

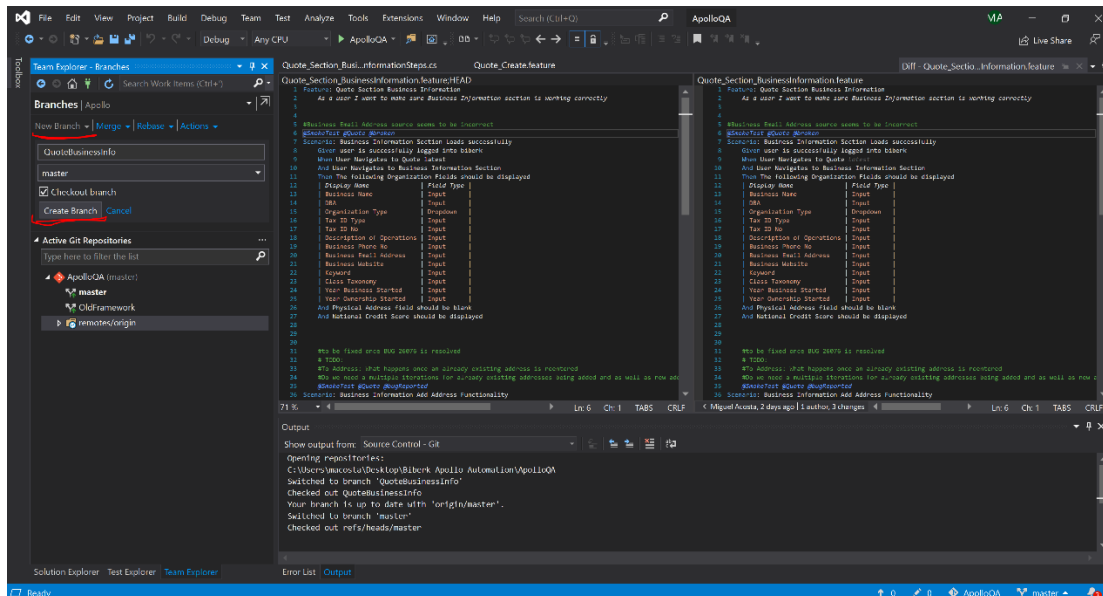
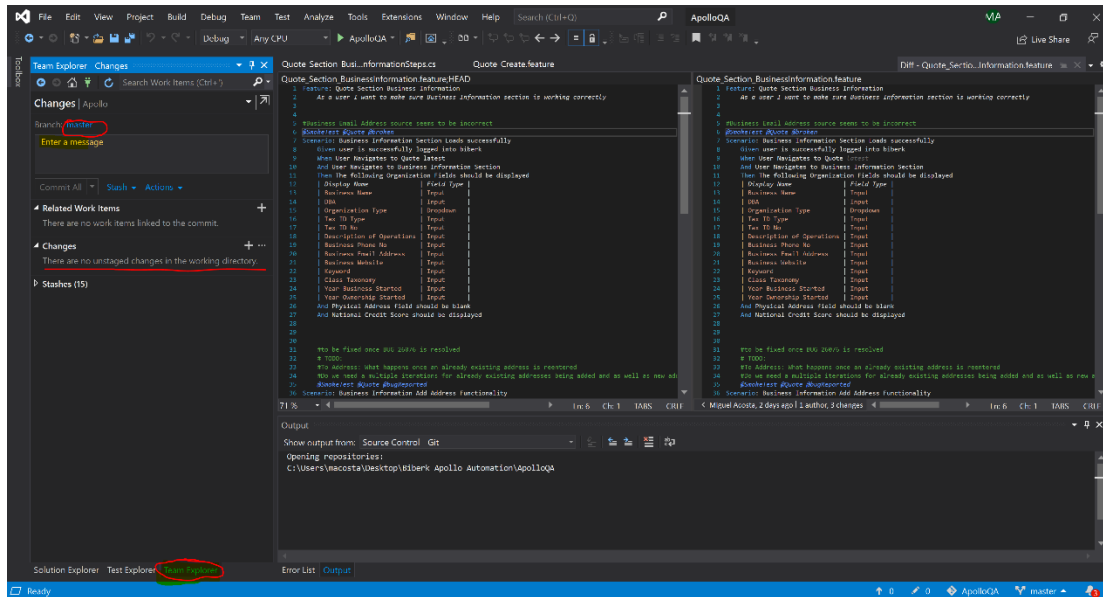
## Test Automation Development Guidelines

Creating a new branch .....	2
Commit changes.....	3
Pushing changes.....	4
Run test in Azure Pipeline .....	5
Create a Pull Request .....	6
Link work items .....	7
Set auto complete.....	8

**\*Important:** An automation Task (Azure work item) should have one to one relationship to a specific branch.  
A task is not considered complete until a pull request has been completed for it

## Creating a new branch

1. Make sure current branch is Master with no changes
2. Make sure name is according to the user story or functionality to be automated

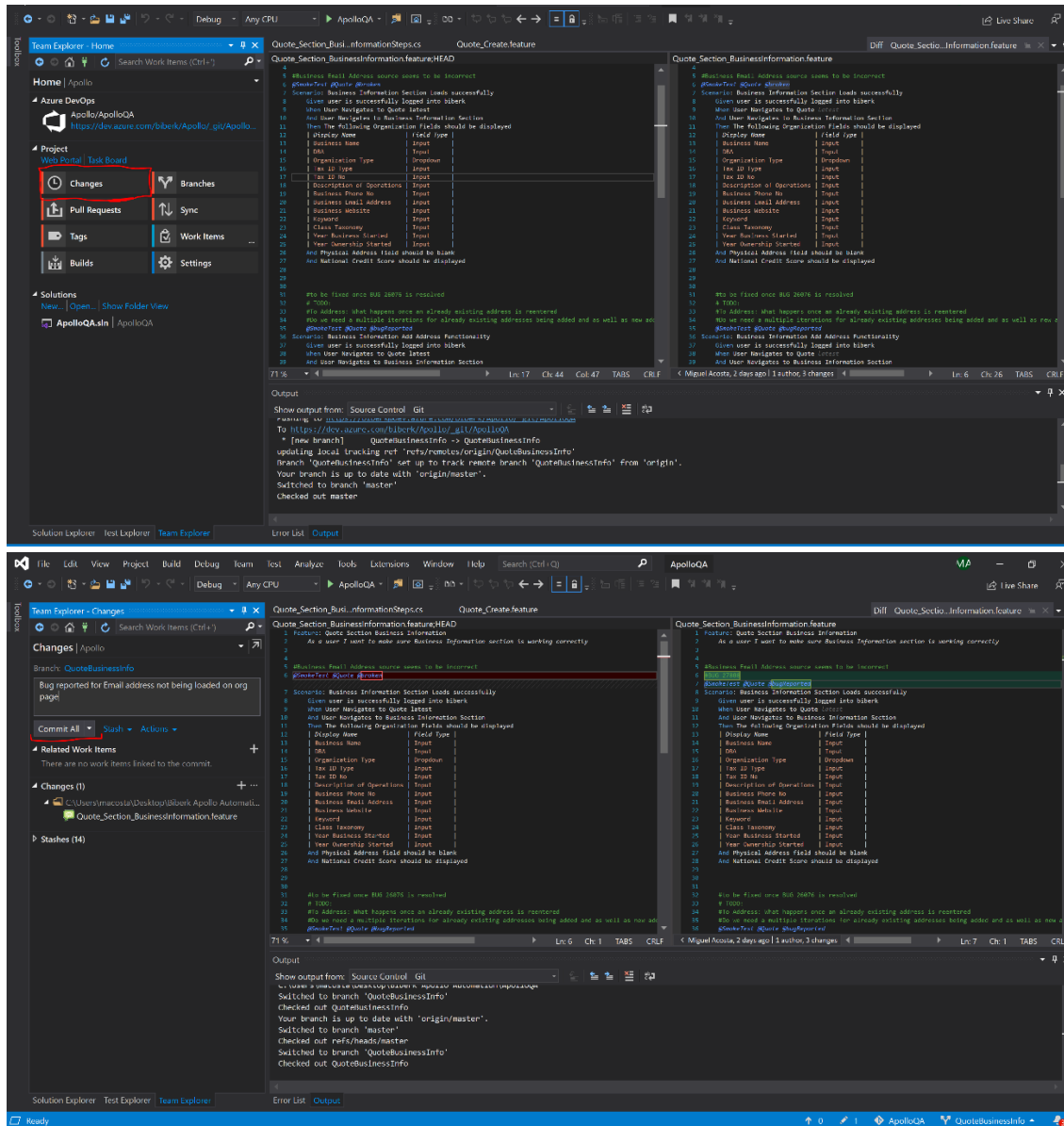


# Commit changes

ideally one should commit for every scenario completed, one could also commit once the whole User Story or functionality is automated

note: is ideal to divide changes into various commits to then push everything altogether

## 1. Commit with a brief description of the changes



# Pushing changes

Once the whole user story or functionality is completed and working locally

1. Push the previously committed changes to the previously created branch

The image consists of two screenshots of the Visual Studio IDE, illustrating the steps to push changes to a remote branch.

**Top Screenshot:** The 'Team Explorer' pane on the left shows the 'Quote\_BusinessInfo' branch selected. The 'Changes' pane shows a list of changes, with the 'Quote\_BusinessInfo' branch highlighted. The 'Sync' button is visible. The main editor area displays the 'Quote\_Create.feature' file, which contains Gherkin syntax for a user story. The 'Output' pane at the bottom shows the command prompt output, indicating that the branch 'Quote\_BusinessInfo' was created locally and is now up to date with the remote.

**Bottom Screenshot:** The 'Team Explorer' pane on the left shows the 'Quote\_BusinessInfo' branch selected. The 'Changes' pane shows a list of changes, with the 'Quote\_BusinessInfo' branch highlighted. The 'Sync' button is visible. The main editor area displays the 'Quote\_Create.feature' file, which contains Gherkin syntax for a user story. The 'Output' pane at the bottom shows the command prompt output, indicating that the branch 'Quote\_BusinessInfo' was created locally and is now up to date with the remote.

## Run test in Azure Pipeline

To make sure the implemented solution is working in the Azure Pipeline, our official test runner, we're going to run the pipeline for the previously created branch.

1. Navigate to [ApolloQA Pipeline](#) in Azure DevOps
2. Run the pipeline for the previously created branch

The screenshot displays the Azure DevOps web interface for the 'ApolloQA' pipeline. The left sidebar shows the navigation menu with 'Pipelines' selected. The main area shows a list of pipeline runs. A 'Run pipeline' dialog box is open in the foreground, allowing manual execution of the pipeline.

**ApolloQA Pipeline Runs**

Description	Stages	Time
#20201221.2 Removed colons causing reporting issues Manually triggered for master d9674e1	1	2h ago 25m 10s
#20201221.1 Tagged org tests as bugReported Manually triggered for master 14fad22	2	3h ago 42m 12s
#20201218.13 Global Search adding bugReported tag Manually triggered for master 09601e5	1	Friday 12m 7s
#20201218.12 Global Search adding bugReported tag Manually triggered for master 09601e5	1	Friday 38m 27s
#20201218.11 Global Search adding bugReported tag Manually triggered for master 09601e5	1	Friday 9m 58s
#20201218.10 Global Search adding bugReported tag Manually triggered for master 09601e5	1	Friday 13m 7s
#20201218.9 minor bug fix Manually triggered for master 0cb7b4f	1	Friday <1s
#20201218.8 minor bug fix Manually triggered for master 0cb7b4f	1	Friday 12m 27s
#20201218.7 excluding broken scenarios	-	Friday

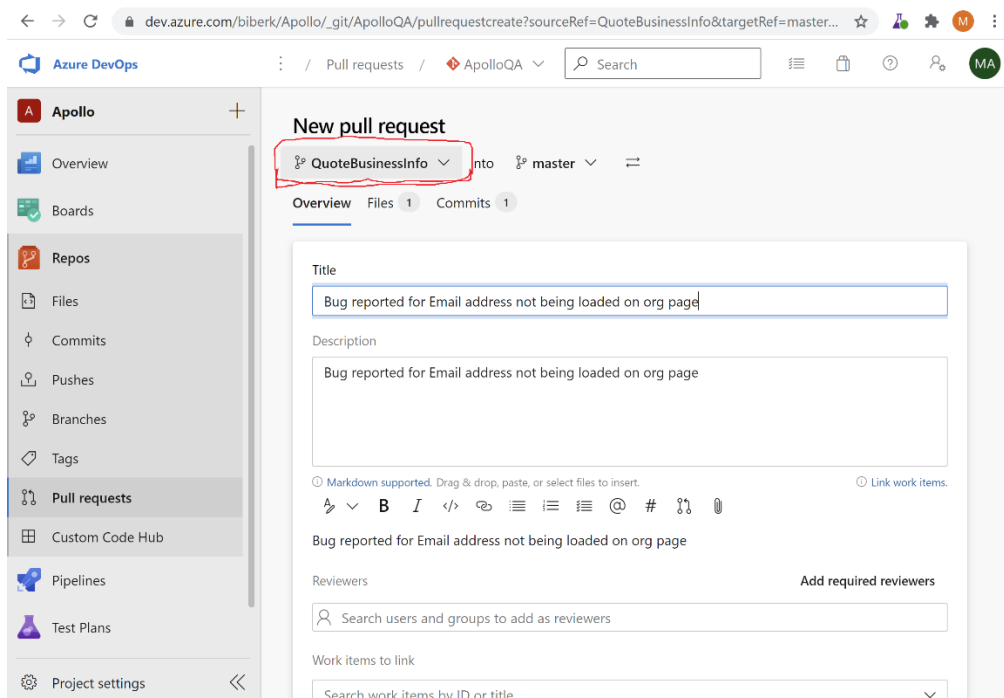
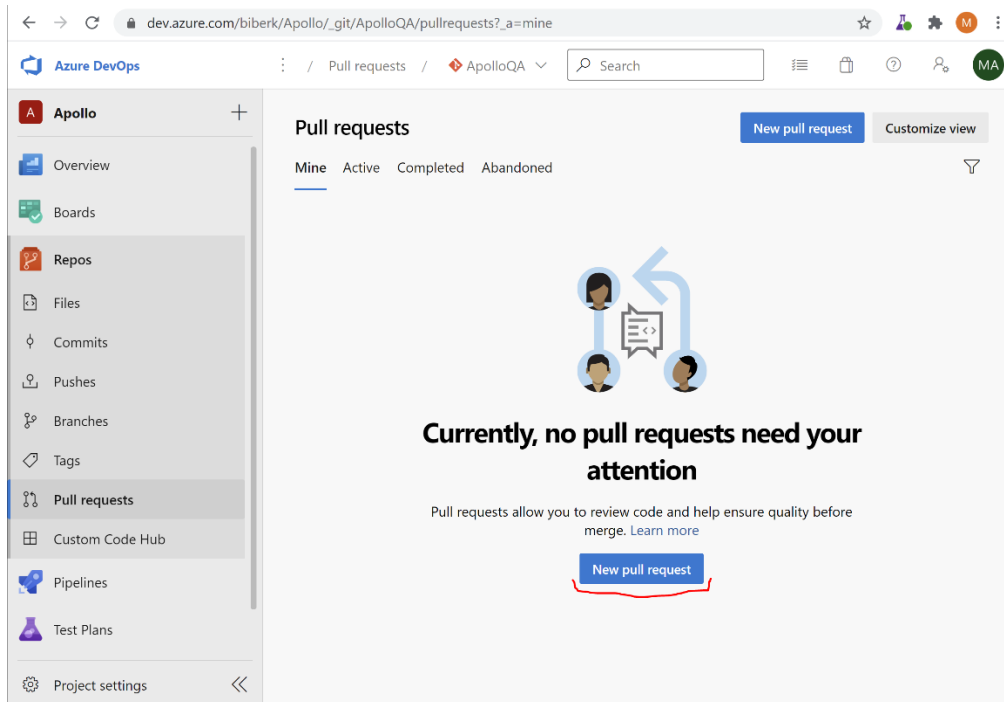
**Run pipeline dialog box:**

- Branch/tag:** QuoteBusinessInfo
- Advanced options:**
  - Variables:** This pipeline has no defined variables
  - Stages to run:** Run as configured
  - Resources:** Use latest version of all resources
- ☐ Enable system diagnostics
- Buttons:** Cancel, Run

## Create a Pull Request

Once the pipeline test is successful, a pull request should be created to be reviewed before merging into master

1. Navigate to [Pull Requests](#) in Azure DevOps
2. Click on the “New Pull Request” Button
3. Select the previously created branch
4. Click on the “Create” Button



## Link work items

Every pull request must have two linked work items associated.

1. Related Task
2. Test Case must be created/updated according to the work done.
3. Test Case must be added to:  
Regression Testing->Automated.

The screenshot displays the Azure DevOps interface for the 'Apollo' project. The left sidebar contains a navigation menu with the following items: Overview, Boards, Repos, Pipelines, Test Plans (selected), Test plans, Progress report, Parameters, Configurations, Runs, Load test, Load test, Custom Test Hub, and Artifacts. The main content area shows the 'Test Plans' section for the 'Apollo' project. At the top, it indicates the date range 'Jan 3 - Jan 10' and the status 'Past' with a 'View report' link. Below this, the 'Test Suites' section lists various test suites, including 'Automated (47)', which is highlighted with a red box. Other test suites listed include 'Smoke Test', 'Manual', and 'Future (7)'. The breadcrumb navigation at the top reads 'biberk / Apollo / Test Plans / Apollo\_TestPlar'.

## Set auto complete

Once work for the related task is fully completed and tested in the pipeline with no failures. The PR can be set to auto complete.

This will trigger completion after all requirements have been met.

