テストコード書いてますか？

特にDB操作をするようなテストを書く場合

- テスト実行前にDBにテストデータを投入する
- テスト実行後にDBからテストデータを削除する

といった処理が必要になり面倒です。

今回は、testcontainers というライブラリを利用してテストコードを書く方法を紹介します。

testcontainers とは

testcontainers は、テストのためのコンテナを提供するライブラリです。
テストの一部として実行されるべきコンテナをプログラムで定義し、テスト終了時に
それらのリソースをクリーンアップできます。

https://golang.testcontainers.org/

Testcontainers for Go をインストールする

```
go get github.com/testcontainers/testcontainers-go
```

その他、テストを実行する環境で Docker が必要となります。

testcontainers-go でテストを書く

コンテナ定義

```
func NewTestDatabase(t *testing.T) testcontainers.Container {
    req := testcontainers.ContainerRequest{
                Hostname:     "postgres-server",
                Image:        "postgres:15.4",
                ExposedPorts: []string{"5432/tcp"},
                HostConfigModifier: func(hostConfig *container.HostConfig) {
                        hostConfig.AutoRemove = true
                },
                Env: map[string]string{
                        "POSTGRES_USER":     "user",
                        "POSTGRES_PASSWORD": "password",
                        "POSTGRES_DB":       "testdb",
                },
                HostConfigModifier: func(hostConfig *container.HostConfig) {
                        hostConfig.AutoRemove = true
                },
                Mounts: testcontainers.ContainerMounts{
                        testcontainers.BindMount(testDataPath, "/docker-entrypoint-initdb.d"),
                },
                WaitingFor: wait.ForSQL(nat.Port("5432/tcp"), "pgx", dbURL).WithStartupTimeout(time.Minute * 5),
    }
```

コードは抜粋です

testcontainers-go でテストを書く

コンテナ起動

```
        postgres, err := testcontainers.GenericContainer(
                ctx,
                testcontainers.GenericContainerRequest{
                        ContainerRequest: req,
                        Started:          true,
                },
        )
        if err != nil {
                log.Printf("err: %v", err)
        }
        t.Cleanup(func() {
                require.NoError(t, postgres.Terminate(ctx))
        })
```

コードは抜粋です

testcontainers-go を利用したテストを実行する

余談

このスライドは、marp というツールを利用して作成しています。
marpは、Markdownでスライドを作成できるツールです。