Sure! Here's a table of commonly used **React action fields (event handlers)** organized by category:

| **Category** | **Event Handler** | **Description** |
| --- | --- | --- |
| **Mouse Events** | onClick | Fires when an element is clicked |
|  | onDoubleClick | Fires when an element is double-clicked |
|  | onMouseDown | Fires when mouse button is pressed |
|  | onMouseUp | Fires when mouse button is released |
|  | onMouseEnter | Fires when mouse enters an element |
|  | onMouseLeave | Fires when mouse leaves an element |
|  | onMouseMove | Fires when mouse moves inside an element |
|  | onMouseOver | Fires when mouse hovers over an element |
|  | onMouseOut | Fires when mouse moves out of an element |
| **Keyboard Events** | onKeyDown | Fires when a key is pressed |
|  | onKeyUp | Fires when a key is released |
|  | onKeyPress | (Deprecated) Fires when a key is pressed |
| **Form Events** | onChange | Fires when form field value changes |
|  | onInput | Fires when user input is received |
|  | onSubmit | Fires when form is submitted |
|  | onReset | Fires when form is reset |
|  | onInvalid | Fires when a form field is invalid |
| **Focus Events** | onFocus | Fires when an element gains focus |
|  | onBlur | Fires when an element loses focus |
| **Touch Events** | onTouchStart | Touch starts |
|  | onTouchMove | Touch moves |
|  | onTouchEnd | Touch ends |
|  | onTouchCancel | Touch is canceled |
| **Pointer Events** | onPointerDown | Pointer (mouse/stylus/touch) is pressed |
|  | onPointerMove | Pointer moves |
|  | onPointerUp | Pointer is released |
|  | onPointerEnter | Pointer enters an element |
|  | onPointerLeave | Pointer leaves an element |
|  | onPointerOver | Pointer is over an element |
|  | onPointerOut | Pointer leaves the boundaries |
|  | onPointerCancel | Pointer action is canceled |
| **Drag Events** | onDrag | Element is being dragged |
|  | onDragStart | Dragging starts |
|  | onDragEnd | Dragging ends |
|  | onDragEnter | Drag enters a valid drop target |
|  | onDragLeave | Drag leaves a valid drop target |
|  | onDragOver | Drag is over a valid drop target |
|  | onDrop | Dragged element is dropped |
| **Clipboard Events** | onCopy | Fires when content is copied |
|  | onCut | Fires when content is cut |
|  | onPaste | Fires when content is pasted |
| **Misc Events** | onScroll | Fires on scroll |
|  | onWheel | Fires on mouse wheel action |
|  | onLoad | Fires when resource finishes loading |
|  | onError | Fires on a resource loading error |
|  | onAnimationStart | CSS animation starts |
|  | onAnimationEnd | CSS animation ends |
|  | onAnimationIteration | CSS animation repeats |
|  | onTransitionEnd | CSS transition ends |

Let me know if you'd like this exported to a file like CSV or Markdown!

Absolutely! Here's a **filtered table** of the **most commonly used React event handlers** — the ones you'll run into most often in real-world apps:

| **Category** | **Event Handler** | **Description** |
| --- | --- | --- |
| **Mouse** | onClick | Fires when an element is clicked |
|  | onDoubleClick | Fires when element is double-clicked |
| **Keyboard** | onKeyDown | Fires when a key is pressed |
|  | onKeyUp | Fires when a key is released |
| **Form** | onChange | Fires when form field value changes |
|  | onSubmit | Fires when form is submitted |
|  | onInput | Fires on user input |
| **Focus** | onFocus | Element gains focus |
|  | onBlur | Element loses focus |
| **Clipboard** | onCopy | Content is copied |
|  | onPaste | Content is pasted |
| **Misc** | onScroll | Fires on scroll |
|  | onLoad | Resource finishes loading |
|  | onError | Resource loading fails |

These are the ones you'd typically use when working with buttons, forms, inputs, modals, dynamic content, or custom keyboard interactions.

Want this as a cheat sheet or code snippet reference too?

**🛠️ Git Cheat Sheet (VS Code Terminal)**

**🔧 Repository Setup**

| **Command** | **Description** |
| --- | --- |
| git init | Initialize a new Git repo |
| git clone <url> | Clone a repo from GitHub or elsewhere |

**📄 Basic Workflow**

| **Command** | **Description** |
| --- | --- |
| git status | Check repo status |
| git add <file> | Stage specific file |
| git add . | Stage all changes |
| git commit -m "message" | Commit staged changes |
| git log | Full commit history |
| git log --oneline | Short commit history |

**🔁 Branching**

| **Command** | **Description** |
| --- | --- |
| git branch | List branches |
| git branch <name> | Create a new branch |
| git checkout <name> | Switch to a branch |
| git checkout -b <name> | Create and switch to branch |
| git merge <branch> | Merge into current branch |
| git branch -d <name> | Delete a branch |

**🌐 Remote Repositories**

| **Command** | **Description** |
| --- | --- |
| git remote add origin <url> | Connect to remote |
| git remote -v | View remotes |
| git push -u origin <branch> | Push branch & set upstream |
| git push | Push commits |
| git pull | Fetch and merge from remote |

**🧪 Stashing**

| **Command** | **Description** |
| --- | --- |
| git stash | Save uncommitted changes |
| git stash list | View stashes |
| git stash apply | Reapply last stash |

**🔍 Diff & Undo**

| **Command** | **Description** |
| --- | --- |
| git diff | View unstaged changes |
| git diff --staged | View staged changes |
| git reset | Unstage changes |
| git checkout -- <file> | Discard file changes |

| **Purpose** | **Git Command** | **Notes** |
| --- | --- | --- |
| Initialize Git repo | git init | Sets up a new local repo |
| Clone a repo | git clone <repo-url> | Downloads project from remote |
| Check repo status | git status | Shows modified, staged, untracked files |
| Stage changes | git add <file> or git add . | Use . to stage all changes |
| Commit changes | git commit -m "Your message" | Save staged changes with a message |
| View commit history | git log | Use q to quit log view |
| Check short log | git log --oneline | Compact commit view |
| View file diff | git diff | See what's changed but not staged |
| View staged diff | git diff --staged | See staged changes |
| Push to remote repo | git push | Requires remote set up |
| Pull from remote repo | git pull | Sync local with remote |
| Add remote | git remote add origin <url> | Add a new remote |
| Check remotes | git remote -v | Show remote URLs |
| Create new branch | git branch <branch-name> | Only creates branch |
| Switch to a branch | git checkout <branch-name> | Move to an existing branch |
| Create & switch to branch | git checkout -b <branch-name> | Shortcut for both actions |
| Merge branches | git merge <branch-name> | Merge into current branch |
| Delete a branch | git branch -d <branch-name> | Use -D to force delete |
| Stash changes | git stash | Temporarily saves changes |
| Apply last stash | git stash apply | Restore latest stashed changes |
| See stashed entries | git stash list | Shows all stashed items |