



# Celerik Scaffold



## How to install?

The screenshot shows the Visual Studio Marketplace page for the 'Celerik Scaffold' extension. The page header includes the extension name, author (Celerik), 17 installs, a 0-star rating, and a 'Free' price tag. A prominent green 'Install' button is visible, along with a link for 'Trouble installing?'. Below the header, there are tabs for 'Overview', 'Version History', 'Q & A', and 'Rating & Review'. The main content area is divided into sections: 'Celerik Scaffolding' (describing it as an extension to improve and speed up file creation), 'Install' (instructing to install via the Extension Marketplace), and 'Usage' (providing a list of steps: 1. Install the extension, 2. Right click on a folder of the project, 3. Select one of the local or remote templates in the list, 4. Replace the variables found in the config.json file with the name you want to assign to it at the time of file creation, 5. Click on generate, 6. The files will be created following the path found inside the template, that is why each template must have the folder structure where you want them to be created, for example src/components). A right-hand sidebar contains 'Categories' (Snippets), 'Works with' (Universal), 'Resources' (Issues, Repository, Homepage, License, Changelog, Download Extension), 'Project Details' (with links to the extension, pull requests, and issues), and 'More Info' (Version: 1.1.0, Released on: 17/11/2022, 10:07:08, Last updated: 26/7/2023, 16:10:47).

This screenshot shows the 'Celerik Scaffold' extension interface within the Visual Studio Code environment. The extension is titled 'Celerik Scaffold v1.1.0' and is described as an 'Extension for VSCode that helps to automate code generation through scaffolding'. It is currently installed and enabled globally. The interface includes tabs for 'DETAILS', 'FEATURE CONTRIBUTIONS', 'CHANGELOG', and 'RUNTIME STATUS'. The 'Celerik Scaffolding' section provides a brief description and installation instructions. The 'Usage' section lists the steps for using the extension, which are identical to those shown in the marketplace screenshot. A 'More Info' table is also present, detailing the extension's publication and update history.

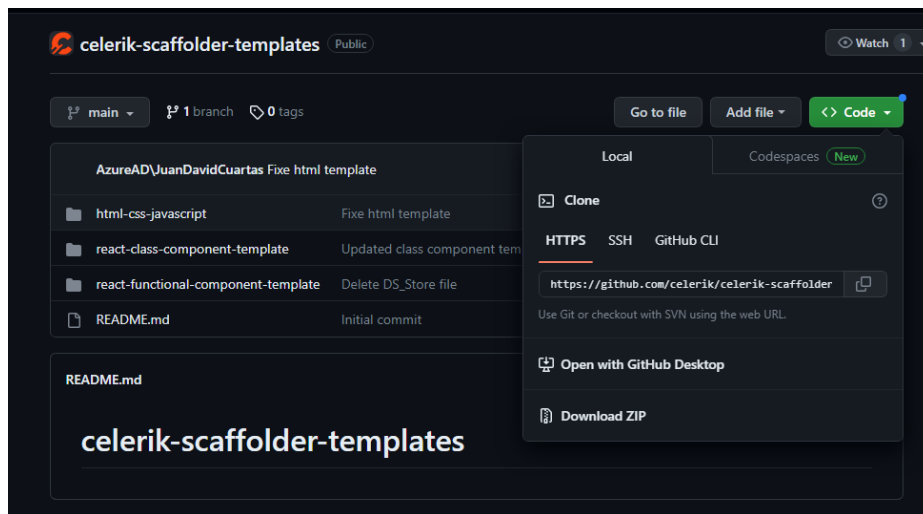
Published	2022-11-17, 10:07:08
Last released	2023-7-26, 16:10:47
Last updated	2023-7-26, 16:10:47
Identifier	celerik.celerik-scaffolder

## Template system

## Locale

```
└─ scaffolding
  └─ react-native-screen-component
    ├── {{componentNameKebab}}
    └─ config.json
  └─ react-native-template
```

## Remote



The screenshot shows the GitHub repository page for 'celerik-scaffolder-templates'. The repository is public and has 1 branch and 0 tags. The file list includes 'html-css-javascript', 'react-class-component-template', 'react-functional-component-template', and 'README.md'. A 'Code' dropdown menu is open, showing options for cloning the repository via HTTPS, SSH, or GitHub CLI, opening it with GitHub Desktop, or downloading it as a ZIP file. The URL for cloning is 'https://github.com/celerik/celerik-scaffolder'.

## Config.json file

```
},  
"isRelative": true  
}
```

```
lding > codepen-class-component-template > {} config.json > ...  
{  
  "variables": [  
    "{{var1}}",  
    {  
      "name": "{{var2}}",  
      "help": "This variable is used to set the var2"  
    },  
    "{{var3}}"  
  ],  
  "expressions": {  
    "{{var1CamelCase}}": {  
      "case": "camel",  
      "variable": "{{var1}}"  
    },  
    "{{var1UpperWithSpaceCase}}": {  
      "case": "upperWithSpace",  
      "variable": "{{var1}}"  
    }  
  },  
  "isRelative": true  
}
```

```
<Box {...boxProps}>  
  {t('Original var1: testVar Example')}  
  <p>Camel: testVarExample</p>  
  <p>kebab: test-var-example</p>  
  <p>lower: testvar example</p>  
  <p>lowerWithSpace: test var example</p>  
  <p>Pascal: TestVarExample</p>  
  <p>snake: test_var_example</p>  
  <p>upper: TESTVAR EXAMPLE</p>  
  <p>upperKebab: TEST-VAR-EXAMPLE</p>  
  <p>upperSnake: TEST_VAR_EXAMPLE</p>  
  <p>upperWithSpace: TEST VAR EXAMPLE</p>  
</Box>
```



Let's get **started.**