// Test the feature suggestion flow

Orion.ai.suggester.fetchTrendingFeatures()

.then(features => Orion.ui.showFeatureSuggestions(features))

.catch(console.error);

// Test feature discovery

Orion.ai.suggester.fetchTrendingFeatures()

.then(features => {

console.log("Discovered features:", features);

Orion.ui.showFeatureSuggestions(features.slice(0, 3)); // Show top 3

})

.catch(err => console.error("Failed:", err));

// Create a test version

Orion.version.createVersion({

description: "Test version update"

});

// Check version history

console.log("Version history:", Orion.version.getVersionHistory());

// Test code generation

Orion.ai.generateFeatureModule({

name: "test-module",

description: "A test AI feature"

}).then(module => {

console.log("Generated module:", module.code);

// Test in sandbox

Orion.update.testInSandbox(module.code)

.then(result => console.log("Sandbox result:", result));

});

// Simulate a complete update flow

const testFeature = {

name: "sentiment-analysis",

description: "Adds sentiment detection to messages",

html\_url: "https://github.com/example/sentiment-ai"

};

// 1. Discover feature

Orion.ui.showFeatureSuggestions([testFeature]);

// 2. Click "Request Integration" (manual)

// 3. Approve the update in UI

// 4. Verify version changed:

console.log("Current version:", Orion.version.getCurrentVersion());

document.addEventListener('DOMContentLoaded', () => {

// Run tests when Orion is ready

setTimeout(() => {

console.log("Running self-tests...");

// Test feature suggester

Orion.ai.suggester.fetchTrendingFeatures()

.then(features => {

if (features.length === 0) throw new Error("No features loaded");

console.log("✅ Feature suggester working");

});

// Test versioning

const v = Orion.version.createVersion({description: "Self-test"});

if (!v) throw new Error("Versioning failed");

console.log("✅ Version control working");

}, 1000);

});