



CONNECTING RESEARCH,  
IDENTIFYING KNOWLEDGE

# DataCite API Training

**Kelly Stathis**

*Technical Community Manager*

**Mike Bennett**

*Application Developer*



[@datacite](https://twitter.com/datacite)

# What we'll cover

1. Accessing DataCite APIs
2. Retrieving DOI metadata with the REST API:
  - a. List DOIs in a repository
  - b. Get metadata for a single DOI
3. Creating/updating DOIs with the REST API:
  - a. Create a DOI using JSON
  - b. Create a DOI using XML (base64 encoded)
  - c. Update DOI metadata

# **Accessing DataCite APIs**

## APIs for creating/updating DOIs

<b>REST API</b> (new integrators should use this API!)	<ul style="list-style-type: none"><li>• Create/update DOIs (repositories only)</li><li>• Query/retrieve DOI metadata</li><li>• Query/retrieve DOI citations, usage reports, provenance</li><li>• Query/retrieve information about DataCite prefixes, members and consortium organizations (providers), and and repositories (clients)</li></ul>
<b>MDS API</b>	<ul style="list-style-type: none"><li>• Create/update DOIs</li></ul>

# API endpoints

	Test environment	Production environment	Authentication Required?
<b>REST</b>	<a href="https://api.test.datacite.org">https://api.test.datacite.org</a>	<a href="https://api.datacite.org">https://api.datacite.org</a>	<b>Yes</b> for member API No for public API
<b>MDS</b>	<a href="https://mds.test.datacite.org">https://mds.test.datacite.org</a>	<a href="https://mds.datacite.org">https://mds.datacite.org</a>	<b>Yes</b>

## Member vs Public REST API

- Same endpoint URLs; use authentication to access the member API
- Member API uses dedicated servers; not affected by public API traffic
- DOIs can only be registered/updated using member API
- Draft DOIs and contact information are excluded from public API responses

# What else do I need?

## To make API requests, you'll need:

- Repository ID
- Repository password
- Prefix assigned to that repository
- Tool capable of sending/receiving HTTP requests (cURL, Postman, etc)

## If you don't have an existing repository, you'll need to:

- Create a repository in Fabrica and set the password

For this training, we'll use the test APIs and test Fabrica environment,  
**<https://doi.test.datacite.org>**

# Set-up

You can run these requests using Postman, a web-based tool. We have pre-configured the requests in a Postman collection. To access it:

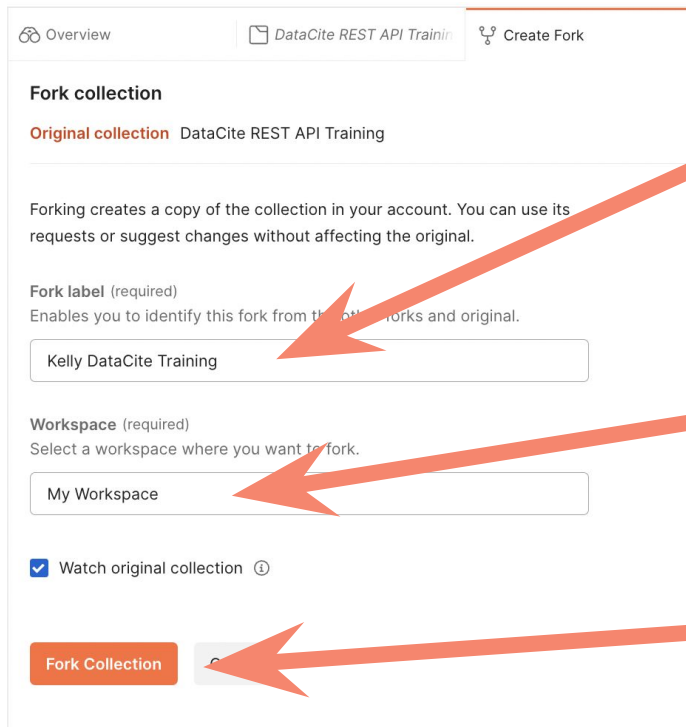
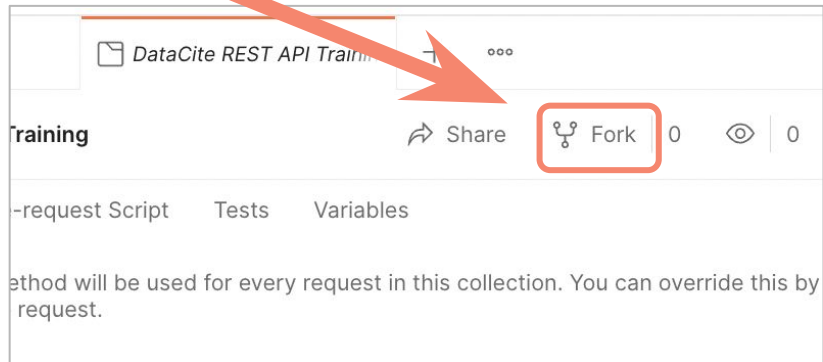
1. Create a free Postman account (or login, if you already have an account)  
<https://identity.getpostman.com/signup>
2. Visit the DataCite REST API Training collection at  
<https://www.postman.com/datacite/workspace/951e6d09-cbe8-46ca-881d-a26c10746ce1/>

If you're familiar with curl, you can use it to run these example requests in the command line.

# Set-up

Fork the collection to copy it into your account (required in order to run the requests)

Click "Fork"



The screenshot shows the 'Fork collection' form. At the top, there are tabs for 'Overview', 'DataCite REST API Training', and 'Create Fork'. The form contains the following fields and options:

- Fork collection**
  - Original collection:** DataCite REST API Training
- Forking description:** Forking creates a copy of the collection in your account. You can use its requests or suggest changes without affecting the original.
- Fork label (required):** Enables you to identify this fork from the other forks and original. The input field contains 'Kelly DataCite Training'.
- Workspace (required):** Select a workspace where you want to fork. The dropdown menu shows 'My Workspace'.
- Watch original collection:** A checkbox that is checked.
- Buttons:** 'Fork Collection' (orange) and a 'Cancel' button (grey).

Red arrows point from the instruction boxes to the 'Fork label' field, the 'Workspace' dropdown, and the 'Fork Collection' button.

Give your fork a name different from the original

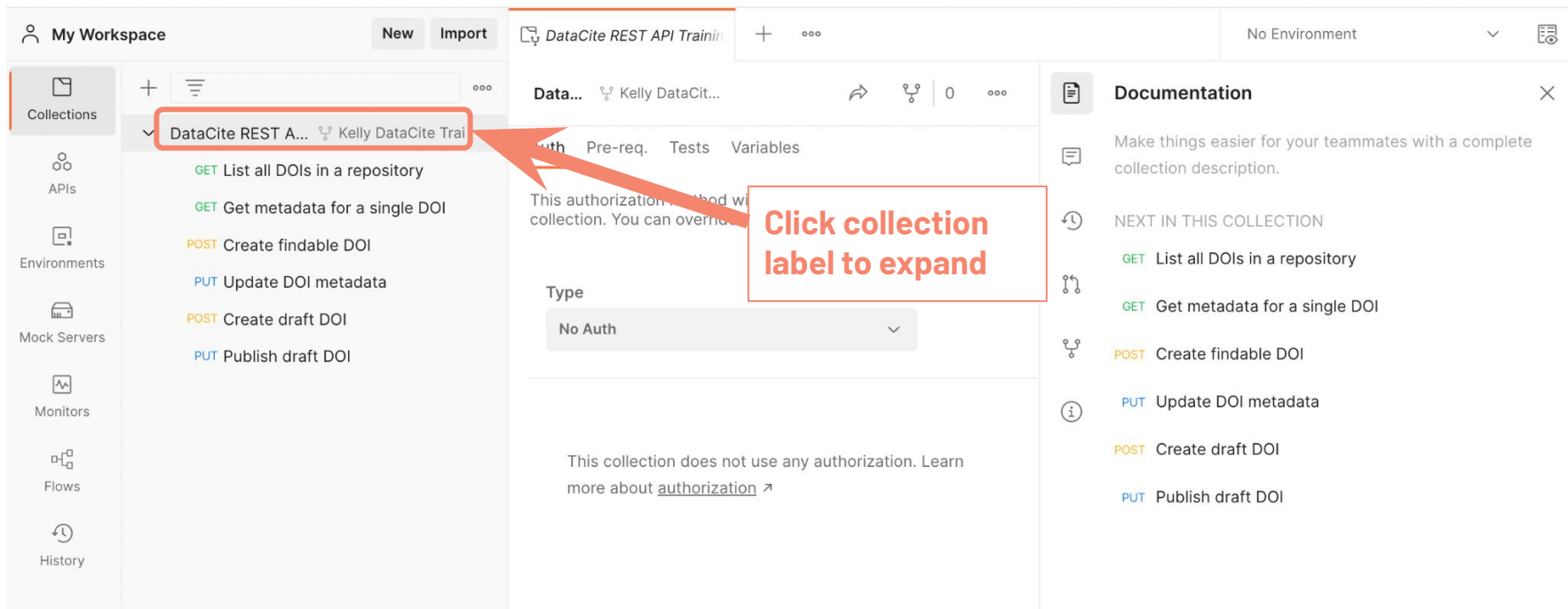
Leave workspace set to "My Workspace"

Click "Fork collection"



# Set-up

You can now access your copy of the collection.



The screenshot displays the DataCite REST API Training interface. On the left sidebar, under 'My Workspace', the 'Collections' section is expanded, showing a list of collections. The collection 'DataCite REST A...' is highlighted with a red box. A red arrow points from a text box 'Click collection label to expand' to this collection. The main panel shows the details of the selected collection, including a list of endpoints (GET List all DOIs in a repository, GET Get metadata for a single DOI, POST Create findable DOI, PUT Update DOI metadata, POST Create draft DOI, PUT Publish draft DOI) and a 'Type' dropdown set to 'No Auth'. The right sidebar shows the 'Documentation' section with a description and a list of endpoints.

**My Workspace** New Import

**Collections**

- APIs
- Environments
- Mock Servers
- Monitors
- Flows
- History

**DataCite REST A...** Kelly DataCite Trai

- GET List all DOIs in a repository
- GET Get metadata for a single DOI
- POST Create findable DOI
- PUT Update DOI metadata
- POST Create draft DOI
- PUT Publish draft DOI

**Data...** Kelly DataCit...

Auth Pre-req. Tests Variables

This authorization method will be used for this collection. You can override this at the endpoint level.

Type

No Auth

This collection does not use any authorization. Learn more about [authorization](#)

**Documentation**

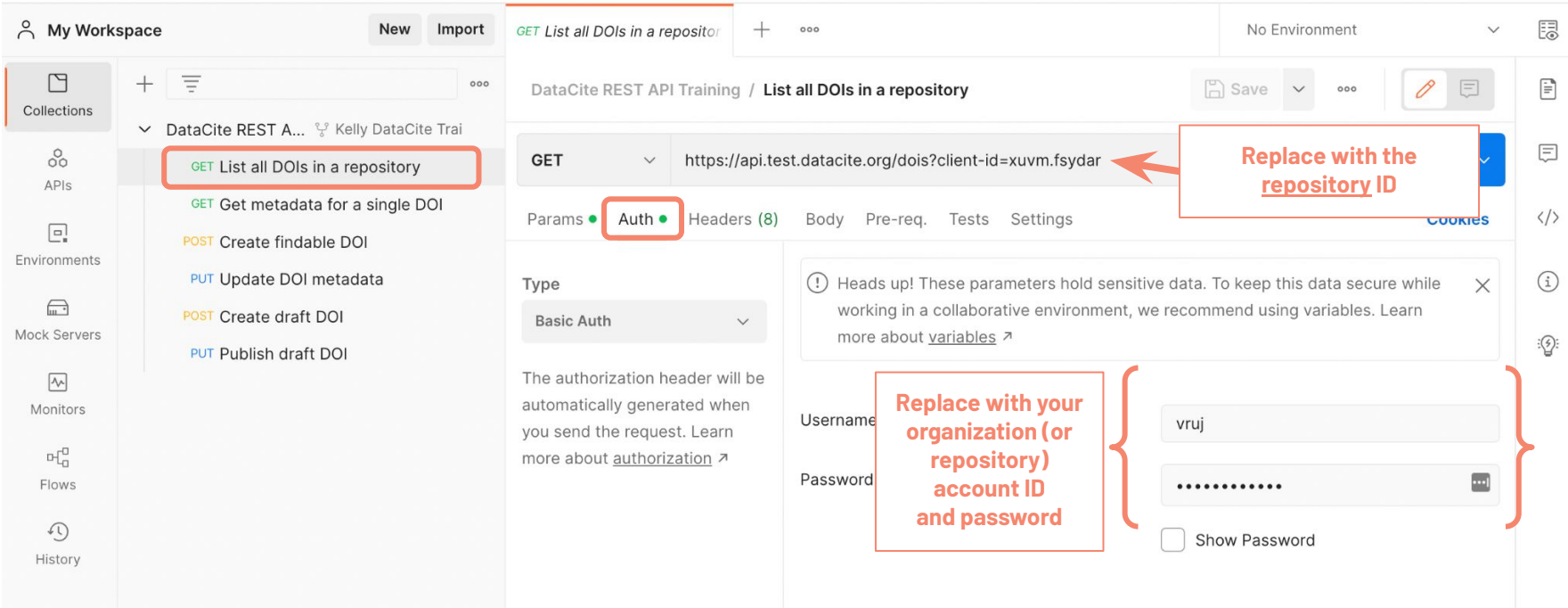
Make things easier for your teammates with a complete collection description.

NEXT IN THIS COLLECTION

- GET List all DOIs in a repository
- GET Get metadata for a single DOI
- POST Create findable DOI
- PUT Update DOI metadata
- POST Create draft DOI
- PUT Publish draft DOI

# **Retrieving data with the REST API**

# List all DOIs in a repository



The screenshot displays the DataCite REST API Training interface. On the left, the 'My Workspace' sidebar shows a list of collections, with 'DataCite REST A...' selected. The 'GET List all DOIs in a repository' endpoint is highlighted. The main panel shows the endpoint configuration for the 'GET' method, with the URL 'https://api.test.datacite.org/does?client-id=xuvm.fsydar'. The 'Auth' tab is selected, showing 'Basic Auth' as the authentication type. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)'. The 'Username' field is set to 'vruj' and the 'Password' field is masked with dots. A 'Show Password' checkbox is present. Red annotations highlight the URL and the authentication fields, indicating where to replace the repository ID and the organization/repository account ID and password.

**GET List all DOIs in a repository**

URL: `https://api.test.datacite.org/does?client-id=xuvm.fsydar`

Auth: Basic Auth

Username: vruj

Password: .....

Show Password

Replace with the repository ID

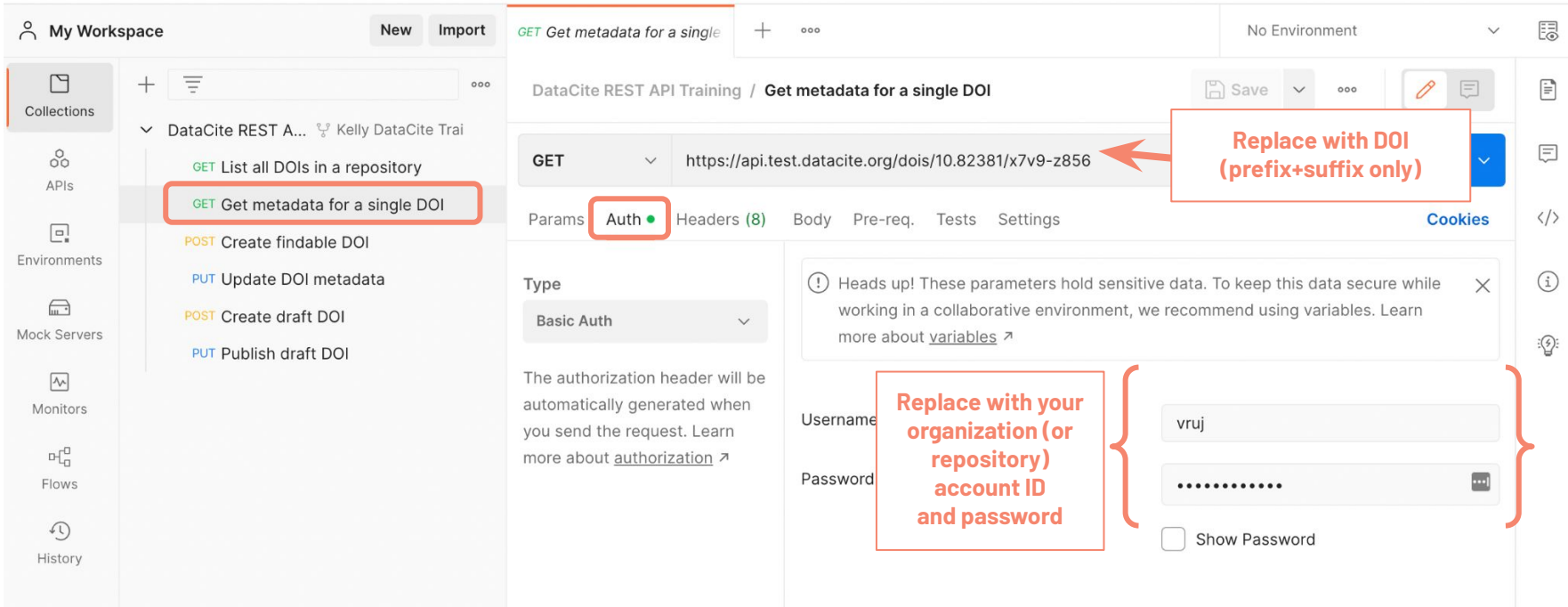
Replace with your organization (or repository) account ID and password

# List all DOIs in a repository

```
curl --user YOUR_ORG_ID:YOUR_ORG_PASSWORD  
https://api.test.datacite.org/does?client-id=REPOSITORY_ID
```

```
curl --user vruj:trainingtest  
https://api.test.datacite.org/does?client-id=xuvm.fsydar
```

# Get metadata for a single DOI



The screenshot displays the DataCite REST API Training interface. On the left, the 'My Workspace' sidebar shows a list of collections, with 'DataCite REST A...' selected. Under this collection, the endpoint 'GET Get metadata for a single DOI' is highlighted with a red box. The main panel shows the configuration for this endpoint. The method is 'GET' and the URL is 'https://api.test.datacite.org/does/10.82381/x7v9-z856'. A red arrow points to the URL with the text 'Replace with DOI (prefix+suffix only)'. The 'Auth' tab is selected, showing 'Basic Auth' as the type. A red box highlights the 'Auth' tab with the text 'Replace with your organization (or repository) account ID and password'. The 'Username' field contains 'vruij' and the 'Password' field is masked with dots. A 'Show Password' checkbox is visible. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)'.

**Replace with DOI (prefix+suffix only)**

**Replace with your organization (or repository) account ID and password**

# Get metadata for a single DOI

```
curl --user YOUR_ORG_ID:YOUR_ORG_PASSWORD  
https://api.test.datacite.org/doi/YOUR_DOI
```

```
curl --user vruj:trainingtest  
https://api.test.datacite.org/doi/10.82381/x7v9-z856
```

# **Creating & updating DOIs with the REST API**

# Create DOI metadata file

## XML Metadata Files

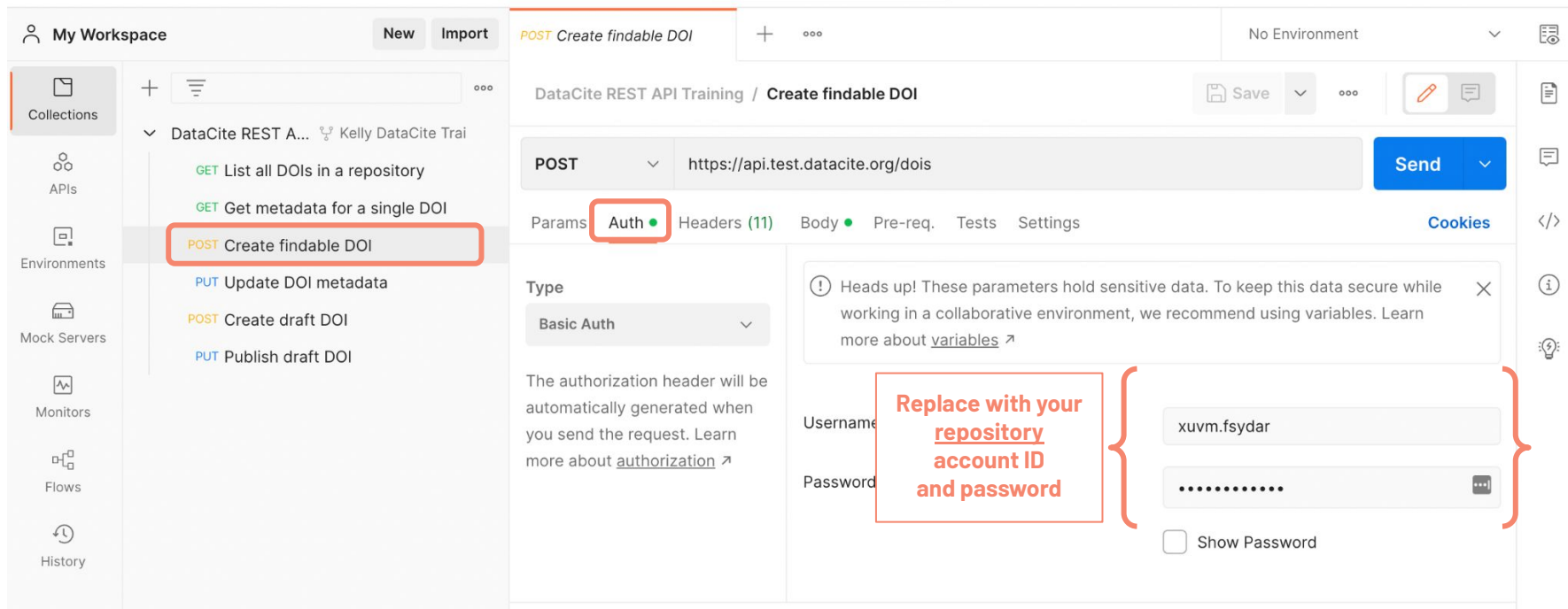
- Can be compared against the schema documentation  
<https://schema.datacite.org>
- Can be validated against a XSD file  
<https://schema.datacite.org/meta/kernel-4/metadata.xsd>
- Historically the only way to specify metadata
- For use with REST API must be a Base64 encoded string in “xml” attribute

## Individual Attributes (JSON) - REST API only

- Can specify individual metadata attributes via the REST API, both creating and updating
- While this closely follows the xml naming schema, there are some naming differences for the benefit of JSON
- Can be parsed easier with new integrations
- Use the API reference:  
[https://support.datacite.org/reference/post\\_dois](https://support.datacite.org/reference/post_dois)



# Create a new DOI



The screenshot displays the DataCite REST API client interface. On the left, the 'My Workspace' sidebar shows a list of collections, with 'DataCite REST A...' expanded. The 'POST Create findable DOI' endpoint is highlighted. The main panel shows the configuration for this endpoint, including the URL 'https://api.test.datacite.org/does' and the 'Auth' tab selected. A red box highlights the 'Auth' tab, and another red box highlights the 'Basic Auth' section, which contains fields for 'Username' and 'Password'. A red callout box with the text 'Replace with your repository account ID and password' points to these fields. The 'Username' field contains the text 'xuvm.fsydar' and the 'Password' field contains a masked password '.....'. A 'Show Password' checkbox is also visible. A warning message at the top right states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)'.

**POST Create findable DOI**

DataCite REST API Training / Create findable DOI

POST <https://api.test.datacite.org/does> Send

Params Auth Headers (11) Body Pre-req. Tests Settings Cookies

Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

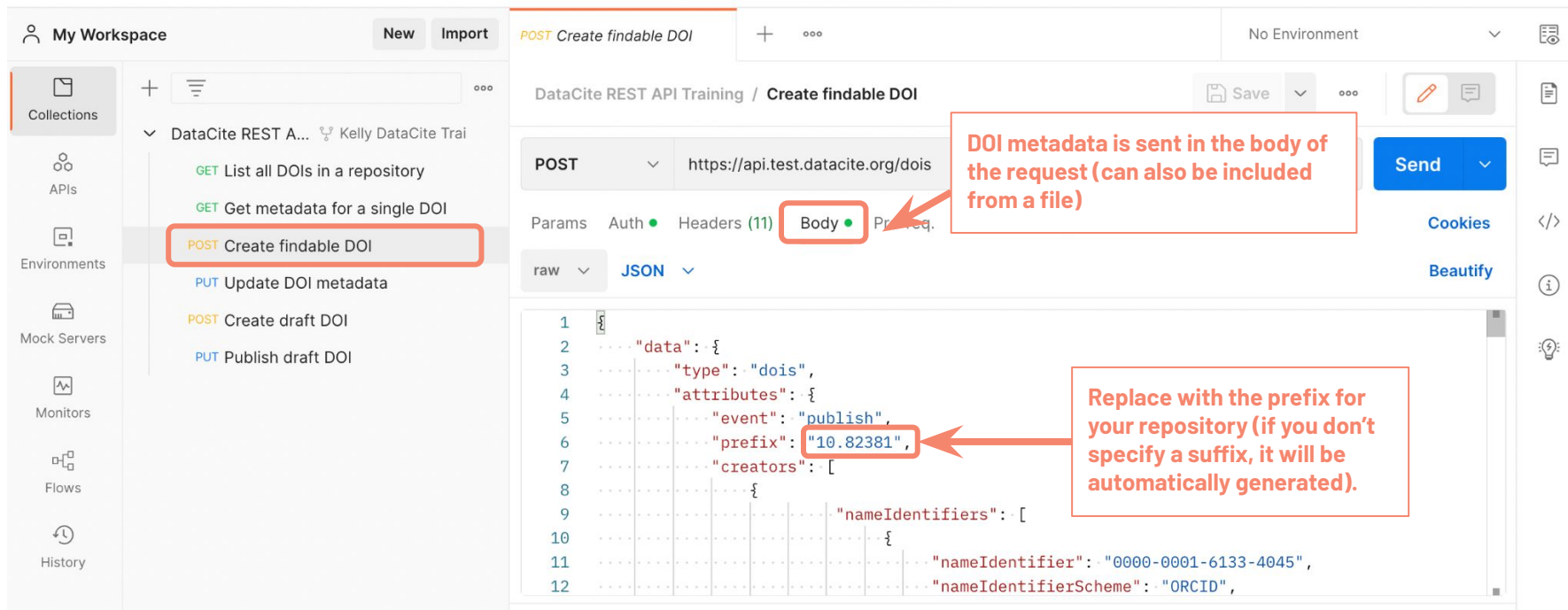
Username: xuvm.fsydar

Password: ..... Show Password

Replace with your repository account ID and password

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

# Create a new DOI



**POST Create findable DOI**

DataCite REST API Training / Create findable DOI

POST `https://api.test.datacite.org/does`

Params Auth Headers (11) **Body** Preview

raw JSON

DOI metadata is sent in the body of the request (can also be included from a file)

```
1 {
2   "data": {
3     "type": "dois",
4     "attributes": {
5       "event": "publish",
6       "prefix": "10.82381",
7       "creators": [
8         {
9           "nameIdentifiers": [
10            {
11              "nameIdentifier": "0000-0001-6133-4045",
12              "nameIdentifierScheme": "ORCID",
```

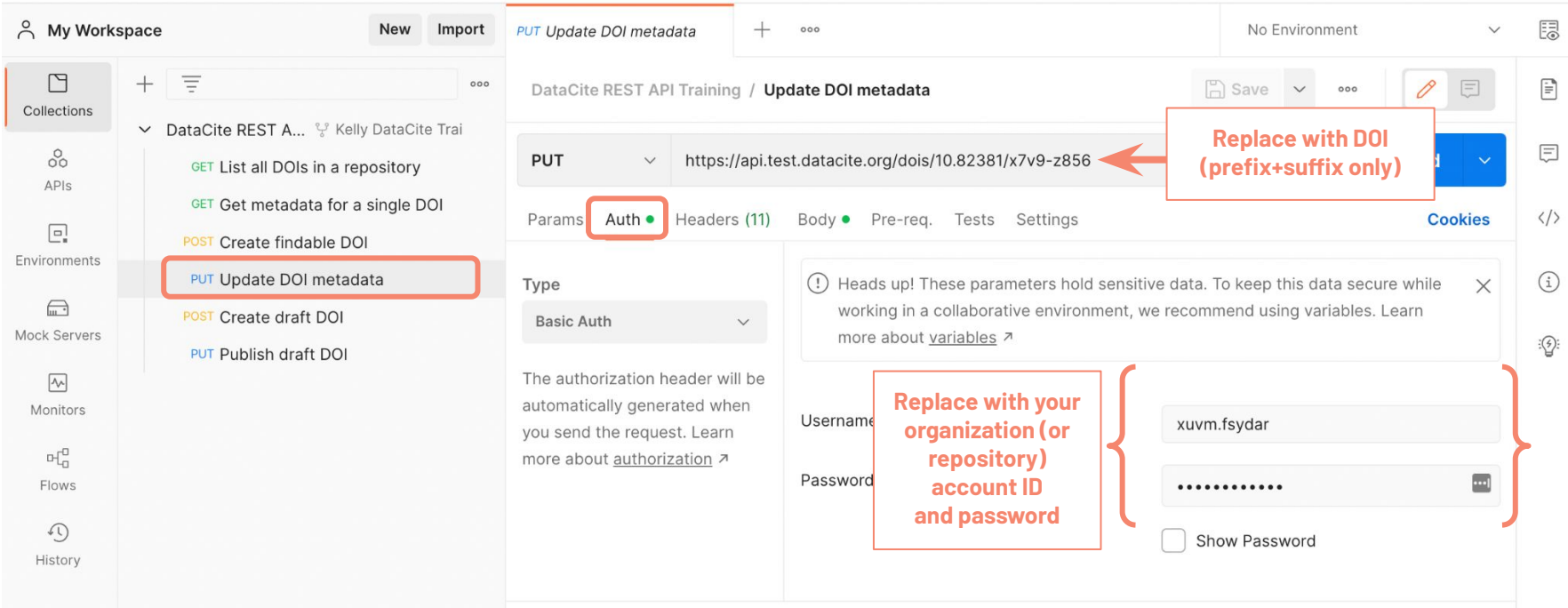
Replace with the prefix for your repository (if you don't specify a suffix, it will be automatically generated).

# Create a new DOI

```
curl -X POST -H "Content-Type: application/vnd.api+json" --user  
YOUR_REPOSITORY_ID:YOUR_REPOSITORY_PASSWORD -d @my_metadata.json  
https://api.test.datacite.org/does
```

```
curl -X POST -H "Content-Type: application/vnd.api+json" --user xuvms.fsydar:trainingtest -d  
'{"data":{"type":"dois","attributes":{"event":"publish","prefix":"10.82381","creators":[{"nameIdentifiers":[{"nameIdentifier":"0000-0001-6133-4045","nameId  
entifierScheme":"ORCID","schemeUri":"https://orcid.org/"},""],"nameType":"Personal","name":"Stathis,  
Kelly","givenName":"Kelly","familyName":"Stathis","affiliation":[{"affiliationIdentifier":"https://ror.org/04wxnsj81","affiliationIdentifierScheme":"ROR","  
name":"DataCite","schemeUri":"https://ror.org/"}]},{"nameIdentifiers":[{"nameIdentifier":"https://orcid.org/0000-0002-4795-7817","nameIdentifierScheme":"OR  
CID","schemeUri":"https://orcid.org/"},""],"nameType":"Personal","name":"Bennett,  
Mike","givenName":"Mike","familyName":"Bennett","affiliation":[{"affiliationIdentifier":"https://ror.org/04wxnsj81","affiliationIdentifierScheme":"ROR","na  
me":"DataCite","schemeUri":"https://ror.org/"}]},{"titles":[{"title":"DataCite API  
Training","lang":"en"}]},{"titleType":"TranslatedTitle","lang":"fr","title":"Formation API DataCite"}]},{"publisher":"DataCite  
e.V."},"publicationYear":2022,"types":{"resourceTypeGeneral":"Text","resourceType":"Presentation"},"subjects":[{"subject":"Application program interfaces  
(Computer  
software)","valueUri":"http://id.loc.gov/authorities/subjects/sh98004527","schemeUri":"https://id.loc.gov/authorities/subjects.html","subjectScheme":"Libra  
ry of Congress Subject Headings  
(LCSH)","lang":"en"}]},{"contributors":[{"nameIdentifiers":[{"nameIdentifier":"0000-0002-6628-8225","nameIdentifierScheme":"ORCID","schemeUri":"https://orcid  
.org/"},""],"affiliation":[{"affiliationIdentifier":"https://ror.org/04wxnsj81","affiliationIdentifierScheme":"ROR","name":"DataCite","schemeUri":"https://ror  
.org/"},""],"name":"Hirsch,  
Mary","givenName":"Mary","familyName":"Hirsch","contributorType":"Other"}]},{"dates":[{"date":"2022-09-22","dateType":"Issued"}]},{"relatedIdentifiers":[{"rela  
tedIdentifierType":"URL","relationType":"IsPartOf","resourceTypeGeneral":"Text","relatedIdentifier":"https://datacite.org/member-meeting-2022.html"}]},{"desc  
riptions":[{"description":"Slides for the 2022 DataCite Member Meeting APIs  
Training","descriptionType":"Abstract","lang":"en"}]},{"geoLocations":[{"geoLocationPoint":{"pointLatitude":"51.5072","pointLongitude":"0.1276"},"geoLocation  
Place":"London,  
UK"}]},{"fundingReferences":[{"funderName":"DataCite","funderIdentifierType":"ROR","funderIdentifier":"https://ror.org/04wxnsj81","awardTitle":"Example Award  
Title","awardNumber":"1234","awardUri":"https://datacite.org/"},""],"language":"en","url":"https://docs.google.com/presentation/d/1SqjmhKhJJi5anF1Cmr3mVO2GqN9  
k_KXnx1JWKPTQrAA/","schemaVersion":"http://datacite.org/schema/kernel-4"}]}' https://api.test.datacite.org/does
```

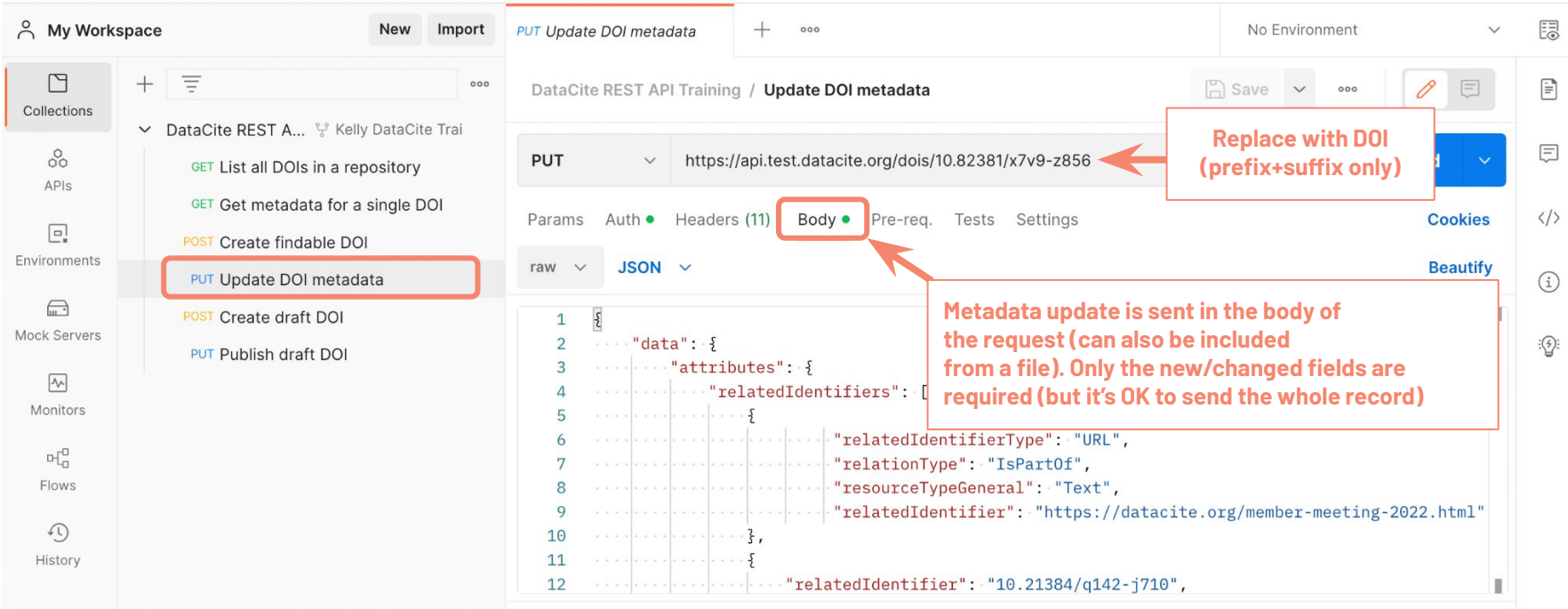
# Update an existing DOI's metadata



The screenshot displays the 'PUT Update DOI metadata' endpoint configuration in the DataCite REST API Training interface. The interface is divided into several sections:

- My Workspace:** Contains a sidebar with 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The 'DataCite REST API Training' collection is expanded, showing a list of endpoints. The 'PUT Update DOI metadata' endpoint is highlighted with a red box.
- Endpoint Configuration:** The 'PUT' method is selected, and the URL is 'https://api.test.datacite.org/doi/10.82381/x7v9-z856'. A red arrow points to the URL with the text 'Replace with DOI (prefix+suffix only)'.
- Auth:** The 'Auth' tab is selected and highlighted with a red box. The 'Basic Auth' type is chosen. A red box highlights the 'Auth' tab with the text 'Replace with your organization (or repository) account ID and password'.
- Username and Password:** The 'Username' field contains 'xuvvm.fsydar' and the 'Password' field contains a masked password. A red bracket groups these fields with the text 'Replace with your organization (or repository) account ID and password'.
- Warning:** A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)'.

# Update an existing DOI's metadata



The screenshot displays the DataCite REST API client interface. On the left, the 'My Workspace' sidebar shows a list of API endpoints under 'DataCite REST A...'. The 'PUT Update DOI metadata' endpoint is highlighted with a red box. The main panel shows the details of this PUT request, including the URL 'https://api.test.datacite.org/doi/10.82381/x7v9-z856' and the 'Body' tab selected. A red box highlights the 'Body' tab, and a red arrow points to the URL with the text 'Replace with DOI (prefix+suffix only)'. The request body is shown in JSON format, containing metadata for a DOI. A red box highlights the JSON body, and a red arrow points to it with the text 'Metadata update is sent in the body of the request (can also be included from a file). Only the new/changed fields are required (but it's OK to send the whole record)'.

**PUT Update DOI metadata**

https://api.test.datacite.org/doi/10.82381/x7v9-z856

Replace with DOI (prefix+suffix only)

Body

Metadata update is sent in the body of the request (can also be included from a file). Only the new/changed fields are required (but it's OK to send the whole record)

```
1 {
2   "data": {
3     "attributes": {
4       "relatedIdentifiers": [
5         {
6           "relatedIdentifierType": "URL",
7           "relationType": "IsPartOf",
8           "resourceTypeGeneral": "Text",
9           "relatedIdentifier": "https://datacite.org/member-meeting-2022.html"
10        },
11        {
12          "relatedIdentifier": "10.21384/q142-j710",
```

# Update an existing DOI's metadata

```
curl -X PUT -H "Content-Type: application/vnd.api+json" --user  
YOUR_REPOSITORY_ID:YOUR_REPOSITORY_PASSWORD -d  
@my_doi_update.json https://api.test.datacite.org/does/YOUR_DOI
```

```
curl -X PUT -H "Content-Type: application/vnd.api+json" --user  
xuvvm.fsydar:trainingtest -d  
'{"data":{"attributes":{"relatedIdentifiers":[{"relatedIdentifierType":"URL",  
relationType":"IsPartOf","resourceTypeGeneral":"Text","relatedIdentifier":"htt  
ps://datacite.org/member-meeting-2022.html"}, {"relatedIdentifier":"10.21384/q1  
42-j710","relatedIdentifierType":"DOI","relationType":"IsDerivedFrom","resourc  
eTypeGeneral":"Dataset"}]}}}'  
https://api.test.datacite.org/does/DOI_CREATED_IN_PREVIOUS_STEP
```

# Resources

# API support resources

- REST API
  - Documentation: <https://support.datacite.org/docs/api>
  - Reference: <https://support.datacite.org/reference>
- MDS API
  - Documentation: <https://support.datacite.org/docs/mds-api-guide>
  - Reference: <https://support.datacite.org/reference/overview>
- API FAQs
  - <https://support.datacite.org/docs/apis>
- Metadata schema documentation
  - Schema website: <https://schema.datacite.org/>





CONNECTING RESEARCH,  
IDENTIFYING KNOWLEDGE



[info@datacite.org](mailto:info@datacite.org)



[pidforum.org](http://pidforum.org)



[datacite.org](http://datacite.org)  
[blog.datacite.org](http://blog.datacite.org)



[support.datacite.org](http://support.datacite.org)  
[support@datacite.org](mailto:support@datacite.org)



[@datacite](https://twitter.com/datacite)



[DataCite](https://www.youtube.com/DataCite)



[@datacite](https://www.linkedin.com/company/datacite)