

Beyond Functional: Crafting a World-Class API Experience

A proposal for elevating our API documentation
to the industry standard.

We have a solid blueprint.

Our API is comprehensive and well-defined. The current documentation captures all the necessary technical details for our four core endpoints.

GET /dashboard-sync

Fetches all data for the dashboard UI.

POST /urls/create

Generates a new short link.

GET /{alias}

The public-facing redirect endpoint.

DELETE /urls/delete/{id}

Removes a specified link.

But a blueprint isn't the final product.

While technically accurate, our current format places the entire burden of interpretation on the developer.

♦ Endpoint 2: Create Short URL

Purpose: Generates a new short link. Supports custom aliases...

- **Method:** POST
- **Path:** /urls/create
- **Auth Required:** Yes

Request Body:

Validation Logic:

1. **Alias:** Must match Regex ^[a-zA-Z0-9-_]{3,20}\$.
2. **Original URL:** Saved **exactly** as input...

Side Effects:

- Atomically increments total_links...

Error Responses:

- 400 Bad Request: "Alias invalid..."
- 409 Conflict: "Alias already taken!"

↑ Fancer Content

Is it easy to read?

Can you test it instantly?

Does it reflect the quality of our code?

The best APIs treat their documentation as a user interface.

This is the experience professional developers at companies like Stripe and Twilio have come to expect. It's clear, interactive, and builds trust in the underlying product.

The screenshot shows a user interface for an API Reference. On the left, a sidebar menu includes 'API Reference', 'Introduction', 'Authentication', 'URLs' (with 'Create URL' selected), 'Delete URL', and 'Get URL Info'. The main content area has a title 'Create Short URL' and a description: 'Generates a new short link from a given long URL. Supports optional custom aliases.' Below this is a 'Parameters' table:

| Name | Type | Description | Required |
|------------|---------|-----------------------------------|----------|
| url | string | The original long URL. | Required |
| alias | string | Optional custom short code. | Optional |
| expires_in | integer | Time in seconds until expiration. | Optional |
| expires_in | string | Optional custom code. | Optional |

Below the table is a 'Request Body' section with a partial JSON example:

```
{  
  "url": "string"  
}
```

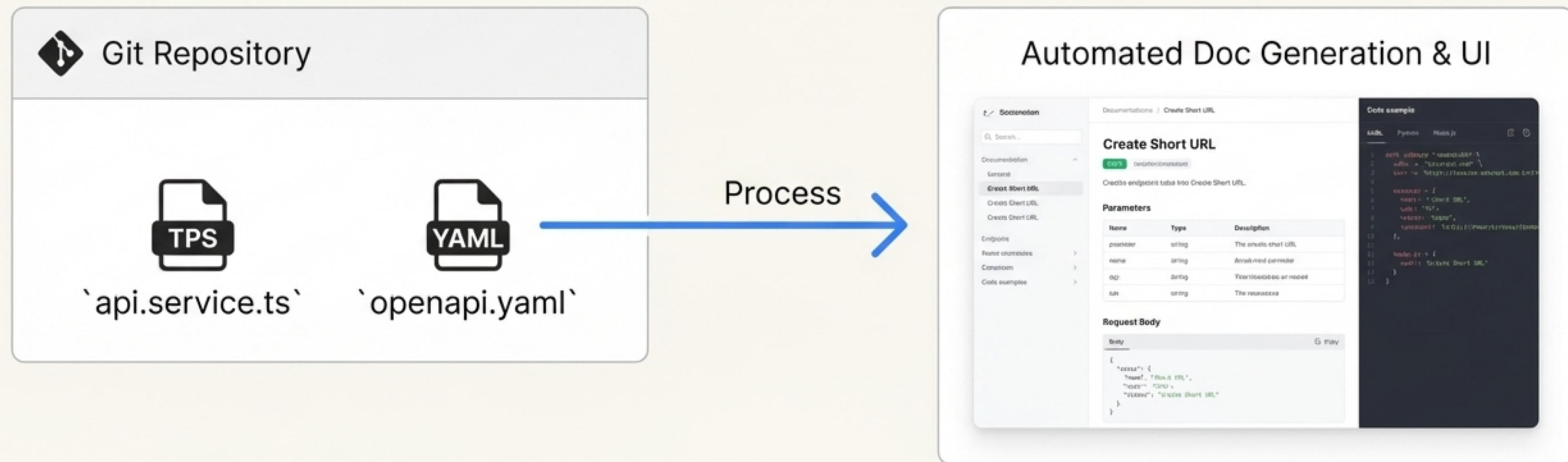
On the right, there are tabs for 'cURL', 'Python', and 'Node.js', each showing a command or code snippet for creating a short URL. The 'cURL' tab is active, displaying:

```
curl -X POST https://api.example.com/urls/create \  
-H 'Authorization: Bearer YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "url": "https://verylongurl.com/some-resource"  
  "alias": "my-link"  
}'
```

In the bottom right corner, there is a small logo for 'NotebookLM'.

Great documentation lives with the code.

Modern docs aren't written in Word or Confluence. They are defined in a machine-readable format (OpenAPI) that ensures they are always in sync with the actual code. If the code changes, the docs update automatically.



It's more than just a list of endpoints.



GUIDES (The 'How-To' Manual)

Essential for getting started.

- Authentication (e.g.,
`Authorization: Bearer <token>`)
- Base URLs (Prod vs. Staging)
- A comprehensive Error Reference
- Rate Limit policies



API REFERENCE (The Dictionary)

The technical specification for every endpoint.

- In-depth details for every parameter
- Clear definitions for request bodies
- Exhaustive response schemas
- Must be precise and complete



INTERACTIVITY (The Sandbox)

The game-changer for developer experience.

- Ability to make live API calls
- Test endpoints directly from the browser
- Static text is dead

Our Current `POST /urls/create`

Purpose: Generates a new short link. Supports custom aliases and 'Raw Input' mode.

Method: POST

Path: /urls/create

Request Body: A plain list of fields.

Validation Logic: Alias: Must match Regex `^[a-zA-Z0-9-_]{3,20}$`.

Original URL: Saved exactly as input...

Side Effects: Atomically increments total_links...

Creates LINK#<code> entity in DynamoDB.

Error Responses: 400 Bad Request: 'Alias invalid...', 409 Conflict:

'Alias already taken!'

The Standard: `POST /urls/create`

POST /urls/create

Generates a new shortened URL for a given destination.
Supports optional custom aliases.

REQUEST BODY

| Field | Type | Required | Description | Constraints |
|-------------|--------|----------|------------------------|----------------------------|
| originalUrl | string | Yes | The destination URL. | Must be valid URL format. |
| customCode | string | No | Optional custom alias. | 3-20 chars, [a-zA-Z0-9-_]. |

Example Request

```
{  
  "originalUrl": "https://www.example.com/long/path/to/resource",  
  "customCode": "my-cool-link"  
}
```

Success Response (200 OK)

```
{  
  "success": true,  
  "data": {  
    "shortUrl": "https://api.rayasoft.tech/my-cool-link",  
    "originalUrl": "https://www.example.com/long/path/to/resource",  
    "createdAt": "2023-10-27T10:00:00Z"  
  }  
}
```

Try It Out

We don't build the UI. We describe the API.

This is a solved problem. We leverage a standard specification and a rich ecosystem of tools to generate professional documentation automatically.



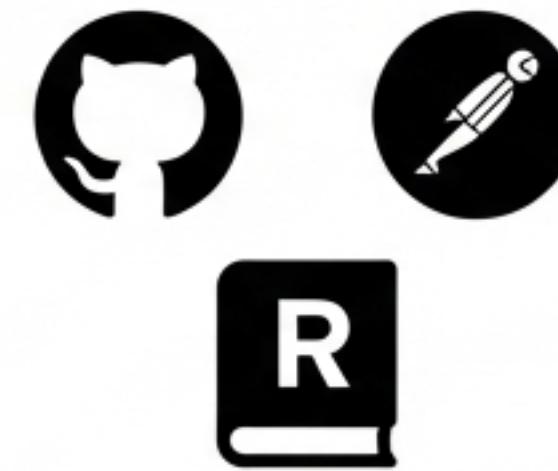
DEFINE

The Single Source of Truth.
(`openAPI 3.0 Specification`)



RENDER

The UI Generators.

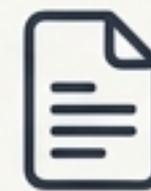


HOST

The Publishing Platforms.

Our roadmap to a world-class experience.

This is not a multi-month project. We can achieve a significant improvement in a matter of days with this three-step plan.



PHASE 1: SPECIFICATION

Convert our existing API contract into a formal `openapi.yaml` file.

Est. Time: 2-3 days



PHASE 2: IMPLEMENTATION

Choose and configure a renderer (Recommendation: Redoc for its clean UI) and host it on GitHub Pages.

Est. Time: 1 day



PHASE 3: AUTOMATION

Integrate the documentation generation into our CI/CD pipeline to ensure it's always up-to-date.

Est. Time: Ongoing

The ROI of a great developer experience.



Faster Onboarding

New developers become productive in hours, not days, thanks to clear, interactive guides.



Reduced Support Load

Self-serve, interactive docs answer questions before they're ever asked in Slack or email.



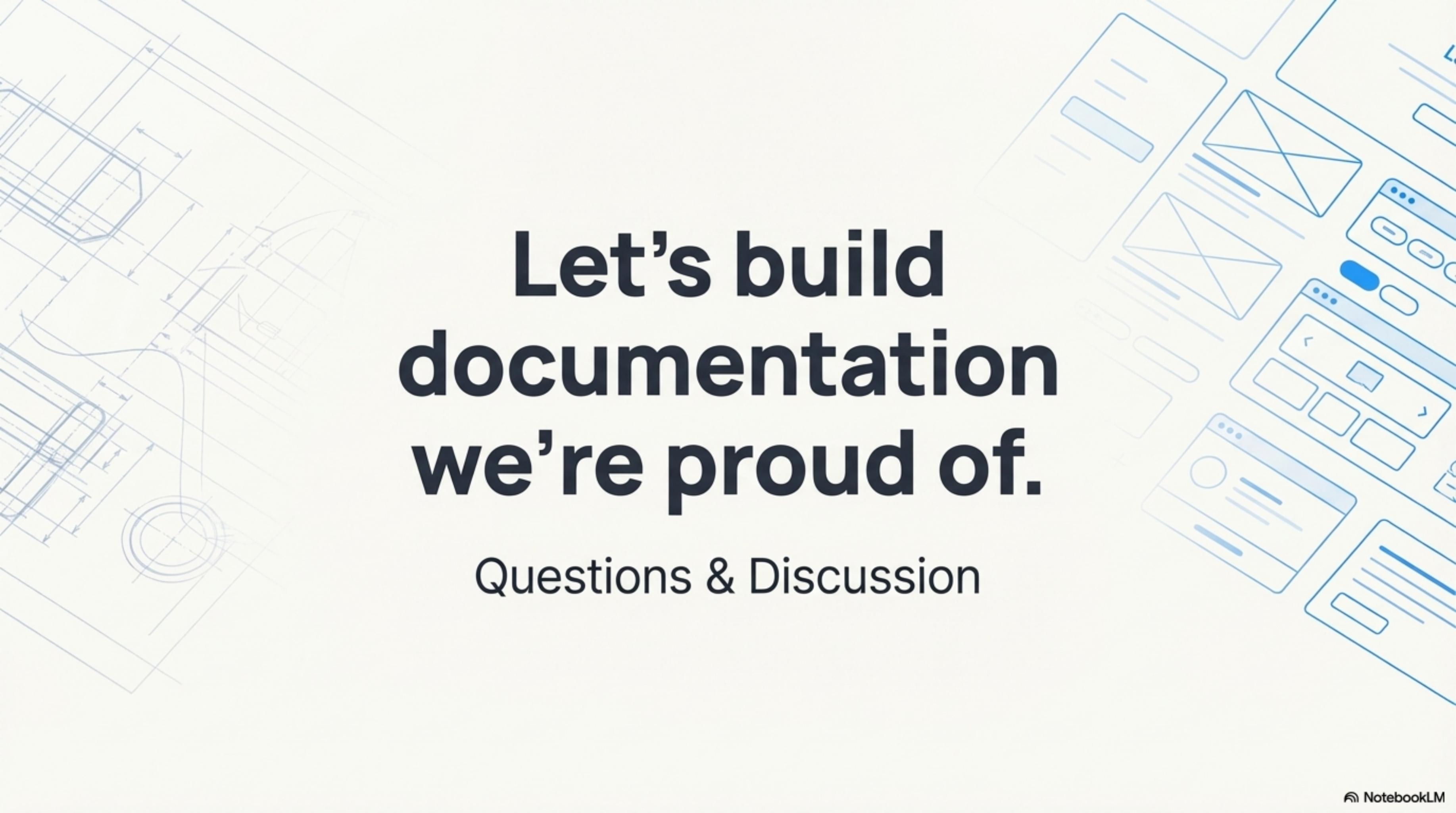
Increased API Adoption & Trust

A professional presentation signals a reliable, high-quality product worth building on.



Enables Automation

A machine-readable spec is the foundation for auto-generating client SDKs, integration tests, and monitoring tools.



Let's build documentation we're proud of.

Questions & Discussion