Generic Packages in CI/CD

Forgejo (and Gitea) provide a **generic package registry** that allows you to publish arbitrary files from CI/CD pipelines — ideal for distributing compiled binaries, configuration files, or any static assets.

Authentication

Forgejo requires **Basic authentication** for the generic packages API. This differs from the standard REST API which accepts `Authorization: token` headers.

```
curl --user "<username>:<PAT>" ...
```

Where `<PAT>` is a Personal Access Token with the `write:package` scope.

Publishing Files

Upload a single file

```
curl -X PUT \
  --user "<username>:<PAT>" \
  --upload-file ./mybinary \
  https://git.example.com/api/packages/<owner>/generic/<repo>/<version>/<filename>
```

Path components:

Component	Description
`<owner>`	Your username or organization name
`<repo>`	Repository name (used as package namespace)
`<version>`	Version string (e.g. `v1.0.0`, `0.2.1-rc1`)
`<filename>`	The actual file to upload

Upload multiple files for the same version

```
curl -X PUT \
  --user "<username>:<PAT>" \
  --upload-file ./linux-binary \
  https://git.example.com/api/packages/<owner>/generic/<repo>/<version>/linux-binary

curl -X PUT \
  --user "<username>:<PAT>" \
  --upload-file ./macos-binary \
  https://git.example.com/api/packages/<owner>/generic/<repo>/<version>/macos-binary
```

Each file is stored independently under the same version path.

Downloading Files

Direct download with authentication

```
curl -fsSL --user "<username>:<PAT>" \
  https://git.example.com/api/packages/<owner>/generic/<repo>/<version>/<filename> \
  --output ./downloaded-binary
```

The `-fsSL` flags:

- `-f`: Fail silently on HTTP errors
- `-s`: Silent mode (no progress meter)
- `-S`: Show errors if silent mode is used
- `-L`: Follow redirects

Without authentication (public packages)

If the package repository is public, you can download without credentials:

```
curl -fsSL \
  https://git.example.com/api/packages/<owner>/generic/<repo>/<version>/<filename> \
  --output ./downloaded-binary
```

GitHub Actions Workflow Example

```
name: Release
on:
  push:
    tags:
      - "v*"

permissions:
  contents: write
  packages: write

jobs:
  release:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Build release binary
        run: cargo build --release

      - name: Upload to generic registry
        env:
          GITEA_TOKEN: ${{ secrets.PACKAGE_TOKEN }}
        run: |
          curl -X PUT \
            --user "${{ github.actor }}:${GITEA_TOKEN}" \
            --upload-file target/release/myapp \
            https://git.example.com/api/packages/${{ github.repository_owner
}}/generic/myapp/${{ github.ref_name }}/myapp
```

Gitea Actions Workflow Example

```
name: Release
on:
  push:
```

```
    tags:
      - "v*"

jobs:
  release:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Build release binary
        run: cargo build --release

      - name: Upload to generic registry
        env:
          GITEA_TOKEN: ${{ secrets.PACKAGE_TOKEN }}
        run: |
          curl -X PUT \
            --user "username:$GITEA_TOKEN" \
            --upload-file target/release/myapp \
            https://git.example.com/api/packages/${{ github.repository_owner
}}/generic/myapp/${{ github.ref_name }}/myapp
```

Troubleshooting

Error	Cause	Solution
`401 Unauthorized`	Missing or invalid PAT	Verify token has `write:package` scope
`403 Forbidden`	Insufficient permissions	Ensure user has write access to the repository
`404 Not Found`	Wrong path or private package	Check owner/repo/version/filename; use auth for private packages
`405 Method Not Allowed`	Wrong HTTP method	Use `PUT`, not `POST`
`500 Internal Server Error`	Content-Type mismatch	Remove `Content-Type: application/json`; let curl auto-detect

Key Differences: Forgejo vs Gitea

Both Forgejo and Gitea support generic packages, but authentication differs:

Endpoint	Gitea Auth	Forgejo Auth
`/api/v1/...` (REST)	`Authorization: token <PAT>`	`Authorization: token <PAT>`
`/api/packages/...` (packages)	`Authorization: token <PAT>`	**Basic auth** (`--user user:token`)

Always use Basic auth for the packages API to ensure compatibility across both platforms.

Revision #8
Created 2026-05-08 21:25:35 UTC by Clive
Updated 2026-05-09 08:38:12 UTC by Clive