

Version 1.79 (/updates) is now available! Read about the new features and fixes from May.



Topics

In this article

 (<https://vscode.dev/github/microsoft/vscode-docs/blob/main/api/extension-capabilities/overview.md>)

Extensions Capabilities Overview

Visual Studio Code offers many ways for extensions to extend its capabilities. It can sometimes be hard to find the right Contribution Points (/api/references/contribution-points) and VS Code API (/api/references/vscode-api) to use. This topic splits extension capabilities into a few categories. Each category describes:

- Some functionalities your extension could use
- Links to more detailed topics for using these functionalities
- A few extension ideas

However, we also impose restrictions upon extensions to ensure the stability and performance of VS Code. For example, extensions cannot access the DOM of VS Code UI.

Common Capabilities

Common Capabilities (/common-capabilities) are core pieces of functionality that you can use in any extension.

Some of these capabilities include:

- Registering commands, configurations, keybindings, or context menu items.
- Storing workspace or global data.
- Displaying notification messages.
- Using Quick Pick to collect user input.
- Open the system file picker to let users select files or folders.
- Use the Progress API to indicate long-running operations.

Theming

Тематизация (/theming) определяет внешний вид VS Code, как цвета исходного кода в редакторе, так и цвета пользовательского интерфейса VS Code. Если вы когда-либо хотели создать впечатление, что вы кодируете матрицу, используя различные оттенки зеленого в VS Code, или просто хотели создать максимально минималистичное рабочее пространство в оттенках серого, то темы для вас.

Идеи расширения

- Измените цвета вашего исходного кода.

- Измените цвета пользовательского интерфейса VS Code.
- Перенесите существующую тему TextMate в VS Code.
- Добавьте пользовательские значки файлов.

Возможности декларативного языка

Функции декларативного языка (</api/language-extensions/overview#declarative-language-features>) добавляют базовую поддержку редактирования текста для языка программирования, такую как сопоставление скобок, автоматический отступ и подсветка синтаксиса. Это делается декларативно, без написания какого-либо кода. Более продвинутые языковые функции, такие как IntelliSense или отладка, см. в разделе Возможности языка программирования.

Идеи расширения

- Объедините общие фрагменты JavaScript в расширение.
- Расскажите VS Code о новом языке программирования.
- Добавьте или замените грамматику для языка программирования.
- Расширьте существующую грамматику с помощью грамматических инъекций.
- Перенесите существующую грамматику TextMate в VS Code.

Возможности языка программирования

Функции языка программирования (</api/language-extensions/overview#programmatic-language-features>) добавляют расширенную поддержку языка программирования, такую как наведение курсора мыши, переход к определению, диагностические ошибки, IntelliSense и CodeLens. Эти языковые возможности доступны через `vscode.languages.*` (</api/references/vscode-api#languages>) API. Расширение может либо использовать эти API напрямую, либо написать языковой сервер и адаптировать его к VS Code, используя языковую серверную библиотеку (<https://github.com/microsoft/vscode-languageserver-node>) VS Code.

Хотя мы предоставляем список языковых возможностей (</api/language-extensions/programmatic-language-features>) и их предполагаемого использования, ничто не мешает вам творчески использовать эти API. Например, CodeLens и Hovers - отличный способ встроенного представления дополнительной информации, в то время как диагностические ошибки можно использовать для выделения орфографических ошибок или ошибок стиля кода.

Идеи расширения

- Добавьте элементы наведения, которые показывают пример использования API.
- Сообщайте об ошибках правописания или линтера в исходном коде с помощью диагностики.
- Зарегистрируйте новый форматировщик кода для HTML.
- Предоставляет богатый контекстно-зависимый IntelliSense.
- Добавьте поддержку сворачивания, "панировочных сухарей" и outline для языка.

Расширения Workbench

Расширения Workbench (`./extending-workbench`) расширяют пользовательский интерфейс VS Code Workbench. Добавьте новые действия по щелчку правой кнопкой мыши в проводник файлов или даже создайте пользовательский проводник, используя `TreeView` (`/api/extension-guides/tree-view`) API VS Code. И если вашему расширению требуется полностью настраиваемый пользовательский интерфейс, используйте `Webview` API (`/api/extension-guides/webview`) для создания собственного предварительного просмотра документа или пользовательского интерфейса, используя стандартные HTML, CSS и JavaScript.

Идеи расширения

- Добавьте пользовательские действия контекстного меню в проводник файлов.
- Создайте новый интерактивный вид дерева на боковой панели.
- Определите новый вид панели действий.
- Отображать новую информацию в строке состояния.
- Визуализируйте пользовательский контент с помощью `webview` API.
- Добавьте поставщиков системы управления версиями.

Отладка

Вы можете воспользоваться преимуществами функций отладки (`/docs/editor/debugging`) VS Code, написав расширения отладчика (`/api/extension-guides/debugger-extension`), которые подключают интерфейс отладки VS Code к определенному отладчику или среде выполнения.

Идеи расширения

- Подключите пользовательский интерфейс отладки VS Code к отладчику или среде выполнения, предоставив реализацию адаптера отладки (<https://microsoft.github.io/debug-adapter-protocol/implementors/adapters/>).
- Укажите языки, поддерживаемые расширением debugger.
- Предоставьте расширенную информацию IntelliSense и наведите указатель мыши на атрибуты конфигурации отладки, используемые отладчиком.
- Предоставьте фрагменты конфигурации отладки.

С другой стороны, VS Code также предлагает набор `Debug Extension API` (`/api/references/vscode-api#debug`), с помощью которого вы можете реализовать функциональность, связанную с отладкой, поверх любого отладчика VS Code, чтобы автоматизировать процесс отладки пользователей.

Идеи расширения

- Запускайте сеансы отладки на основе динамически созданных конфигураций отладки.
- Отслеживайте жизненный цикл сеансов отладки.
- Создавайте точки останова программно и управляйте ими.

Рекомендации по UX

Чтобы помочь вашему расширению без проблем вписаться в пользовательский интерфейс VS Code, ознакомьтесь с Рекомендациями по UX (</api/ux-guidelines/overview>), где вы узнаете о лучших методах создания пользовательского интерфейса расширения и соглашениях по следованию предпочитаемым рабочим процессам VS Code.

Ограничения

Существуют определенные ограничения, которые мы налагаем на расширения. Вот ограничения и их назначение.

Нет доступа к DOM

Расширения не имеют доступа к DOM пользовательского интерфейса VS Code. Вы **не** можете написать расширение, которое применяет пользовательский CSS к VS Code или добавляет HTML-элемент в пользовательский интерфейс VS Code.

В VS Code мы постоянно пытаемся оптимизировать использование базовых веб-технологий, чтобы обеспечить всегда доступный редактор с высокой адаптивностью, и мы будем продолжать настраивать использование DOM по мере развития этих технологий и нашего продукта. Чтобы гарантировать, что расширения не могут влиять на стабильность и производительность VS Code и что мы можем продолжать улучшать DOM VS Code, не нарушая работу существующих расширений, мы запускаем расширения в процессе узла (</api/advanced-topics/extension-host>) расширения и предотвращаем прямой доступ к DOM.

Нет пользовательских таблиц стилей

Пользовательская таблица стилей, предоставляемая пользователями или расширениями, будет работать против структуры DOM и имен классов. Они не документированы, поскольку мы считаем их внутренними. Для разработки, рефакторинга или улучшения VS Code нам нужна свобода вносить изменения в пользовательский интерфейс. Любое изменение в DOM может нарушить работу существующих пользовательских таблиц стилей, что приведет к разочарованию поставщиков таблиц стилей и ухудшению работы пользователей из-за сбоев пользовательского интерфейса, возникающих из-за неработающей таблицы стилей.



Вместо этого VS Code стремится предоставить хорошо разработанный extension API, поддерживающий настройки пользовательского интерфейса. API документирован, поставляется с инструментами и примерами и поддерживается стабильным во всех предстоящих выпусках VS Code.

Была ли эта документация полезной?

☐ Да

☐ Нет

6/8/2023

-  [Subscribe\(/feed.xml\)](/feed.xml)
-  [Ask questions\(https://stackoverflow.com/questions/tagged/vscode\)](https://stackoverflow.com/questions/tagged/vscode)
-  [Follow @code\(https://go.microsoft.com/fwlink/?LinkID=533687\)](https://go.microsoft.com/fwlink/?LinkID=533687)
-  [Request features\(https://go.microsoft.com/fwlink/?LinkID=533482\)](https://go.microsoft.com/fwlink/?LinkID=533482)
-  [Report issues\(https://www.github.com/Microsoft/vscode/issues\)](https://www.github.com/Microsoft/vscode/issues)
-  [Watch videos\(https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w\)](https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w)

Привет из Сиэтла. Следуйте @code (<https://go.microsoft.com/fwlink/?LinkID=533687>)

Звезда 147,349

Поддержка (<https://support.serviceshub.microsoft.com/supportforbusiness/create?sapId=d66407ed-3967-b000-4cfb-2c318cad363d>)

Конфиденциальность (<https://go.microsoft.com/fwlink/?LinkId=521839>)

Условия использования (<https://www.microsoft.com/legal/terms-of-use>) Лицензия (/License)

 (<https://www.microsoft.com>)

© 2023 Microsoft