *(continuous query)*
- (5) Mode type: json
- (6) Destination Type: email
- (7) Email to: *(use your email address)*

ii) J1 jobId (you can get it from the response when you create it or from the job list request)

b) Add all validation that you consider necessary but at least:

- i) The response header `Content-Type` is `"application/json"`
- ii) `status` is `"0"`
- iii) `object.status` is `"REMOVED"`
- iv) `object.id` is `J1 jobID`
- v) `Table` is `demo.ecommerce.data`
- vi) `Domain` is `qacodechallenge`

Note: *Oauth token provided could be used instead of API key/secret in all the requests.*

Note: *QueryID provided could be used instead of LinQ query in all the requests.*