WTD PDX 2022

Unit Test the Docs

Why You Should Test Your Code Examples

Ben Perlmutter
Education Engineer @ MongoDB

whoareyou

This talk is for you if...

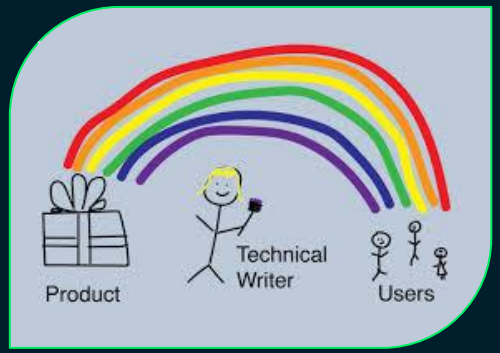


Image from Medium

- You write docs about code
- You want your docs to be more accurate
- You want to spend less time maintaining docs

whoami

Ben Perlmutter

- Work on MongoDB Realm docs team
- Focus on the JavaScript SDKs, Dart/Flutter SDK, and backend-as-a-service
- Interested in docs tooling

Agenda

What code examples and unit tests are

Why you should unit test code examples

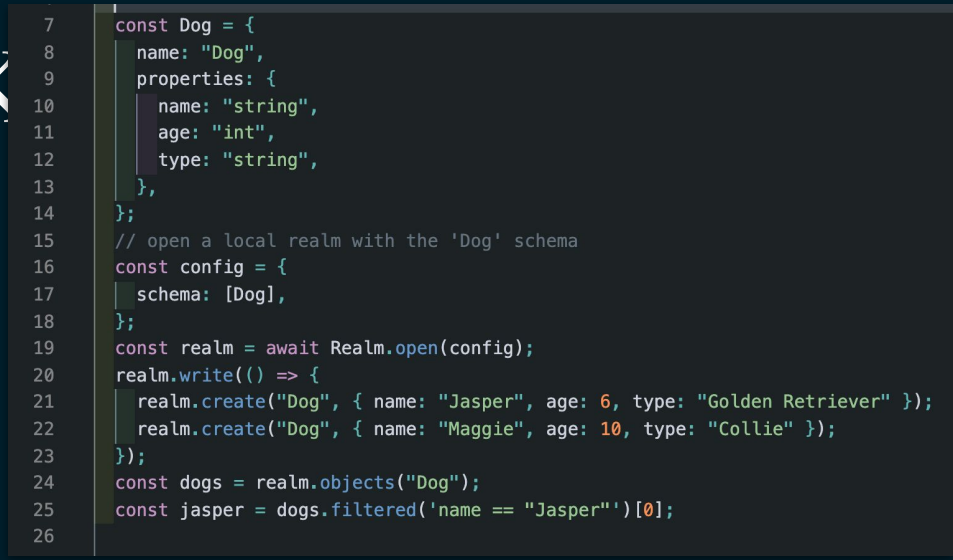
When to unit test code examples

How to unit test code examples

Examples

Realm Node.js SDK

- Mobile database
- Save application data on devices like phones
- Sync data to server



```
const Dog = {
  name: "Dog",
  properties: {
    name: "string",
    age: "int",
    type: "string",
  },
};
// open a local realm with the 'Dog' schema
const config = {
  schema: [Dog],
};
const realm = await Realm.open(config);
realm.write(() => {
  realm.create("Dog", { name: "Jasper", age: 6, type: "Golden Retriever" });
  realm.create("Dog", { name: "Maggie", age: 10, type: "Collie" });
});
const dogs = realm.objects("Dog");
const jasper = dogs.filtered('name == "Jasper"')[0];
```

Unit Tests & Code Examples

```
11
12  describe("Local Realm", () => {
13    let realm;
14    const config = {
15      schema: [Dog],
16    };
17    beforeEach(async () => {
18      // open a local realm with the 'Dog' schema
19      realm = await Realm.open(config);
20    });
21    afterEach(() => {
22      // clean up
23      !realm.isClosed && realm.close();
24      Realm.deleteFile(config);
25    });
26    test("Open Realm", async () => {
27      expect(realm.isClosed).toBe(false);
28    });
29    test("Create data", () => {
30      realm.write(() => {
31        realm.create("Dog", { name: "Jasper", age: 6, type: "Golden Retriever" });
32        realm.create("Dog", { name: "Maggie", age: 10, type: "Collie" });
33      });
34
35      expect(dogs.length).toBe(2);
36      expect(jasper.age).toBe(6);
37    });
38  });
39
```

JavaScript tests in Jest Framework

Overview

Unit Tests

- Simple tests to see if code works as intended
- Part of software development process
- Repeatable
- Frameworks to help

Overview

Code Examples

- Pieces of code embedded in docs
- Show readers how to use code
- Often can copy-and-paste
- Usually stylized with syntax highlighting

Realm Node.js SDK

Realm Node.js SDK

The MongoDB Realm Node.js SDK allows you to use Realm Database and backend Realm apps from Node.js applications written in JavaScript or TypeScript.

Define Object Schema

Define your object schema by creating an object or a class.

```
import Realm from "realm";

const Dog = {
  name: "Dog",
  properties: {
    name: "string",
    age: "int",
    type: "string",
  },
};
// open a local realm with the 'Dog' schema
const config = {
  schema: [Dog],
};
```

Query Realm Database

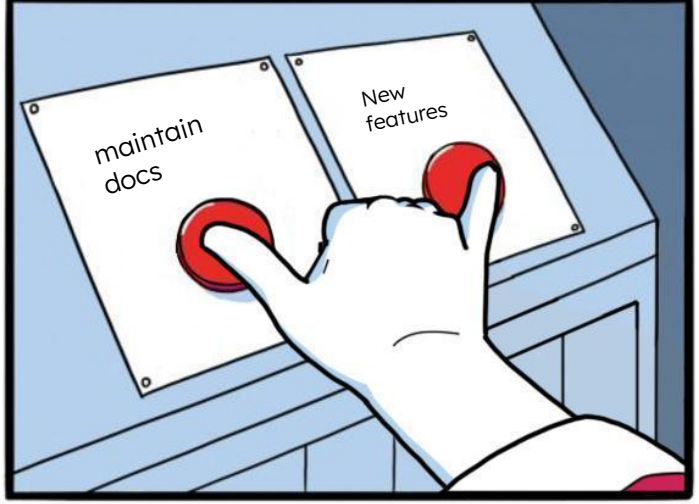
Query for stored objects:

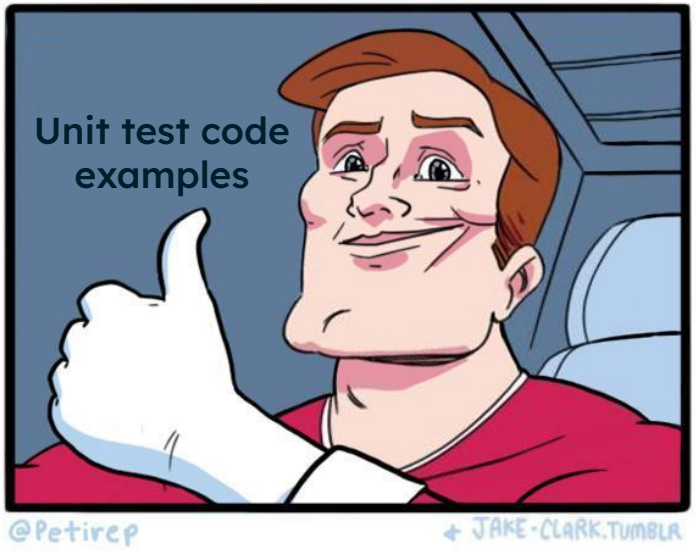
Challenges Documenting Code Examples

Difficult to...

- Keep in sync with interface
- Verify functionality
- Have single source of truth
- Maintain over time
- Work with as a team

How Unit Testing Code Examples Solves These Problems





How Unit Testing Code Examples Solves These Problems

Why Unit Test Code

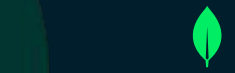
A few of many reasons...

- Validates that code works as intended
- Ensures code actually used by developer
- Makes it easier for other developers to work with the code
- Form of documentation
- Incorporate into CI/CD

Why Unit Test Code Examples

A few of many reasons...

- Validates that code example works as intended
- Ensures code actually used by ~~developer~~ documentarian
- Makes it easier for other ~~developers~~ documentarians to work with the code example
- Form of documentation
- Incorporate into CI/CD for docs as code



And unit tests probably already exist!

Considerations

What unit testing framework should you use?

Are there any language/technology specific tools you can use?

- Python/Jupyter Notebooks
  - *Where Documentation, Cloud-hosted Interactive Tutorials and Continuous Integration Testing Intersect* by Dan Gunter, WTD Portland 2020

When to test code examples

- Code libraries
  - SDKs, middleware
- Tutorials
- Complex examples
- Integration of services

Implementation

Implementation Methods

Copy & Paste (Manual)

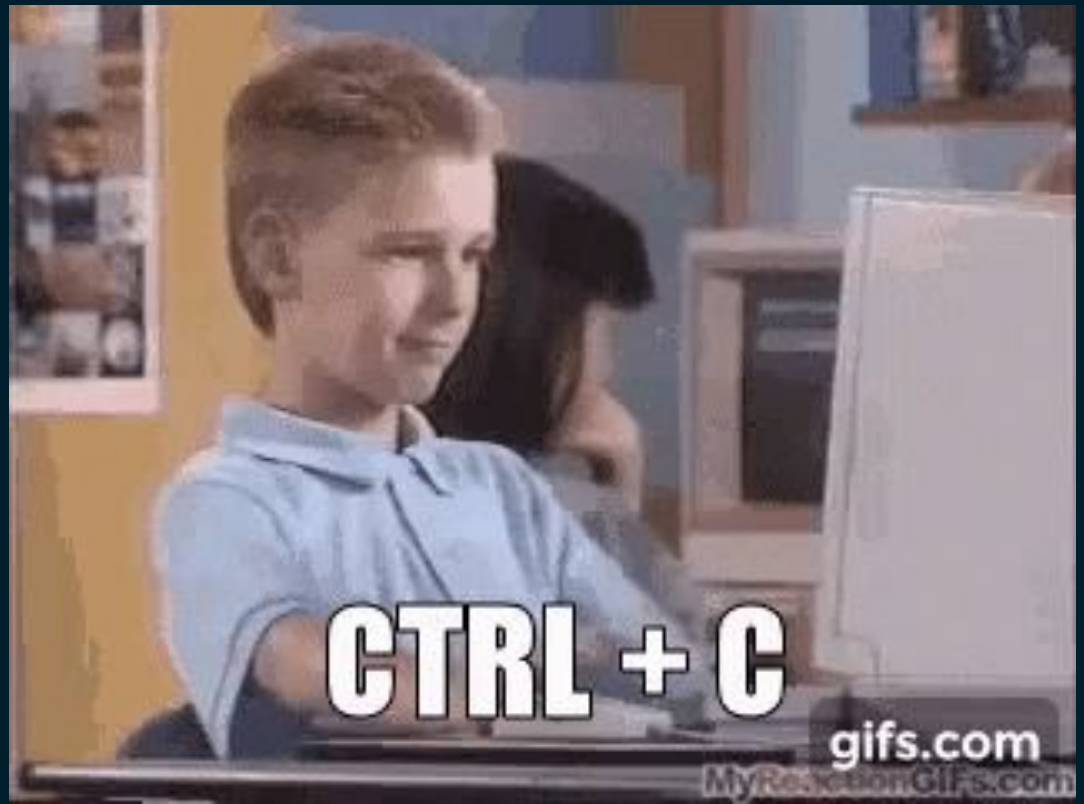
Extract & Transclude (Semi-Automated)

MANUAL IMPLEMENTATION

Copy & Paste

A great starting point for trying out unit testing code examples and less technical teams.

GIF BREAK



FLOW

Copy & Paste

Step 1

Write code example
in unit test
& make test pass

Step 2

Copy code example

Step 3

Paste into docs

STEP 1

Add Code Example to Unit Test

crud.test.js

```
12  describe("Local Realm", () => {
13    let realm;
14    const config = {
15      schema: [Dog],
16      path: "copyPaste.realm",
17    };
18    beforeEach(async () => {
19      // open a local realm with the 'Dog' schema
20      realm = await Realm.open(config);
21    });
22    afterEach(() => {
23      // clean up
24      !realm.isClosed && realm.close();
25      Realm.deleteFile(config);
26    });
27
28    test("Create data", () => {
29      realm.write(() => {
30        realm.create("Dog", { name: "Jasper", age: 6, type: "Golden Retriever" });
31        realm.create("Dog", { name: "Maggie", age: 10, type: "Collie" });
32      });
33
34      expect(dogs.length).toBe(2);
35      expect(jasper.age).toBe(6);
36    });
37  });
38
```

STEP 2

Copy Code Example

crud.test.js

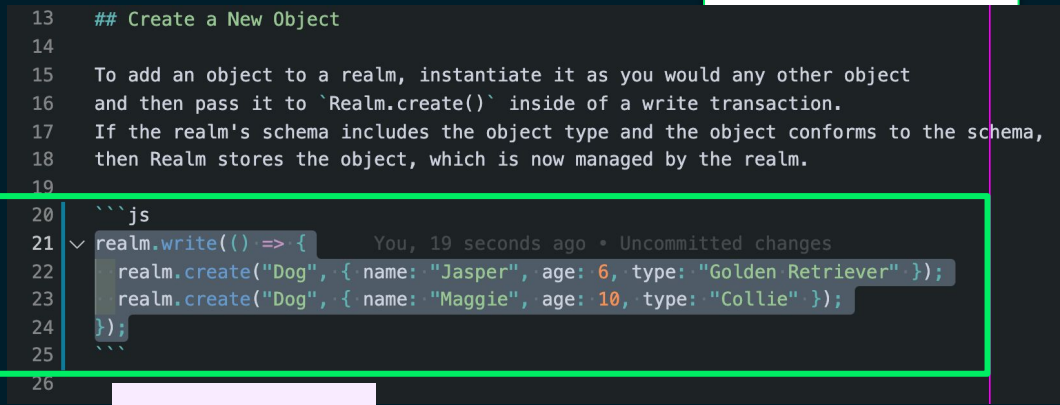
```
12  describe("Local Realm", () => {
13    let realm;
14    const config = {
15      schema: [Dog],
16      path: "copyPaste.realm",
17    };
18    beforeEach(async () => {
19      // open a local realm with the 'Dog' schema
20      realm = await Realm.open(config);
21    });
22    afterEach(() => {
23      // clean up
24      !realm.isClosed && realm.close();
25      Realm.deleteFile(config);
26    });
27
28    test("Create data", () => {
29      realm.write(() => {
30        realm.create("Dog", { name: "Jasper", age: 6, type: "Golden Retriever" });
31        realm.create("Dog", { name: "Maggie", age: 10, type: "Collie" });
32      });
33
34      expect(dogs.length).toBe(2);
35      expect(jasper.age).toBe(6);
36    });
37  });
38
```

CMD|CTRL+C

STEP 3

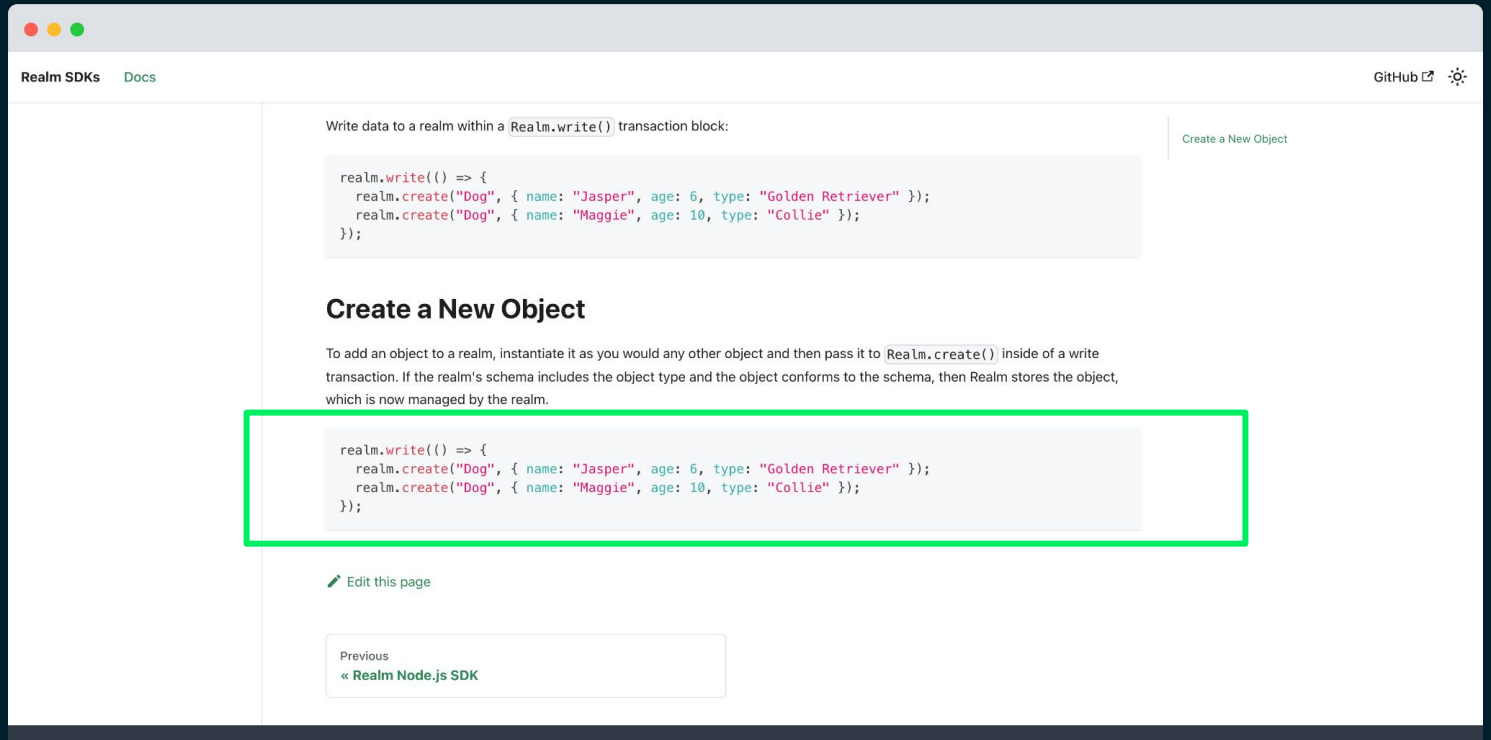
Paste into Docs

write-data.mdx



CMD|CTRL+V

Rendered Docs



Copy & Paste

✅ PROS	🚫 CONS
• Tested code examples • No new tools	• Manual process to include code samples • Hard to maintain • No single source of truth • Formatting gets weird • No programmatic modification

SEMI-AUTOMATED IMPLEMENTATION

Extract & **Transclude**

Automate pulling code examples from unit tests. Add to your docs with markup language transclusion and Bluehawk. Good for quick set up and incremental adoption.

Not the real logo
(no real logo)

TOOL

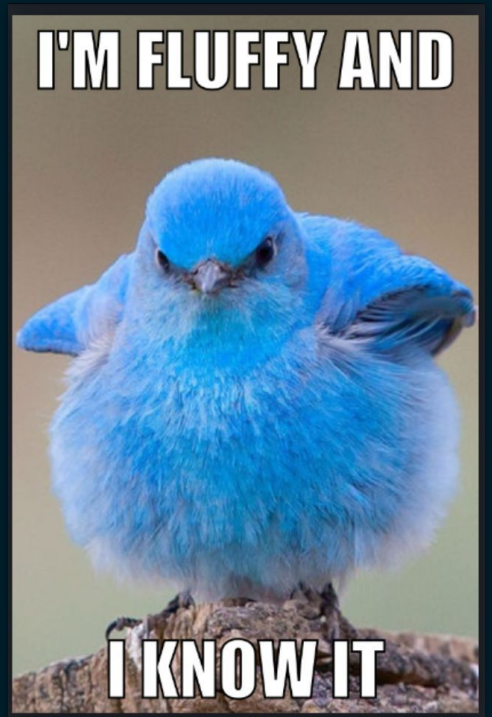
Bluehawk

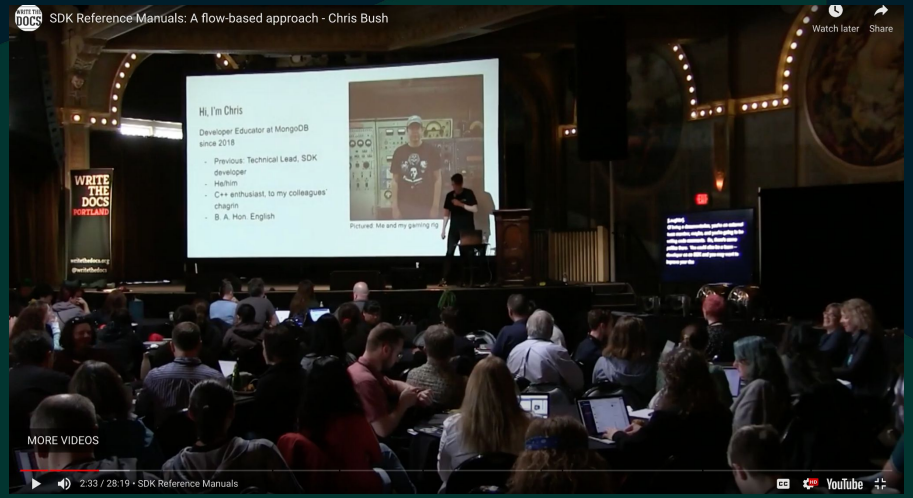
CLI-based tool to extract code examples from source files.

1. Annotate code with comments that specify what to extract
2. Extract code examples using CLI to separate files

MEME BREAK

Alt logo consideration





Bluehawk: A Brief History

- Developed by the Realm Docs team
  - Before my time
  - Led by Caleb Thompson and Chris Bush of previous WTD fame
- Goal to extract code examples in variety of languages
- Just came out with 1.0.0 release
- Get started: `npm install -g bluehawk`

Realm Docs Use of Bluehawk

8

Realm SDKs

41

Contributors

2067

Generated code examples

github.com/mongodb/docs-realm

Bluehawk Walkthrough

1. Annotate Code
2. Extract snippets with CLI
3. Generated file output

bluehawk-example.js

```
1  const beforeSnippet = null;
2
3  // :snippet-start: example-snippet
4
5  const myVar = "example";
6
7  // :remove-start:
8  const removedFromOutput = "won't see me";
9  // :remove-end:
10
11 // :uncomment-start:
12 // const addedToOutput = "will see me!"
13 // :uncomment-end:
14
15 const otherFeatures = `bluehawk also has more advanced features inc
16   - ability to replace content
17   - conditionally render content
18   - and more!`;
19
20 // :snippet-end:
21
22 const afterSnippet = null;
23
```

Remove segments

Include segments

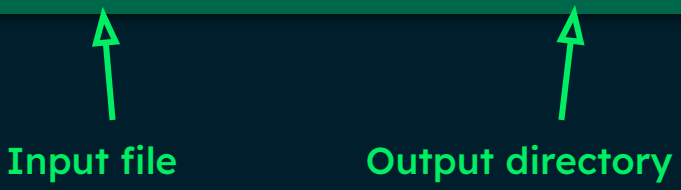
Define start & end

And more!

- Conditionally render
- Replace
- Emphasize

Extract with CLI

```
$ bluehawk snip bluehawk-example.js -o ..
```



Generated Output File

bluehawk-example.codeblock.example-snippet.js

```
1
2   const myVar = "example";
3
4
5   const addedToOutput = "will see me!";
6
7   const otherFeatures = `bluehawk also has more advanced features including:
8   - ability to replace content
9   - conditionally render content
10  - and more!`;
11
12
```

Learn More

mongodb-university.github.io/Bluehawk/

DEFINITION

Transclusion

When you include contents of one document inside of another.

Common feature in documentation platforms:

- Sphinx literalinclude
- Docusaurus raw-loader imports
- Hugo readFile
- Jekyll includes

Example

Source files

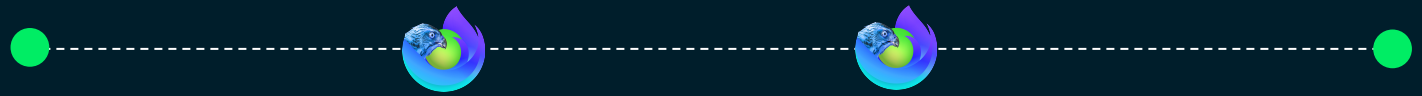
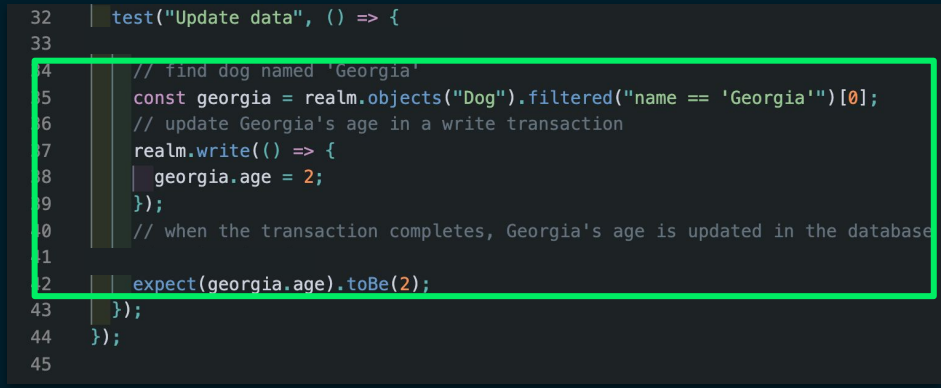
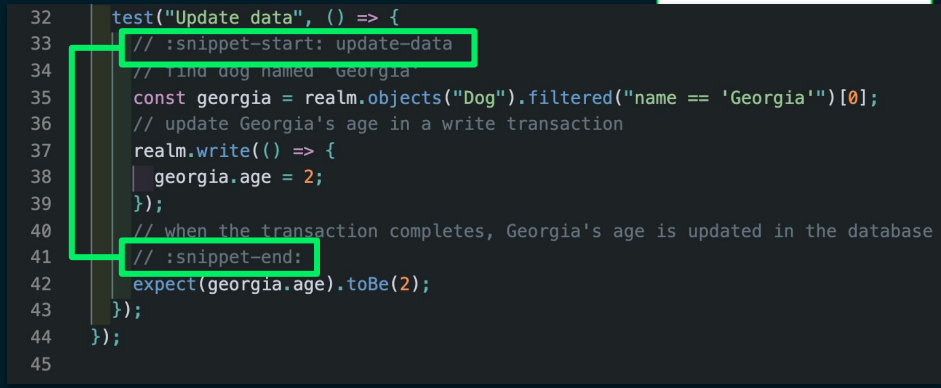
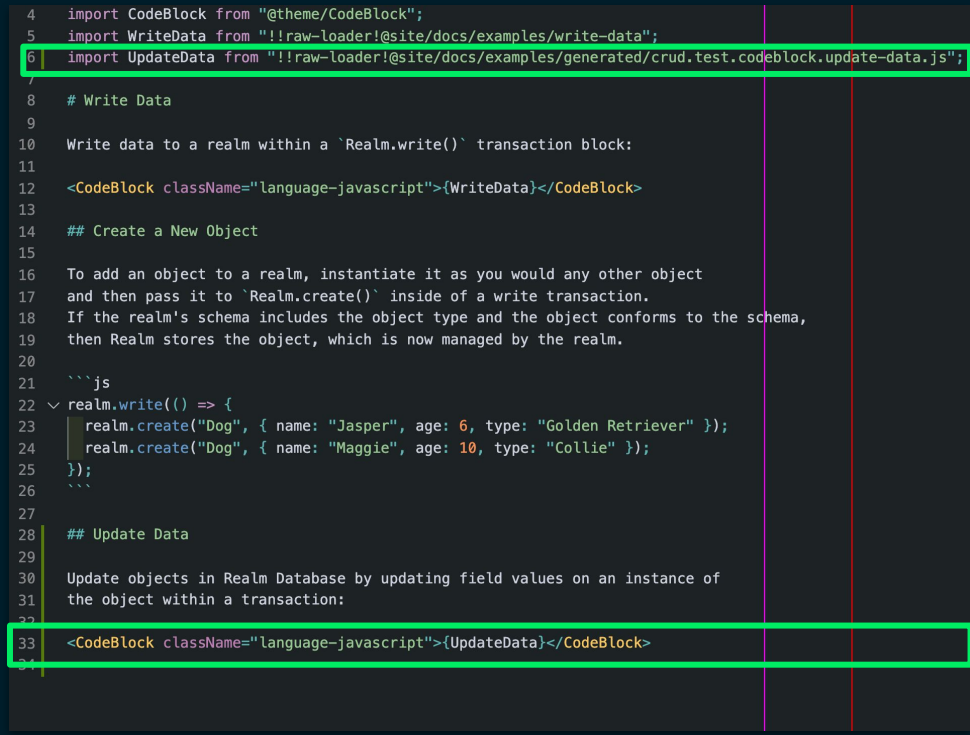
ex.js

```
const foo = “bar”;
```

doc.md

```
# My code example

```js
{{ex.js}}
````

Explanation…
```

Rendered docs

example.com/doc.html

My code Example

```
const foo = “bar”;
```

Explanation…

FLOW

Extract & Transclude



Step 1

Write code example in unit test & make test pass

Step 2

Annotate unit test noting code to extract

Step 3

Extract code example into new file

Step 4

Transclude code example in docs

STEP 1

Add Code Example to Unit Test



```
32    test("Update data", () => {
33
34      // find dog named 'Georgia'
35      const georgia = realm.objects("Dog").filtered("name == 'Georgia'")[0];
36      // update Georgia's age in a write transaction
37      realm.write(() => {
38        georgia.age = 2;
39      });
40      // when the transaction completes, Georgia's age is updated in the database
41
42      expect(georgia.age).toBe(2);
43    });
44  });
45
```

STEP 2

Annotate Unit Test with Code to Extract

crud.test.js



STEP 3

Extract Code Examples into New File

```
$ bluehawk snip crud.test.js \
-o $DOCS_SITE/examples/generated
```

crud.test.codeblock.update-data.js

```
1  // find dog named 'Georgia'
2  const georgia = realm.objects("Dog").filtered("name == 'Georgia'")[0];
3  // update Georgia's age in a write transaction
4  realm.write(() => {
5    georgia.age = 2;
6  });
7  // when the transaction completes, Georgia's age is updated in the database
8
```

STEP 4

Transclude Code Example in Docs



```
import CodeBlock from "@theme/CodeBlock";
import WriteData from "!!raw-loader!@site/docs/examples/write-data";
import UpdateData from "!!raw-loader!@site/docs/examples/generated/crud.test.codeblock.update-data.js";

# Write Data

Write data to a realm within a `Realm.write()` transaction block:

<CodeBlock className="language-javascript">{WriteData}</CodeBlock>

## Create a New Object

To add an object to a realm, instantiate it as you would any other object
and then pass it to `Realm.create()` inside of a write transaction.
If the realm's schema includes the object type and the object conforms to the schema,
then Realm stores the object, which is now managed by the realm.

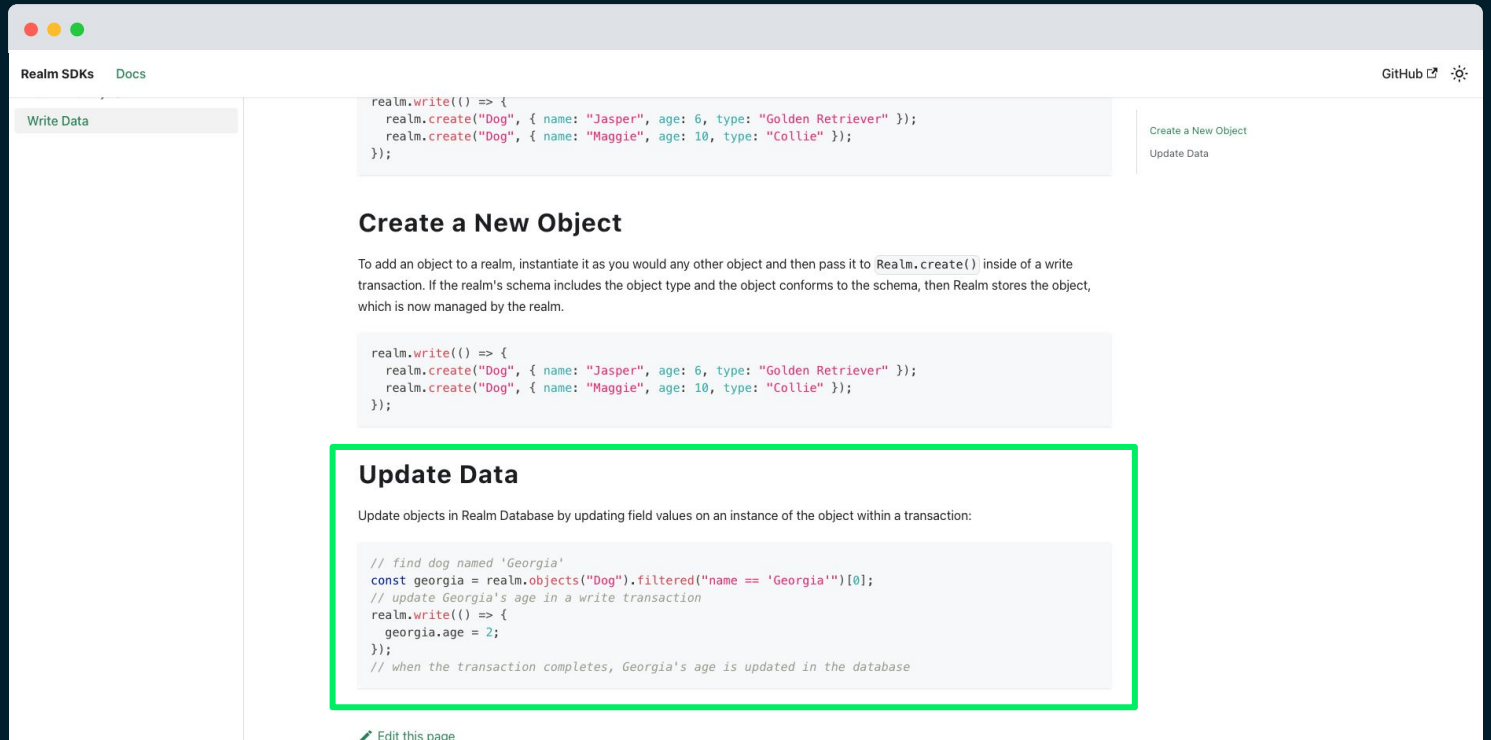
```js
realm.write(() => {
  realm.create("Dog", { name: "Jasper", age: 6, type: "Golden Retriever" });
  realm.create("Dog", { name: "Maggie", age: 10, type: "Collie" });
});
```

Update Data

Update objects in Realm Database by updating field values on an instance of
the object within a transaction:

<CodeBlock className="language-javascript">{UpdateData}</CodeBlock>
```

Rendered Docs



Extract & Transclude

✅ PROS	🚫 CONS
• Advantages of unit tested code examples • Better record of where code examples are • Single source of truth • More repeatable • Can use Bluehawk to modify code examples from source code	• Less automated than we may like • Have to rerun CLI operations if regenerating same file • Commit generated files to version control

Continuous Integration

DEFINITION

Continuous Integration (CI) Tests

Tests that run when you add code to a remote version control system. (Like PR against `main` branch of a git repository on Github)

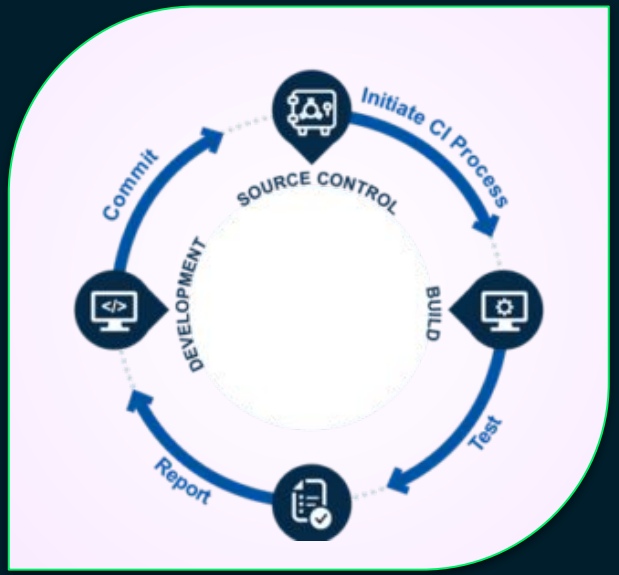


Image from Worksoft

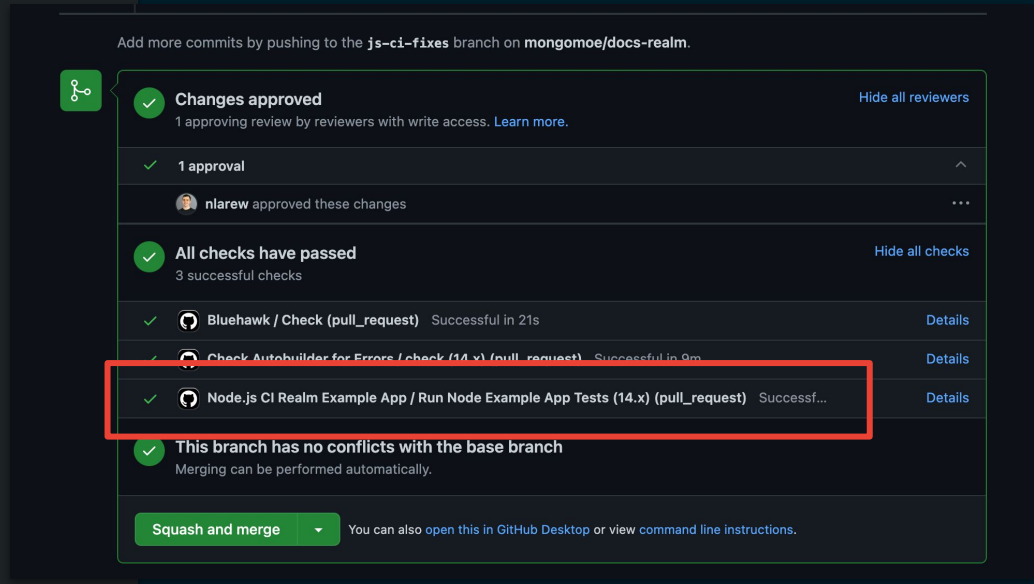
Why CI Test Code Examples

- Make sure examples are working before publishing docs
- Public record that working before publication
- Use tools like Github Actions/TravisCI

Github Action CI to test Realm Node.js

GitHub Workflow (github-workflow.json)

```yaml
name: Node.js CI Realm Example App

on:
  pull_request:
    paths:
      - "examples-src/**"

jobs:
  build:
    name: Run Node Example App Tests
    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [14.x]

    steps:
      - uses: actions/checkout@v3
      - name: Use Node.js ${{ matrix.node-version }}
        uses: actions/setup-node@v1
        with:
          node-version: ${{ matrix.node-version }}
      - run: cd examples-src && npm install
      - run: cd examples-src && npm test

```



What We Covered

Why Unit Test Code Examples

- Validates that code works as intended
- Ensures code actually used
- Makes it easier for others to work with the code example
- Form of documentation
- Incorporate into CI/CD for docs as code

Ways to Implement

- Copy and paste (manual)
- Extract & transclude (semi-automated)

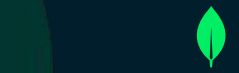
CI/CD

- Add unit tests to your CI/CD pipeline to help docs be accurate

Thank you for your time.

Get in touch

ben.p@mongodb.com



We’re hiring

mongodb.com/careers

Feel free to reach out to me at
ben.p@mongodb.com

FINAL GIF
BREAK

Enjoy the rest of WTD