Block suggestions matching public code

Individual User Settings

To block GitHub Copilot suggestions matching public code, go to your profile settings, select "Copilot" in the sidebar, choose "Block" under "Suggestions matching public code," and save your changes. This prevents Copilot from suggesting code that closely matches public code on GitHub.

Organization-Level Settings

For organizations, access your organization settings, navigate to "Copilot" under "Code, planning, and automation," and select "Blocked" under "Suggestions matching public code" to apply this setting for all members. This helps prevent copyright issues and maintain code originality.

Additional Information

GitHub Copilot can detect and block suggestions matching public code, reducing legal risks and ensuring compliance with licensing requirements. This feature can be enabled by enterprise administrators across all organizations within an enterprise.

Exclude specified files from Copilot

Repository-Level Exclusions

To exclude specific files in a repository, go to repository settings, navigate to "Copilot" under "Code & automation," and enter file paths to be ignored in the "Paths to exclude in this repository" text box using fnmatch pattern matching notation.

Organization-Level Exclusions

For organization-wide exclusions, access organization settings, select "Copilot," and specify repositories and paths to exclude using the format `REPOSITORY-REFERENCE: - "/PATH/TO/DIRECTORY/OR/FILE"` in the "Repositories and paths to exclude" box.

Using .copilotignore File

In VSCode, use a `.copilotignore` file to list patterns of files and directories to exclude from Copilot suggestions, similar to a `.gitignore` file. This helps prevent sensitive or irrelevant files from being used for code suggestions.

Organization-wide policy management

Setting Policies

As an organization owner, manage GitHub Copilot settings by navigating to "Copilot" under organization settings, and enabling or disabling features like Copilot Chat and public code suggestions under "Policies."

Granting and Managing Access

To enable Copilot for members, go to "Access" in organization settings, choose to enable for all members or selected users, and manage requests from members. This helps control who can use Copilot within the organization.

Additional Controls

Exclude certain files from Copilot suggestions by configuring paths in "Content exclusion," and monitor usage with audit logs to ensure compliance with organizational policies.

Reviewing Audit Logs for GitHub Copilot

Accessing Audit Logs

Navigate to organization settings, select "Logs" under "Archives" in the sidebar, and click "Audit log" to access audit logs for GitHub Copilot Business subscription.

Searching Audit Log Events

Search audit logs using qualifiers like `action:copilot` or `actor:username` to filter events by action, operation type, repository, or user. This helps in tracking specific activities related to Copilot.

Copilot-Specific Events

Audit logs track actions such as changes to Copilot settings, seat assignments, and content exclusions. Regularly review these logs to maintain transparency and control over Copilot usage within your organization.

Copilot Chat skills

Core Skills

GitHub Copilot Chat can answer coding questions, write code snippets, fix and improve code, explain code, generate tests, and set up projects. These interactive features assist developers with various coding tasks through a chat interface available on GitHub.com, supported IDEs, and GitHub Mobile.

Copilot pull request summaries

Key Features of Copilot Pull Request Summaries

Copilot can automatically generate summaries for pull requests, including a high-level overview and a detailed outline of modifications. This helps reviewers quickly understand the changes without examining each file individually.

How It Works

Copilot analyzes code diffs to generate summaries using a large language model. Summaries can be initiated when creating or updating a pull request, and feedback mechanisms help improve the feature over time.

Limitations

The feature currently supports English and may have limitations with very large pull requests or complex changes. It should be used as a supplement to human reviews to ensure comprehensive understanding.

Copilot knowledge bases

Creating and Managing Knowledge Bases

Admins can create and manage knowledge bases by selecting repositories containing relevant documentation. These knowledge bases provide contextualized assistance by leveraging internal documentation for Copilot Chat responses.

Usage and Benefits

Knowledge bases ensure Copilot provides accurate, context-aware answers, improving productivity and collaboration by providing consistent information to all team members. Regular updates and feedback loops help maintain the quality of responses.

Zero data retention for code snippets and usage telemetry

Code Snippets and Prompts

GitHub Copilot processes code snippets in real-time without storing them permanently. Users can control whether their code snippets are used for product improvements by adjusting settings in Copilot.

Usage Telemetry

Telemetry data includes user engagement metrics and is collected with pseudonymous identifiers. Organizations can manage telemetry settings to comply with data privacy standards and disable telemetry collection entirely if needed.

Integration with security tools

Integration with Security Tools

GitHub Copilot integrates with security tools like Microsoft Copilot for Security and GitHub Advanced Security to identify and fix vulnerabilities, enhancing security practices and streamlining development workflows.

Benefits and Features

These integrations help developers write secure code, streamline workflows by providing security alerts within existing processes, and ensure data privacy and control. Organizations can manage settings to align with internal policies and regulatory requirements.