

Lab 107: Comment fonctionne GitHub Copilot : contexte, modèles, agents et flux de données

Level: L50 Parcours: copilote Durée: ~25 min Coût: Gratuit

info "Conception gratuite d'abord" Terminez l'exercice avec Copilot Free ou la simulation incluse. Les fonctions payantes sont facultatives et clairement indiquées.

tip "Impact sur les AI Credits" Faible pour l'exercice guidé ; moyen/élevé pour les exécutions agentiques ou répétées. Préférez la sélection automatique du modèle, planifiez avant d'agir, gardez un contexte ciblé et évitez les sessions parallèles inutiles.

Ce que vous apprendrez

Décrire le cycle d'une requête Copilot Séparer sélection du contexte et du modèle

Distinguer compléments, chat et agents Expliquer frontières de données, rétention et crédits IA

Alignement GH-300

GH-300 foundation · Data and architecture. Cette correspondance suit l'instantané GH-300 du 07/08/2026. Le plan officiel peut conserver des termes anciens comme « Copilot Edits » ; ce lab enseigne l'expérience actuelle et signale la différence.

Carte conceptuelle

[diagram or code sample]

Exercice pratique

Étape 1: Tracer une requête

Suivez signaux de l'éditeur, références, instructions, politiques, routage du modèle, génération et vérification humaine.

Étape 2: Comparer les expériences

Les compléments prédisent dans le flux d'édition ; le chat répond ; les agents planifient et utilisent des outils en plusieurs étapes.

Étape 3: Cartographier données et facturation

Documentez contrôle du dépôt, contexte transmis, rétention, exclusions et interactions consommant des crédits IA.

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

Télécharger le PDF généré Télécharger le ZIP généré

Supporting file: copilot_glossary.csv

term,en,pt,fr context,Context,Contexto,Contexte agent,Agent,Agente,Agent

tool,Tool,Ferramenta,Outil

Supporting file: data_flow_handout.md

Copilot data-flow handout

The accompanying SVG traces user context through policy, model selection, tool execution, and human-reviewed output. Keep repository instructions, retrieved content, tool output, and generated changes inside explicit trust boundaries.