✅ Feature Toggle Management Guide

For Trunk-Based Development – Java & .NET Teams

# 🎯 Why Feature Toggles?

When working with trunk-based development, all changes go into the main branch. Feature toggles help hide incomplete or risky code, so we can deploy frequently without impacting users.

# 🧩 Types of Feature Toggles

|  |  |  |
| --- | --- | --- |
| Toggle Type | Purpose | Typical Duration |
| Release Toggle | Hide unfinished features | Short-term |
| Experiment Toggle | A/B testing or gradual rollout | Short to medium |
| Ops Toggle | Control runtime behavior in prod | Long-term |
| Permission Toggle | Enable features for selected users | Long-term |

# 🔁 Simple Toggle Lifecycle

1. Create the toggle with a clear name and add to code.  
2. Set default value (false) in config.  
3. Test with toggle ON and OFF.  
4. Enable the toggle in production when ready.  
5. Remove the toggle and related code when it's no longer needed.

# 🛠️ Java Implementation Guide

Preferred Pattern:  
- Default value in application.properties  
- Override using environment variables

application.properties:  
feature.newLoginUI=false

Environment override:  
export FEATURE\_NEWLOGINUI=true

Code sample:  
  
@Value("${feature.newLoginUI:false}")  
private boolean newLoginUIEnabled;  
  
if (newLoginUIEnabled) {  
 // new feature code  
} else {  
 // old behavior  
}

@ConditionalOnProperty(name = "feature.newLoginUI", havingValue = "true")  
@Bean  
public NewLoginUIService newLoginUIService() {  
 return new NewLoginUIService();  
}

# 🛠️ .NET Implementation Guide

Use Microsoft.FeatureManagement NuGet package.

appsettings.json:  
{  
 "FeatureManagement": {  
 "NewLoginUI": false  
 }  
}

C# Code:  
if (await \_featureManager.IsEnabledAsync("NewLoginUI")) {  
 // new feature code  
}

Startup.cs:  
services.AddFeatureManagement();  
  
Environment override:  
FeatureManagement\_\_NewLoginUI=true

# 📋 Naming Convention

Use lowercase, descriptive names with optional prefixes:  
- feature.newLoginUI  
- ops.enableRetryMechanism  
- experiment.abTestCheckout

# 📑 Toggle Registry Example

| Name | Type | Owner | Added On | Remove By | Description |  
|--------------------|----------|-------|-----------|-----------|-------------------------|  
| feature.newLoginUI | Release | Ravi | 10-Jun-25 | 01-Jul-25 | Toggle for new login UI |

# 🧪 Testing Tips

- Test with toggle ON and OFF  
- Include toggle state in test cases  
- Use mocks/stubs for toggle values in unit tests

# ⏳ Clean-Up Guidelines

- Set a removal date when you create a toggle  
- Review toggles monthly (team checklist)  
- Delete unused toggles and related code after rollout

# 🔍 Quick Reminders

- ✅ Toggle defaults in config (not code)  
- ✅ Use env vars for environment overrides  
- ✅ Avoid long-living toggles unless needed (e.g., ops/permission)  
- ❌ Don’t skip tests with toggles off  
- ❌ Don’t forget to clean up