

Setup accounts, repo and workflow

Prerequisites

You'll need three accounts for this workshop, all services have free options that should be more than enough:

- [GitHub](#)
- [Bitrise](#)
- [Expo](#)
- Expo Go app on your phone ([iOS](#), [Android](#))

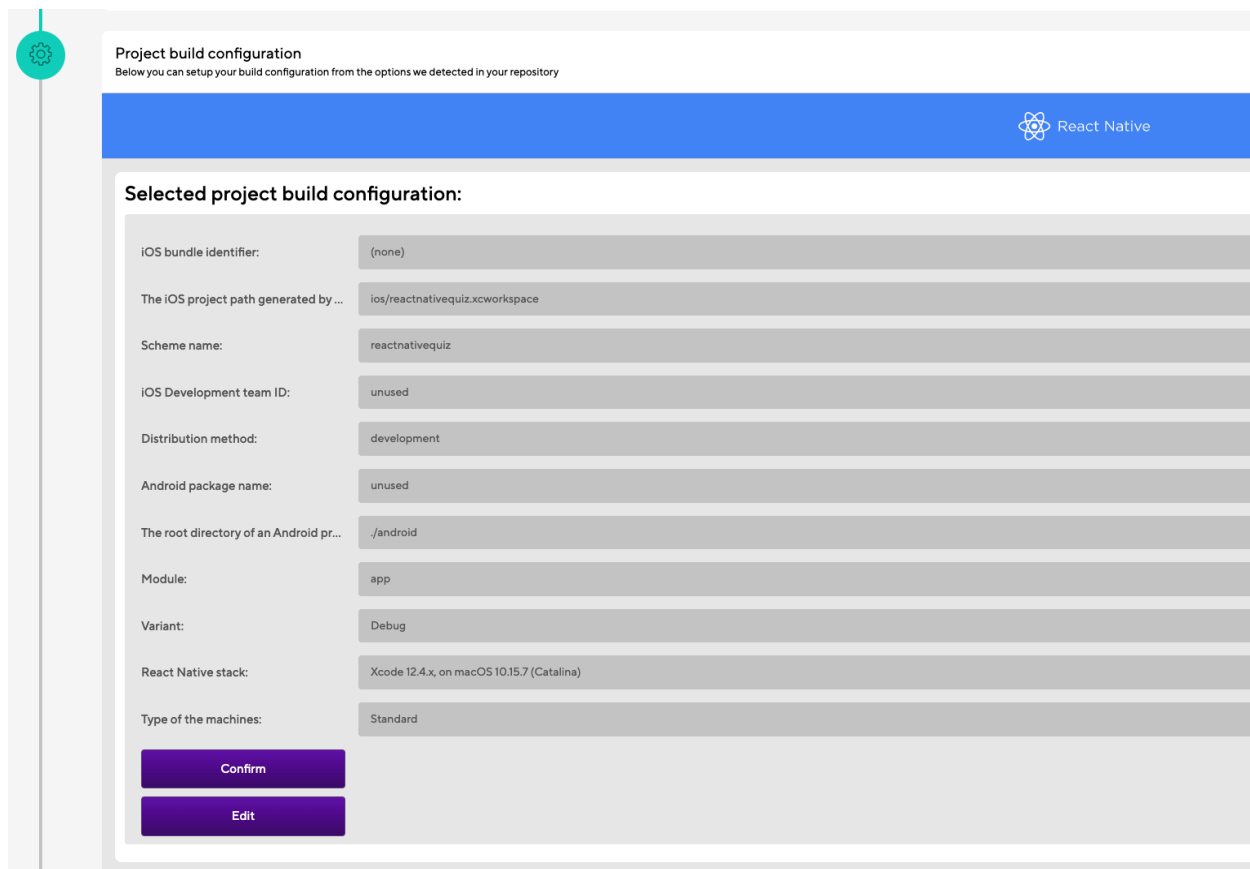
Adding the repo to Bitrise

First, fork the [start repository on Github](#). You'll be working with this code.

Next, open your Bitrise account and create a new Workspace if you haven't already.

Now we can add a new app.

1. Connect your GitHub account, as you'll have to add SSH keys and setup webhooks
2. Select the previously forked Github repository
3. Set the branch to `master`
4. The scanner will recognize your React Native code and prefill what it can
5. For the iOS dev team ID and Android package name you can set any string, we won't be using it during this WS



The screenshot shows the 'Project build configuration' screen in Bitrise. At the top, there's a blue header with the React Native logo and text. Below it, a section titled 'Selected project build configuration:' lists various build settings. At the bottom, there are two buttons: 'Confirm' and 'Edit'.

Field	Value
iOS bundle identifier:	(none)
The iOS project path generated by ...	ios/reactnativequiz.xcworkspace
Scheme name:	reactnativequiz
iOS Development team ID:	unused
Distribution method:	development
Android package name:	unused
The root directory of an Android pr...	./android
Module:	app
Variant:	Debug
React Native stack:	Xcode 12.4.x, on macOS 10.15.7 (Catalina)
Type of the machines:	Standard

6. You can skip adding an icon

7. Make sure to register a webhook, this is required to be able to start Bitrise builds for new PRs and pushes.

Now we need to enable Github checks. Navigate to the app's settings page.

1. First you need to install the Bitrise Checks app for your GitHub repository
2. Then enable the toggle

The screenshot shows the Bitrise web interface. At the top, there's a navigation bar with a search bar and links for 'Getting started', 'Dashboard', 'Integrations', and 'Insights'. Below this, a breadcrumb trail shows 'Dashboard / Workshops / rn-workshop-bitrise'. A 'bitrise' badge is next to the app name. On the right, there are 'Watching' and 'Source' buttons. A horizontal menu contains 'Builds', 'Workflow', 'Add-ons', 'Team', 'Code', and 'Settings' (which is highlighted). A 'Start/Schedule a Build' button is on the far right. The main content area is titled 'App settings' with a note 'Only Admin users can access this page'. It contains four input fields: 'TITLE' (rn-workshop-bitrise), 'REPOSITORY URL' (https://github.com/simonmarton/rn-workshop-bitrise.git), 'DEFAULT BRANCH' (master), and 'YOUR NEXT BUILD NUMBER WILL BE' (2). A 'Save changes' button is at the bottom left of this section. Below the settings is a section titled 'Enable GitHub Checks' with a descriptive text and a toggle switch. Two red arrows point to the toggle switch, which is currently turned on (green checkmark).

RE

Dashboard / Workshops / rn-workshop-bitrise bitrise

Watching Source

Builds Workflow Add-ons Team Code Settings Start/Schedule a Build

APP ICON

RE

App settings
Only Admin users can access this page

TITLE
rn-workshop-bitrise

REPOSITORY URL
https://github.com/simonmarton/rn-workshop-bitrise.git

DEFAULT BRANCH
master

YOUR NEXT BUILD NUMBER WILL BE
2

Save changes

Enable GitHub Checks
Use GitHub Checks instead of classic status updates. To enable this feature you should first [install our Bitrise Checks app to your GitHub repository](#).

Now it's time to set up Bitrise workflows. Open the app's Workflow editor.

Here you can see the default workflow for React Native that Bitrise automatically added for you. But since we already have a `bitrise.yml` in our repository we can use that.

Open the `bitrise.yml` settings and set to Stored in the app repository and hit Update

RE

Dashboard / Workshops / rn-workshop-bitrise / Workflow Editor

Workflows

Code Signing

Secrets

Env Vars

Triggers

Stacks & Machines

<> bitrise.yml

Discard

Saved

Where do you want to store the bitrise.yml file?

Store and manage the bitrise.yml configuration file on either bitrise.io or in your app's repository. Check out the [documentation](#) for the details

**Store on bitrise.io**
The bitrise.yml file is stored and managed on bitrise.io.

**Store in app repository**
The bitrise.yml file is stored in your app's repository and it's versioned and maintained using Git.

Update the bitrise.yml file in your app's repository
The repository already contains a bitrise.yml file. Update the file in the repository with the content of the current one on bitrise.io. Read more in [our documentation](#)

[Copy the content of the current bitrise.yml file to the clipboard](#)

[Download the bitrise.yml file](#)

Update settings

Cancel

We still have to add our Expo credentials. We can do that under the Secrets tab.
Add your login details with `EXPO_USERNAME`, `EXPO_PASSWORD` environment variables.

RE

Dashboard / Workshops / rn-workshop-bitrise / Workflow Editor

Workflows

Code Signing

Secrets

Env Vars

Triggers

Stacks & Machines

<> bitrise.yml

**Your secrets are safe with us**

- Your secrets are not shown in the bitrise.yml
- Your secrets are stored encrypted
- You can prevent exposing secrets on the UI by making them protected
- Note that anyone might be able to do a workaround and log the value of secrets with a pull request, thus we advise **not to expose secrets in PRs**

Secret Environment Variables

\$

EXPO_USERNAME

=



Edit

☐ Replace variables in inputs?

☐ Expose for Pull Requests?

\$

EXPO_PASSWORD

=



Edit

☐ Replace variables in inputs?

☐ Expose for Pull Requests?

Add new

After this we have to configure the Github workflow.
To do this navigate to the repo and open a new pull request. The simplest way to do this is to edit the Readme. Make sure to check the new branch option and then create a new PR.



Commit changes

chore: Edit Readme

Add an optional extended description...

simon.marton@bitrise.io

Choose which email address to associate with this commit

- ☐ Commit directly to the `master` branch.
- ☒ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

simonmarton-patch-2

Propose changes

Cancel

Creating a new PR should kick off a CI build on Bitrise. (Based on the configured trigger map)
It should finish successfully in about a minute.

Add more commits by pushing to the `simonmarton-patch-3` branch on `simonmarton/rn-workshop-bitrise`.



1 successful check

[Hide all checks](#)



Bitrise Successful in 1m — Summary

Required

[Details](#)



This branch has no conflicts with the base branch

Merge pull request



[Convert to draft](#) or view [command line instructions](#).

Now we can enable branch protection rules on Github. With this we can restrict merges to the master branch only if the CI ran successfully. Go to the branch protection rules, add a new rule for the master branch with these settings:

- Require a pull request before merging
- Require status checks to pass before merging
 - Search & add the *Bitrise* check

Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Actions
Environments
Secrets
Pages
Moderation settings

Branch protection rule

Branch name pattern

master

Applies to 1 branch

master

Protect matching branches

☒
Require a pull request before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

☐
Require approvals

When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.

☐
Dismiss stale pull request approvals when new commits are pushed

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐
Require review from Code Owners

Require an approved review in pull requests including files with a designated code owner.

☒
Require status checks to pass before merging

Choose which **status checks** must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

☐
Require branches to be up to date before merging

This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

Q
Search for status checks in the last week for this repository

Status checks that are required.

Bitrise

×

☐
Require conversation resolution before merging

When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that

Now we are able to merge our PR. This should trigger a `publish` workflow.

After the publish is done you should see the app on your phone in the Expo Go app (after you have logged in there too)

