

```
'use client';
```

```
import CodePlayground from '@shared/components/code-editor';  
import { languages } from '@shared/components/code-editor/config';  
import EditorProvider, {  
  useEditorContext,  
} from '@shared/components/ui/editor.provider';  
import React, { useEffect } from 'react';
```

```
interface PlaygroundProps {  
  language: string;  
  agent: string;  
}
```

```
function Playground({ language, agent }: PlaygroundProps) {  
  const codeLanguage =  
    languages.find(  
      (lang) => lang.name.toLowerCase() === language.toLowerCase(),  
    ) || languages[0];  
  const { setCodeValue, setLanguage } = useEditorContext();  
  
  useEffect(() => {  
    setCodeValue?.(agent || 'No agent provided');  
    setLanguage?.(codeLanguage);  
  }, [agent, codeLanguage]);
```

```
const resHeight = agent ? 800 : 200;

return (
  <div className="my-14 bg-[#00000080] border border-[#9f9f959] shadow-2xl pt-7 md:p-5
md:py-7 rounded-lg leading-normal">
    <CodePlayground
      height={resHeight}
      downloadCtaClassName="bg-transparent text-destructive"
      readonly
    />
  </div>
);
}
```

```
export default function AgentPlayground({ language, agent }: PlaygroundProps) {
  return (
    <EditorProvider model="">
      <Playground {...{ language, agent }} />
    </EditorProvider>
  );
}
```