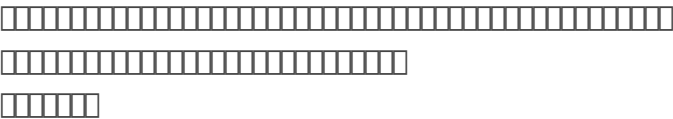




- [\[Diagram\]](#)
- [\[Diagram\]](#)
 - [\[Diagram\]](#)
- [\[Diagram\]](#)
- [\[Diagram\]](#)
 - [JavaScript](#)
 - [PHP](#)
- [Example](#)



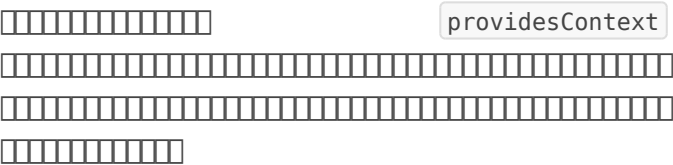
“[Diagram]” [Diagram] ID

[Diagram] [React Context](#)



React Context

render_callback [Diagram]



```
attributes: {
  recordId: {
    type: 'number',
  },
},

providesContext: {
  'my-plugin/recordId': 'recordId',
},
```

XXXXXXXXXX

XXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX

WordPress

XXXXXX

XXXXXXXXXXXX usesContext
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
registerBlockType('my-plugin/record-title', {
  title: 'XXXX',
  category: 'widgets',

  usesContext: ['my-plugin/recordId'],
```

XXXXXX

XXXXXXXXXXXXXXXXXXXX edit JavaScript render_callback PHP
XXXXXXXXXXXXXXXXXXXX JavaScript
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX

XXXXXXXXXXXX save

JavaScript

```
registerBlockType('my-plugin/record-title', {  
  
  edit({ context }) {  
    return 'The record ID: ' + context['my-plugin/recordId'];  
  },  
});
```

PHP

A block's context values are available from the `context` property of the `$block` argument passed as the third argument to the `render_callback` function.

```
register_block_type( 'my-plugin/record-title', array(  
  'render_callback' => function( $attributes, $content, $block ) {  
    return 'The current record ID is: ' . $block->context['my-plugin/recordId'];  
  },  
) );
```

Example

1. Create a `record` block.

```
npm init @wordpress/block --namespace my-plugin record  
cd record
```

2. Edit `src/index.js`. Insert the `recordId` attribute and `providesContext` property in the `registerBlockType` function and add the registration of the `record-title` block at the bottom:

```
registerBlockType( 'my-plugin/record', {  
  // ... cut ...  
  
  attributes: {  
    recordId: {  
      type: 'number',
```

```

    {}},
    {}},

    providesContext: {
      'my-plugin/recordId': 'recordId',
    },

    /**
     * @see ./edit.js
     */
    edit: Edit,

    /**
     * @see ./save.js
     */
    save,
  } );

  registerBlockType( 'my-plugin/record-title', {
    title: 'Record Title',
    category: 'widgets',

    usesContext: [ 'my-plugin/recordId' ],

    edit( { context } ) {
      return 'The record ID: ' + context[ 'my-plugin/recordId' ];
    },

    save() {
      return null;
    },
  } );

```

3. Edit `src/edit.js` for the `record` block. Replace the `Edit` function with the following code:

```

import { TextControl } from '@wordpress/components';
import { InnerBlocks } from '@wordpress/block-editor';

export default function Edit( props ) {
  const MY_TEMPLATE = [ [ 'my-plugin/record-title', {} ] ];

```

```

const {
  attributes: { recordId },
  setAttributes,
} = props;
return (
  <div>
    <TextControl
      __next40pxDefaultSize
      label={ __( 'Record ID' ) }
      value={ recordId }
      onChange={ ( val ) =>
        setAttributes( { recordId: Number( val ) } )
      }
    />
    <InnerBlocks template={ MY_TEMPLATE } templateLock="all" />
  </div>
);
}

```

4. Edit `src/save.js` for the `record` block. Replace the `save` function with the following code:

```

export default function save( props ) {
  return <p>The record ID: { props.attributes.recordId }</p>;
}

```

5. Create a new post and add the `record` block. If you type a number in the text box, you'll see the same number is shown in the `record-title` block below it.

Block Context Example

```

#2
  2026-03-26 17:08:43 UTC
  2026-03-26 17:08:43 UTC

```