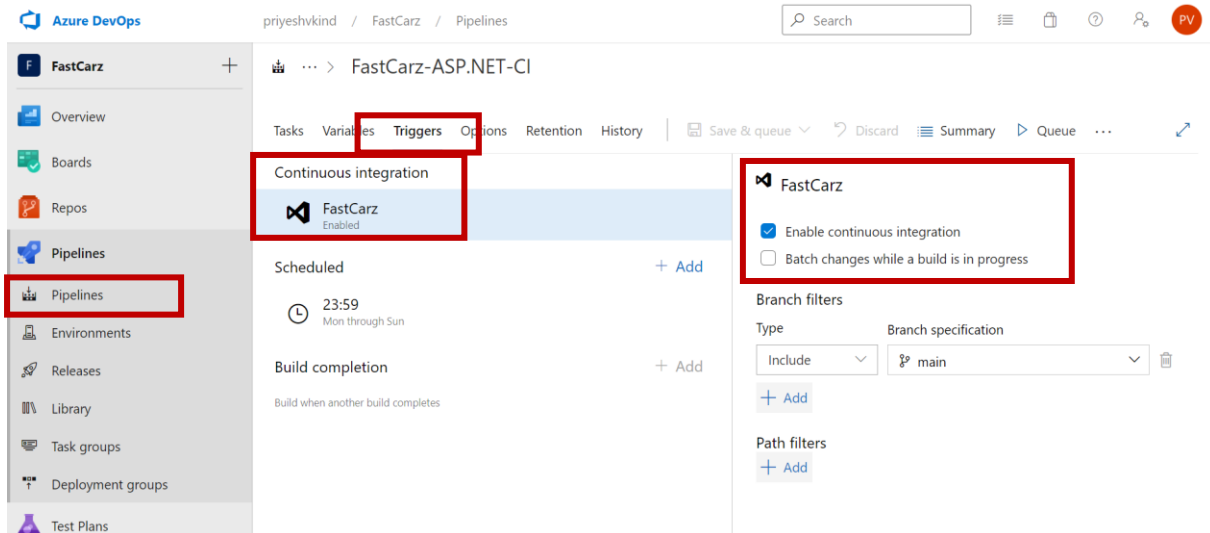


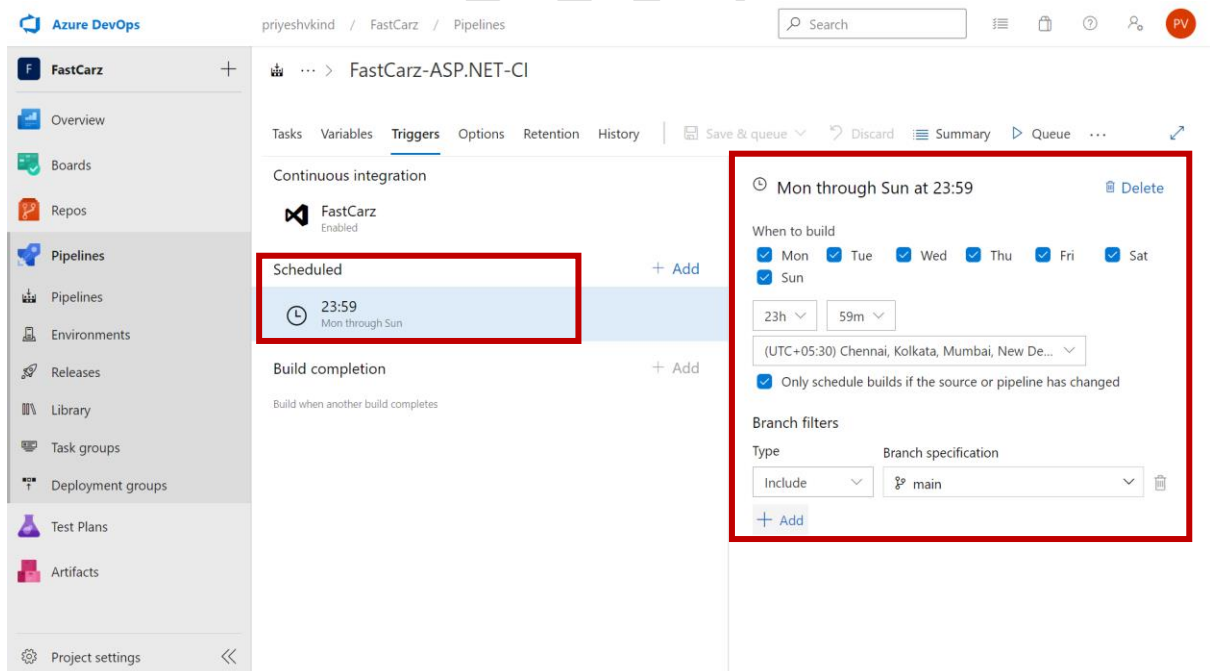
1) The build should trigger as soon as anyone in the dev team checks in code to master branch.

Solution: For creating triggers on any or master (main) branch, we have to:

1. Click on pipelines, go to the build pipeline
2. Click on triggers & within Continuous integration, we have to select the branch name & check enable continuous integration
3. Make sure the branch filters has the master (main) branch

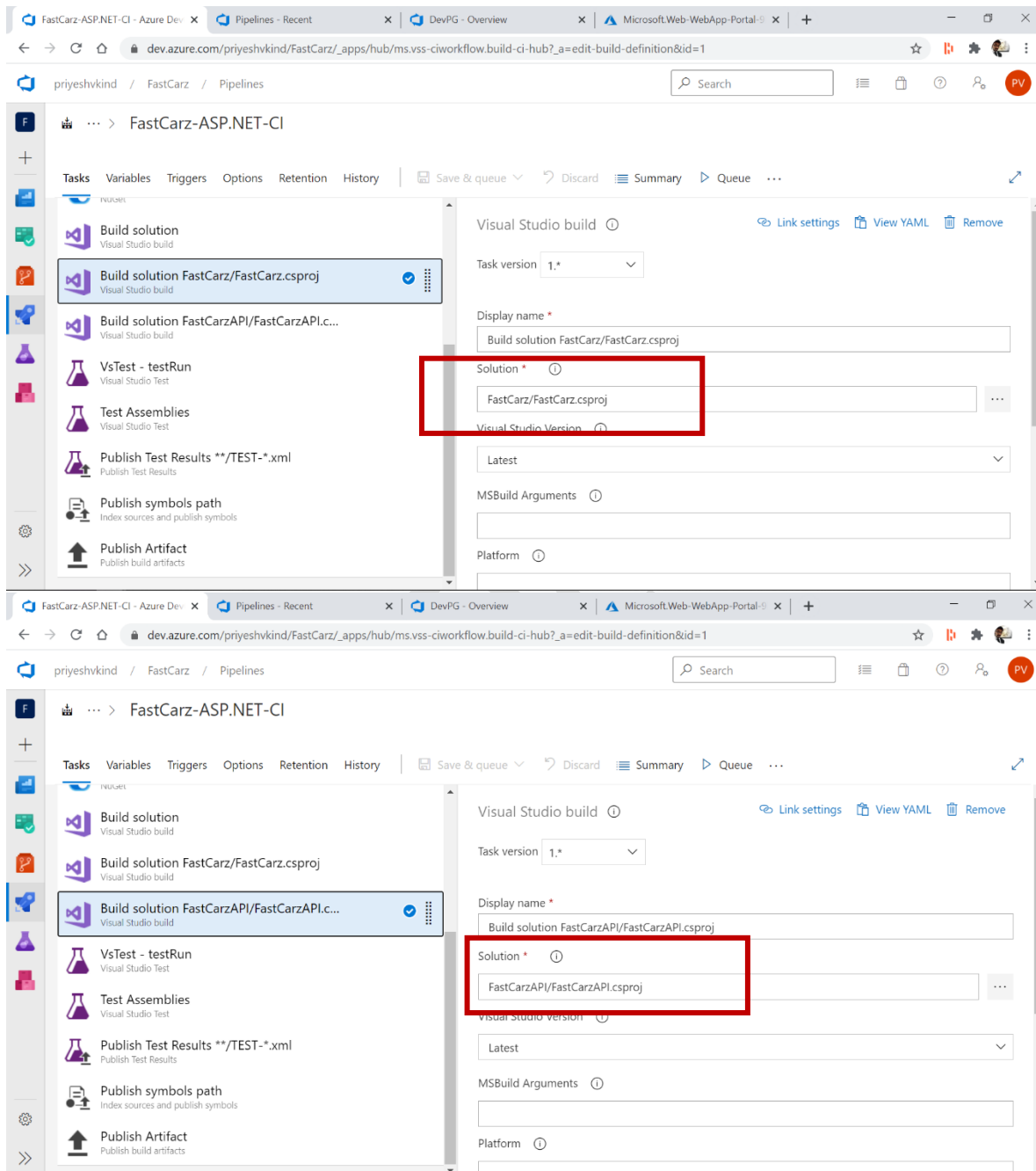


Moreover, because the build used to run every night, we could schedule the build within triggers:



2) There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test. The build should not be successful if any test fails.

Solution: For building all the projects within a single trigger, we can include separate tasks within the pipeline for multiple project builds, specifying the CSProj that we want to build for.

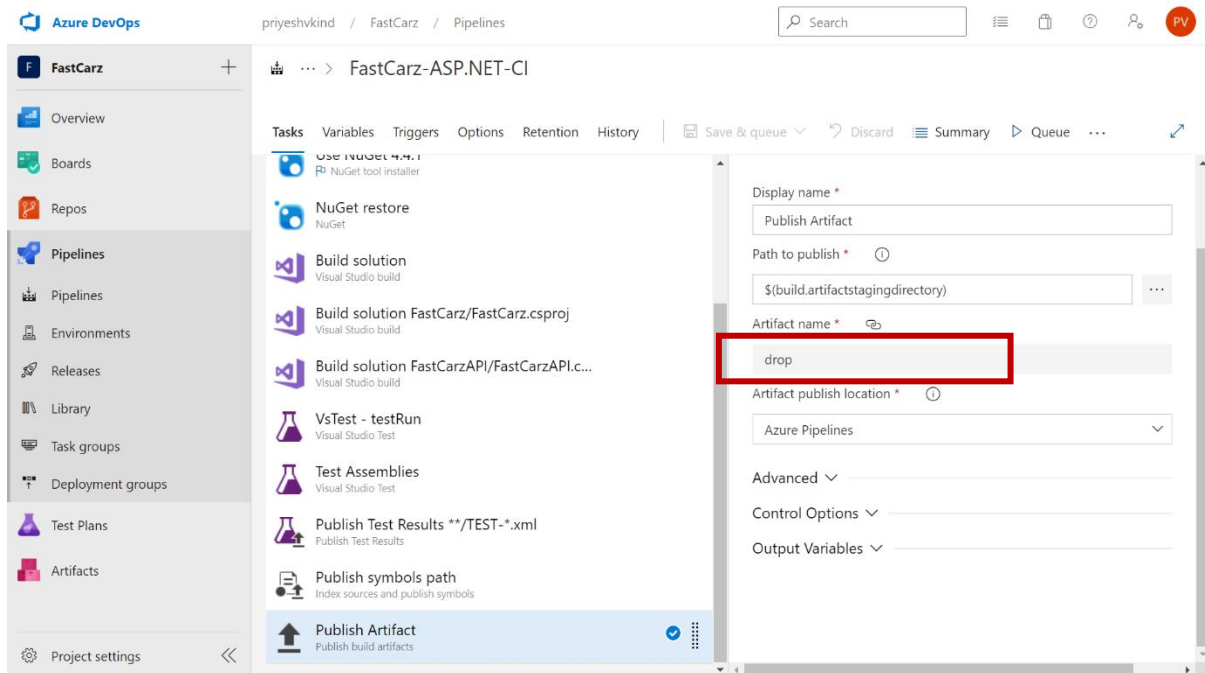


For build to fail on test failures, We have to check “Fail if there are test failures” within the publish test results task

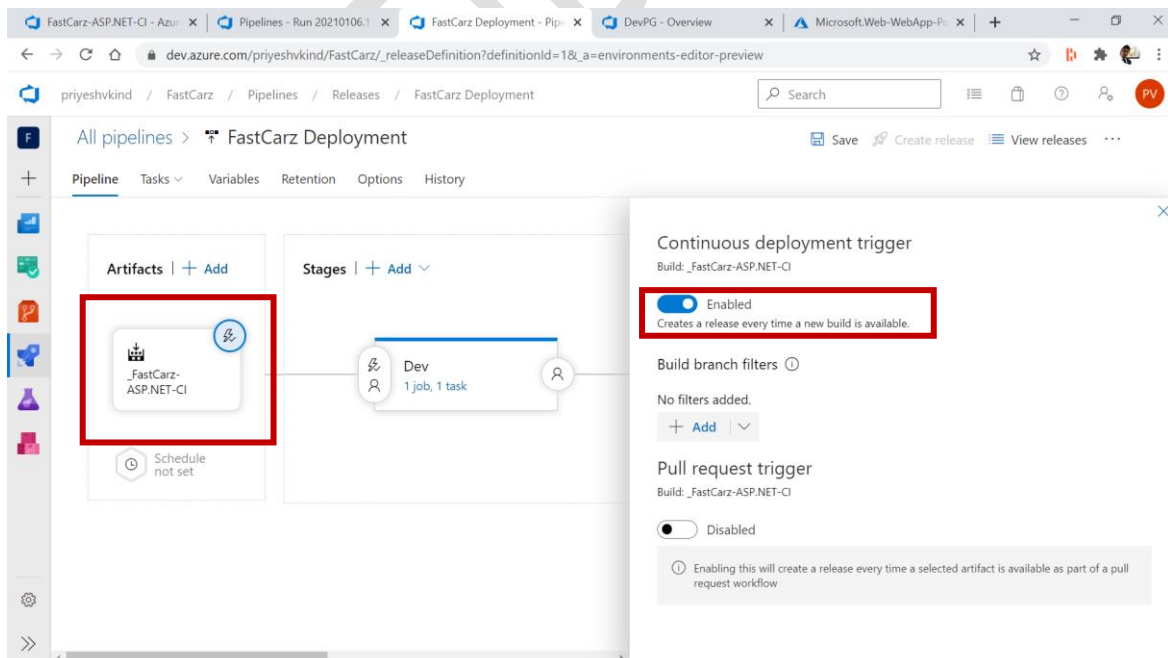
The screenshot displays the Azure DevOps web interface for a pipeline named 'FastCarz-ASP.NET-CI'. The left sidebar shows the 'Pipelines' section. The main area shows the task list, with 'Publish Test Results \*\*/TEST-\*.xml' selected. The right pane shows the configuration for this task. The 'Test result format' is set to 'NUnit'. The 'Test results files' field contains '\*\*/TEST-\*.xml'. The 'Search folder' is set to '\$(System.DefaultWorkingDirectory)'. The checkbox 'Fail if there are test failures' is checked and highlighted with a red box. Other options like 'Test run title', 'Advanced', 'Control Options', and 'Output Variables' are also visible.

3) The deployment of code and artifacts should be automated to Dev environment.

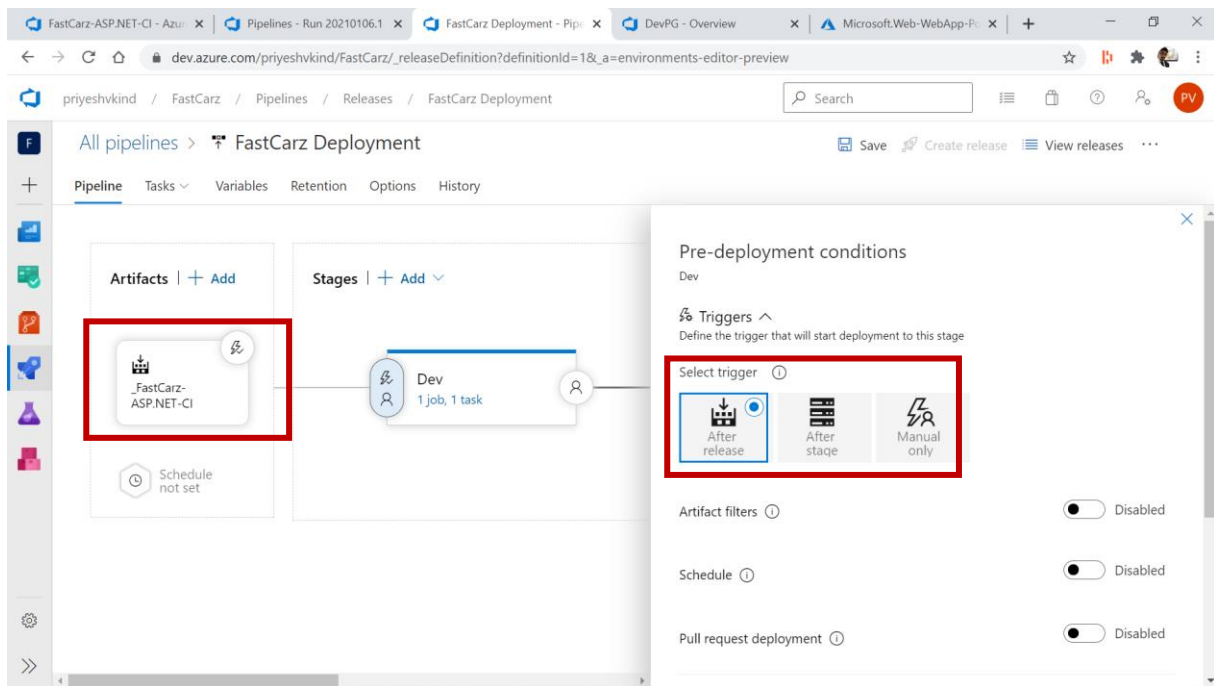
Solution: The build pipeline artifacts should be published within the drop folder using the publish artifact task.



Then, we have to consume it within the release pipeline for Dev Stage. This could be done by enabling Continuous Deployment trigger.

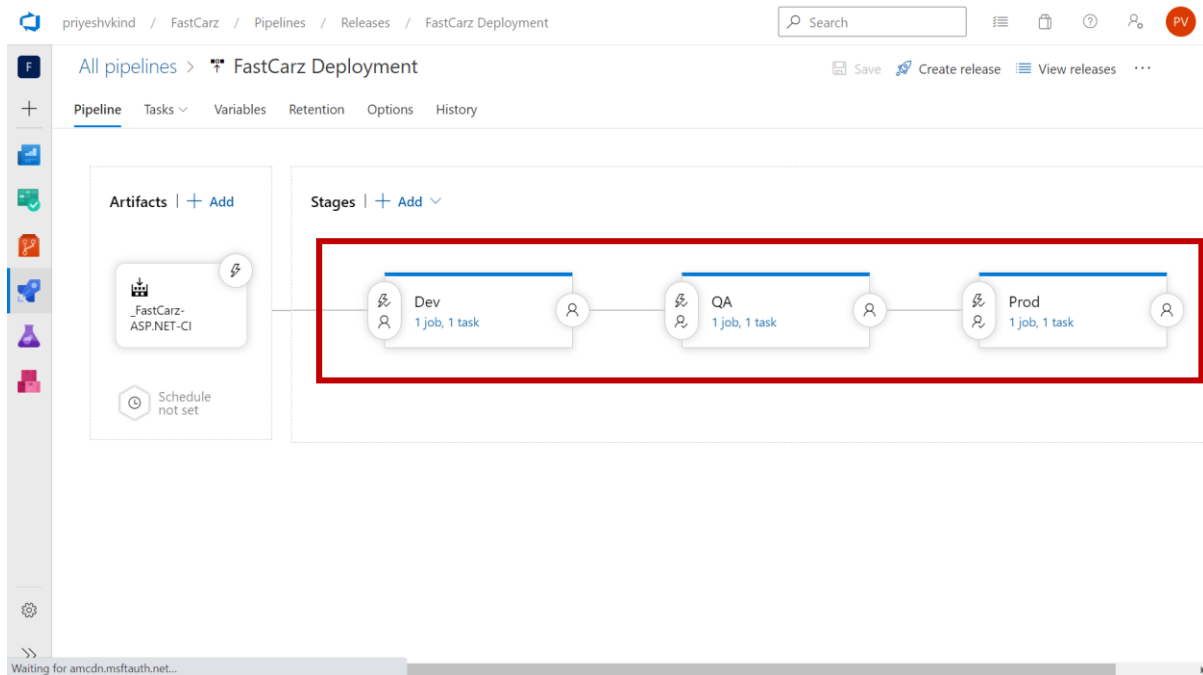


Also, within the Dev stage, “after release” trigger needs to be selected:



4) Upon successful deployment to the Dev environment, deployment should be easily promoted to QA and Prod through automated process.

Solution: We need to enable multi-stage deployment. This could be done by creating multiple stages within the release pipeline & enabling triggers after every stage.



5) The deployments to QA and Prod should be enabled with Approvals from approvers only.

Solution: Pre-deployment approvals should be enabled for deployment on QA & Prod

The image displays two screenshots of the Azure DevOps web interface, specifically the 'FastCarz Deployment' pipeline configuration page. Both screenshots show the 'Pre-deployment conditions' panel for a specific stage, with the 'Pre-deployment approvals' section highlighted by a red rectangle.

**Top Screenshot (QA Stage):**

- Stage:** QA (1 job, 1 task)
- Pre-deployment approvals:** Enabled. The 'Approvers' list includes 'Priyesh Vishwakarma'.
- Timeout:** 30 Days.
- Approval policies:** ☐ The user requesting a release or deployment should not approve it. ☐ Skip approval if the same approver approved the previous stage.
- Gates:** Disabled.

**Bottom Screenshot (Prod Stage):**

- Stage:** Prod (1 job, 1 task)
- Pre-deployment approvals:** Enabled. The 'Approvers' list includes 'Priyesh Vishwakarma'.
- Timeout:** 30 Days.
- Approval policies:** ☐ The user requesting a release or deployment should not approve it. ☐ Skip approval if the same approver approved the previous stage.
- Gates:** Disabled.

NOTE: the above solutions are being configured & screenshotted from my own account within Azure DevOps

The image displays three screenshots of the Azure DevOps web interface, specifically the 'History' tab for a pipeline named 'FastCarz-ASP.NET-CI'. The screenshots are arranged vertically, showing different views of the pipeline's execution history.

**Top Screenshot:** Shows the 'History' tab with a table of changes. A red box highlights the 'Changed Date' column, showing a single entry: '06-01-2021 15:11'. Another red box highlights the user profile of 'Priyesh Vishwakarma' in the top right corner.

**Middle Screenshot:** Shows the 'History' tab with a table of changes. A red box highlights the 'Changed Date' column, showing five entries: '1/6/2021, 8:38 PM', '1/6/2021, 8:37 PM', '1/6/2021, 8:36 PM', '1/6/2021, 8:36 PM', and '1/6/2021, 8:35 PM'.

**Bottom Screenshot:** Shows the 'History' tab with a table of changes. A red box highlights the 'Changed Date' column, showing five entries: '06-01-2021 15:11', '06-01-2021 14:59', '06-01-2021 14:56', '06-01-2021 14:54', and '06-01-2021 14:52'. The 'Comment' column shows 'Init build' and 'scheduling' for the first two entries.

Changed By	Change Type	Changed Date	Comment
Priyesh Vishwakarma	Update	06-01-2021 15:11	
Priyesh Vishwakarma	Update	06-01-2021 14:59	Init build
Priyesh Vishwakarma	Update	06-01-2021 14:56	scheduling
Priyesh Vishwakarma	Update	06-01-2021 14:54	
Priyesh Vishwakarma	Add	06-01-2021 14:52	