

CDA Test Suite Setup Guide

Quick Setup (5 minutes)

1. Place the test file

```
bash

# In your Next.js project
mkdir -p public/test
cp cda-test-suite.html public/test/
```

2. Configure GraphQL endpoint

Edit line 369 in `cda-test-suite.html`:

```
javascript

const GRAPHQL_ENDPOINT = 'http://localhost:10003/graphql'; // Your WordPress URL
```

3. Fix CORS (if needed)

Add the `cors-config.php` code to:

- `/wp-content/mu-plugins/cors-config.php` (recommended)
- OR your theme's `functions.php`

4. Access the test suite

```
http://localhost:3000/test/cda-test-suite.html
```

What This Tool Does

Real Query Testing

- Tests GraphQL queries against your actual WordPress backend
- Shows exact error messages if queries fail
- Validates data structure before you build components

Query Library

- Pre-loaded with 6 working queries for your project
- Saves successful queries to browser localStorage
- Copy-paste tested queries directly into your code

Benefits

1. **Faster Development** - Test queries without building React components
 2. **Debug Easier** - See exact GraphQL errors immediately
 3. **Team Sharing** - Export working queries for team members
 4. **Documentation** - Keep a library of all working queries
-



Pre-Loaded Queries

The tool includes these tested queries:

1. GetPages - List all pages

```
graphql

query GetPages {
  pages {
    nodes {
      id
      title
      slug
      uri
    }
  }
}
```

2. GetHomepage - Homepage content

```
graphql
```

```
query GetHomepage {  
  page(id: "/", idType: URI) {  
    title  
    featuredImage {  
      node {  
        sourceUrl  
        altText  
      }  
    }  
  }  
}
```

3. GetServices - All services

```
graphql  
  
query GetServices {  
  services {  
    nodes {  
      id  
      title  
      slug  
      excerpt  
    }  
  }  
}
```

4. GetGlobalOptions - Global blocks

```
graphql  
  
query GetGlobalOptions {  
  globalOptions {  
    globalSharedContent {  
      whyCdaBlock {  
        title  
        subtitle  
      }  
    }  
  }  
}
```

Troubleshooting

"Connection Error"

1. Check WordPress is running: `http://localhost:10003`
2. Verify WPGraphQL plugin is active
3. Check GraphQL endpoint: `http://localhost:10003/graphql`

CORS Error

Add the cors-config.php file to WordPress (see above)

"Cannot query field"

1. Field doesn't exist in your ACF setup
2. Check field has `show_in_graphql: true`
3. Verify in GraphiQL: `http://localhost:10003/graphql`

Export/Import Queries

Export saved queries:

```
javascript

// Run in browser console
copy(localStorage.getItem('cdaSavedQueries'))
```

Import queries:

```
javascript

// Run in browser console
localStorage.setItem('cdaSavedQueries', 'PASTE_JSON_HERE')
```

Typical Workflow

1. **Open test suite** in browser
2. **Write query** in the textarea
3. **Test it** - see immediate results

4. **Fix errors** if any appear
 5. **Save query** when it works
 6. **Copy to code** when ready to implement
-

ROI Calculator Testing

The calculator section tests your conversion formulas:

- Enter sample data
 - Verify calculations match Excel
 - Test edge cases (0 values, high numbers)
 - Validate business logic
-

Test Status Tracking

- **Yellow:** Not tested yet
- **Green:** Test passed
- **Red:** Test failed

Tracks:

- Total tests: 47
 - Coverage percentage
 - Pass/fail counts
 - Saves passed tests to localStorage
-

Pro Tips

1. **Test incrementally** - Start with simple queries, build complexity
 2. **Save everything** - Keep working queries for reference
 3. **Document fields** - Note which ACF fields map to which queries
 4. **Test early** - Validate data structure before building UI
 5. **Share queries** - Export for team members
-

File Locations Summary

```
your-project/
├── public/
│   ├── test/
│   │   └── cda-test-suite.html    ← Test interface
├── wordpress/
│   ├── wp-content/
│   │   ├── mu-plugins/
│   │   │   └── cors-config.php    ← CORS fix
└── docs/
    └── test-suite-setup.md        ← This file
```