

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение  
высшего образования**

**“Санкт-Петербургский национальный исследовательский университет  
информационных технологий механики и оптики”**

**Мегафакультет: трансляционных информационных технологий**

**Факультет: информационных технологий и программирования**

**Лабораторная работа №1**

**По дисциплине: “Инструментальные средства разработки ПО”**

**Тема: “GIT”**

**Выполнил студент группы №М3119:**

*Калинин Михаил Владимирович*

**Работу проверила:**

*Алексеева Милена Константиновна*

**САНКТ-ПЕТЕРБУРГ**

**2025**

Архитектура VS Code построена на модульной системе расширений, что позволяет легко добавлять новый функционал без изменения ядра IDE. Основные компоненты архитектуры:

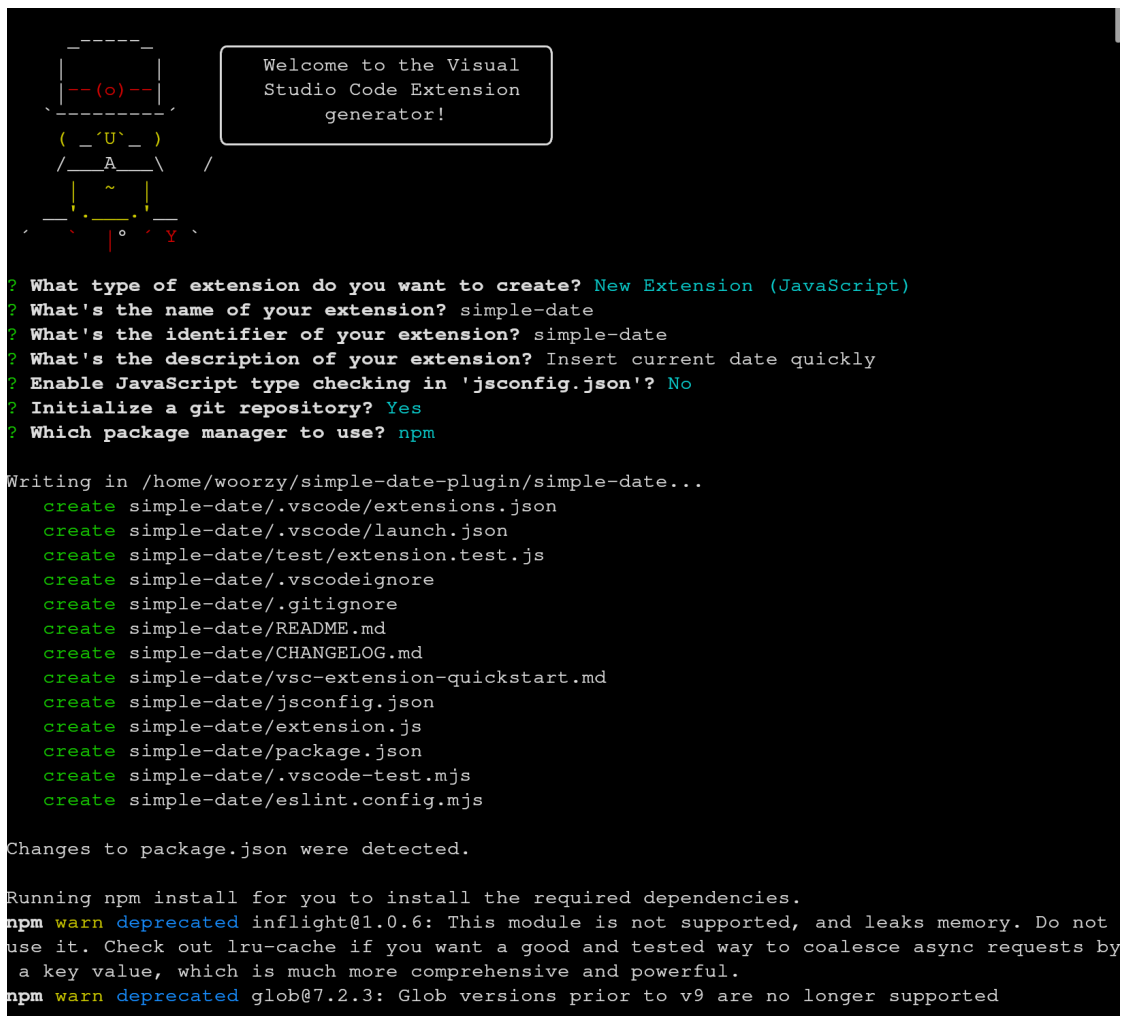
1. Extension Host Process - отдельный процесс для выполнения кода расширений
2. Extension API - богатый API для взаимодействия с редактором, файловой системой, UI
3. Activation Events - система "ленивой" загрузки расширений только когда они нужны
4. Contribution Model - модель объявления возможностей расширения через package.json

Расширения VS Code состоят из:

- package.json - манифест с метаданными и объявлением функциональности
- Основной файл (extension.js) - код расширения, экспортирующий функции activate/deactivate
- Node.js модули - дополнительные зависимости

Процесс разработки: создание манифеста → реализация логики → тестирование в Extension Development Host → публикация.

1. Создание плагина через генератор вскода на JavaScript:



```

  ____
  |  (o)  |
  |_____|
  |  'U'  |
  |_____|
  |  A    |
  |_____|
  |  ~    |
  |_____|
  |  o    |
  |_____|
  |  Y    |
  |_____|

Welcome to the Visual
Studio Code Extension
generator!

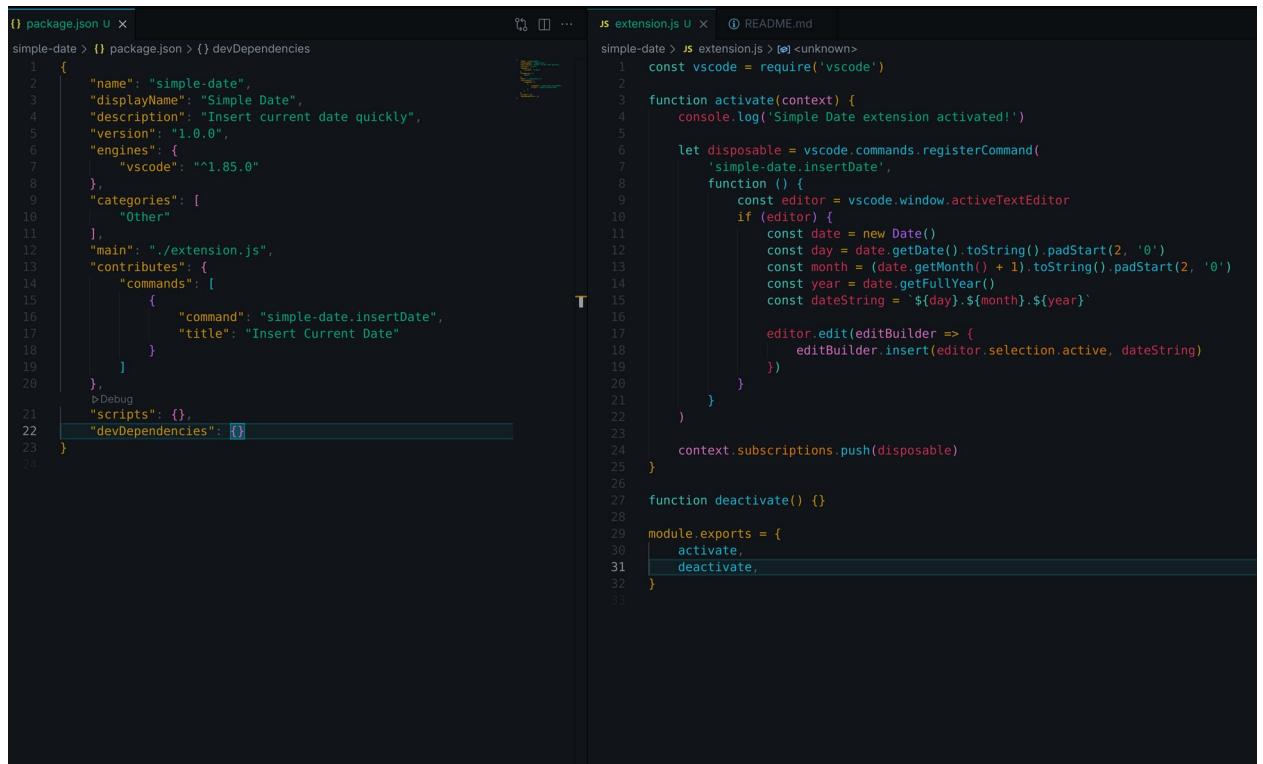
? What type of extension do you want to create? New Extension (JavaScript)
? What's the name of your extension? simple-date
? What's the identifier of your extension? simple-date
? What's the description of your extension? Insert current date quickly
? Enable JavaScript type checking in 'jsconfig.json'? No
? Initialize a git repository? Yes
? Which package manager to use? npm

Writing in /home/woorzy/simple-date-plugin/simple-date...
  create simple-date/.vscode/extensions.json
  create simple-date/.vscode/launch.json
  create simple-date/test/extension.test.js
  create simple-date/.vscodeignore
  create simple-date/.gitignore
  create simple-date/README.md
  create simple-date/CHANGELOG.md
  create simple-date/vsc-extension-quickstart.md
  create simple-date/jsconfig.json
  create simple-date/extension.js
  create simple-date/package.json
  create simple-date/.vscode-test.mjs
  create simple-date/eslint.config.mjs

Changes to package.json were detected.

Running npm install for you to install the required dependencies.
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not
use it. Check out lru-cache if you want a good and tested way to coalesce async requests by
a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
```

## 2. Создание базовой структуры плагина:



```
1 package.json x
2 {
3   "name": "simple-date",
4   "displayName": "Simple Date",
5   "description": "Insert current date quickly",
6   "version": "1.0.0",
7   "engines": {
8     "vscode": "^1.85.0"
9   },
10  "categories": [
11    "Other"
12  ],
13  "main": "./extension.js",
14  "contributes": {
15    "commands": [
16      {
17        "command": "simple-date.insertDate",
18        "title": "Insert Current Date"
19      }
20    ]
21  },
22  "scripts": {},
23  "devDependencies": {}
24 }

1 extension.js x
2 const vscode = require('vscode')
3
4 function activate(context) {
5   console.log('Simple Date extension activated!')
6
7   let disposable = vscode.commands.registerCommand(
8     'simple-date.insertDate',
9     function () {
10       const editor = vscode.window.activeTextEditor
11       if (editor) {
12         const date = new Date()
13         const day = date.getDate().toString().padStart(2, '0')
14         const month = (date.getMonth() + 1).toString().padStart(2, '0')
15         const year = date.getFullYear()
16         const dateString = `${day}.${month}.${year}`
17
18         editor.edit(editBuilder => {
19           editBuilder.insert(editor.selection.active, dateString)
20         })
21       }
22     }
23   )
24
25   context.subscriptions.push(disposable)
26 }
27
28 function deactivate() {}
29
30 module.exports = {
31   activate,
32   deactivate
33 }
```

## 3. Тестирование плагина: запускаем режим отладки, открывается окно extension development host ctrl+shift+p: command palette с режимом «Insert Current Date»

