

+++ title = “Add query editor help” +++

## Add a query editor help component

By adding a help component to your plugin, you can for example create “cheat sheets” with commonly used queries. When the user clicks on one of the examples, it automatically updates the query editor. It’s a great way to increase productivity for your users.

1. Create a file `QueryEditorHelp.tsx` in the `src` directory of your plugin, with the following content:

```
import React from 'react';
import { QueryEditorHelpProps } from '@grafana/data';

export default (props: QueryEditorHelpProps) => {
  return <h2>My cheat sheet</h2>;
};
```

2. Configure the plugin to use the `QueryEditorHelp`.

```
import QueryEditorHelp from './QueryEditorHelp';

export const plugin = new DataSourcePlugin<DataSource, MyQuery, MyDataSourceOptions>(DataSource)
  .setConfigEditor(ConfigEditor)
  .setQueryEditor(QueryEditor)
  .setExploreQueryField(ExploreQueryEditor)
  .setQueryEditorHelp(QueryEditorHelp);
```

3. Create a few examples.

```
import React from 'react';
import { QueryEditorHelpProps, DataQuery } from '@grafana/data';

const examples = [
  {
    title: 'Addition',
    expression: '1 + 2',
    label: 'Add two integers',
  },
  {
    title: 'Subtraction',
    expression: '2 - 1',
    label: 'Subtract an integer from another',
  },
];

export default (props: QueryEditorHelpProps) => {
```

```

return (
  <div>
    <h2>Cheat Sheet</h2>
    {examples.map((item, index) => (
      <div className="cheat-sheet-item" key={index}>
        <div className="cheat-sheet-item__title">{item.title}</div>
        {item.expression ? (
          <div
            className="cheat-sheet-item__example"
            onClick={e => props.onClickExample({ refId: 'A', queryText: item.expression })}
          >
            <code>{item.expression}</code>
          </div>
        ) : null}
        <div className="cheat-sheet-item__label">{item.label}</div>
      </div>
    ))}
  </div>
);
};

```