

Creating a GitHub Workflow

Start

GitHub Actions is a popular continuous integration and continuous delivery platform that allows you to automate your build, test, and deployment pipeline. You create workflows that can be triggered when interesting events happen in your GitHub repository, and GitHub will provide the necessary virtual machines to run the workflows.

You can learn more about GitHub Actions at github.com/features/actions, which will give a good overview of the many features included in this platform.

Let's now switch to our dotnetmicroservices GitHub organization and let's use GitHub Actions to create a CI workflow that will trigger any time we push changes to our Play.Identity repository.

In GitHub

1. Browse to the Play.Identity repository
2. Go to Actions
3. Select **Configure** in the .NET action
4. Rename dotnet.yml to cicd.yml
5. Modify cicd.yml to create the NuGet package:

```
name: CICD
```

```
on:
```

```
  push:
```

```
    branches: [ main ]
```

```
jobs:
```

```
  package-and-publish-contracts:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - uses: actions/checkout@v2
```

```
      - name: Setup .NET
```

```
        uses: actions/setup-dotnet@v1
```

```
        with:
```

```
          dotnet-version: 5.0.x
```

```
- name: Pack
  run: dotnet pack src/Play.Identity.Contracts/ --configuration Release -
p:RepositoryUrl=https://github.com/${{ github.repository_owner }}/play.identity -o packages
```

6. Click Start commit and commit changes
7. Go to Actions
8. Check the progress of the job

In Play.Identity repo

9. Pull latest
10. Show new github workflow file

In the next lesson you will learn how to generate the version number needed for your contracts NuGet package.