THE PEAK OF DATA
INTEGRATION
2022UC

FME Server REST API

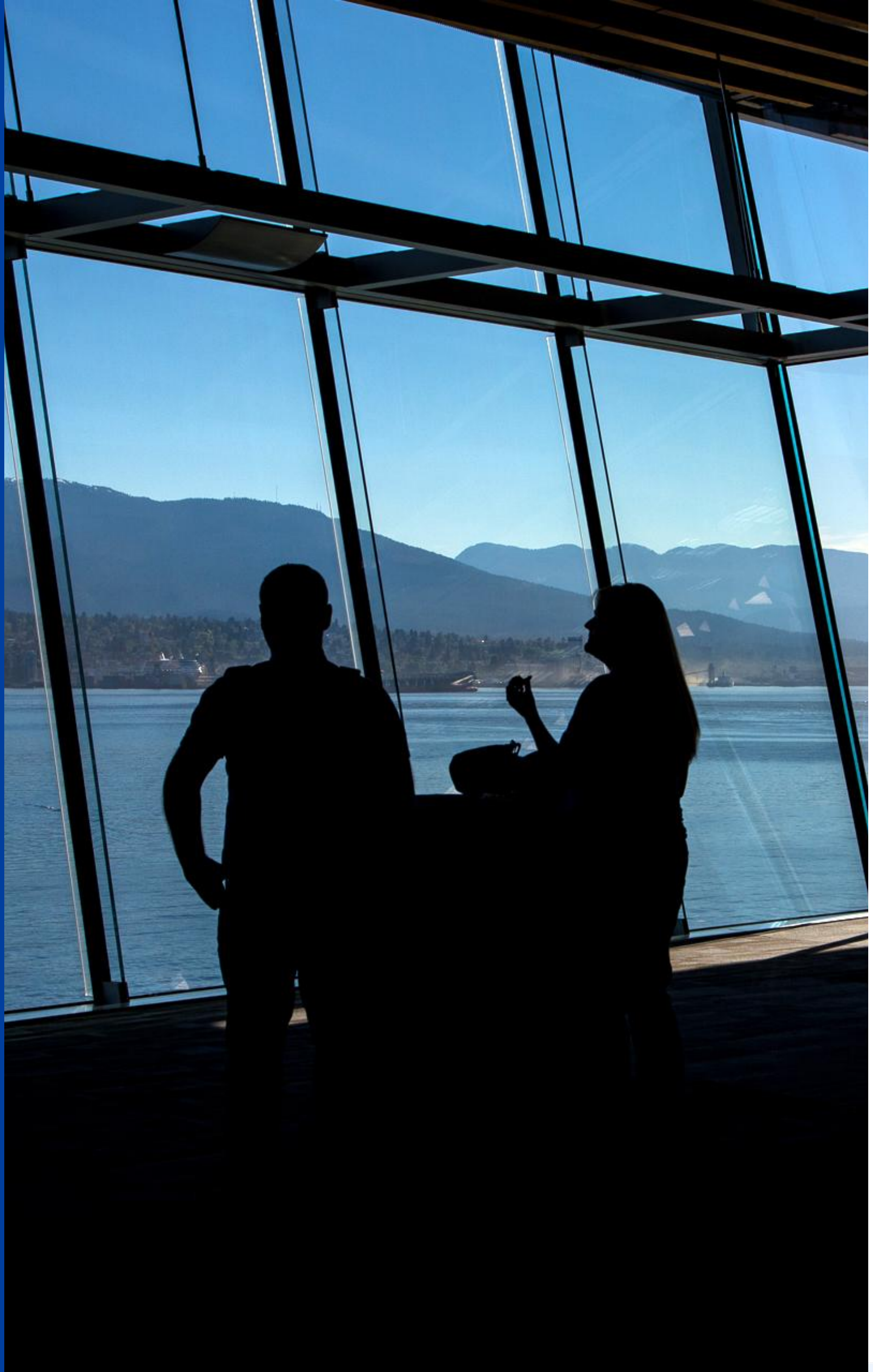
SAFE SOFTWARE®

Presenters

Sienna Emery
Technical Support Specialist, Customer Solutions

Sanae Mendoza
Technical Support Specialist, FME Server

Kezia Yu
Technical Support Specialist, FME Server



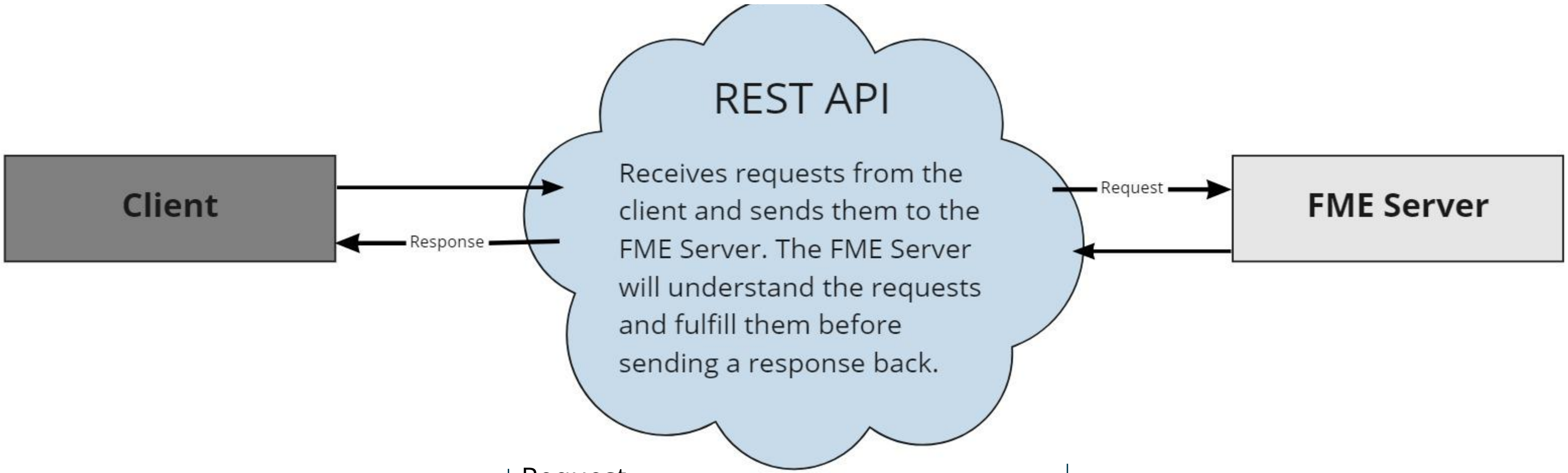
Agenda

- Introduction
- Authorization in the FME Server REST API and Token Management
- Submit a Job from a Third-Party Application
- Create a Job History Report
- Resources & QA

REST API

REpresentational **S**tate **T**ransfer
Application **P**rogramming **I**nterface

Components of an FME Server API Request



Receives requests from the client and sends them to the FME Server. The FME Server will understand the requests and fulfill them before sending a response back.

Requests are made through URLs

Request

- **Method:** GET, POST, PUT, or DELETE
- **URL:** http://<yourServerHost>/fmerest/v3/
- **Parameters**
- **Headers**
- **Body**

Response messages are returned in JSON format

What is an API?

APIs can be used to retrieve, update, or delete data over the web.

Actions can be performed programmatically to easily automate workflows.

APIs can also allow for communication between applications and users or for easy integrations.

What can I do with the FME Server REST API?

- Sending, receiving and modifying notifications
- **Running jobs and viewing job history**
- Managing user accounts, roles, and policies
- Licensing FME Server
- **Performing backup and restore operations**
- Managing cleanup tasks
- **Scheduling jobs**
- Managing user settings and favorites
- **Uploading and managing file resources**
- Publishing workspaces and managing published parameters

How to interact with REST API?

Key API Terminology

REST: AN API design style defined by a uniform structure that allows us to interact with resources via a URL.

JSON: A common data format used for exchanging data over the web

Request: (or “call”) A URL that asks to perform an action on a resource

Response: The information that is returned

Resource: The data, information, files, or items we are interested in

Request Terminology

Endpoint: Part of the URL that specifies the resource location

Method: Determines the action the request is making

Parameters: Options that further specify location, filters, state, or content

e.g. *setting values for FME user parameters*

Headers: Supply context or instructions with the request

e.g. *What format is being sent? What can be sent back?*

Body: Data (“payload”) to be created, modified, deleted or returned by the request.

Request Methods

Method	Action	Uses in FME Server
POST	Create	Create a publication, project, notification
GET	Read	Perform a health check, get information on the server, get notification on the publication
PUT	Update and Replace	Replace a publication, update a project, update user information
DELETE	Delete	Delete topics, users, roles

API documentation

Everything you need to know to build a request

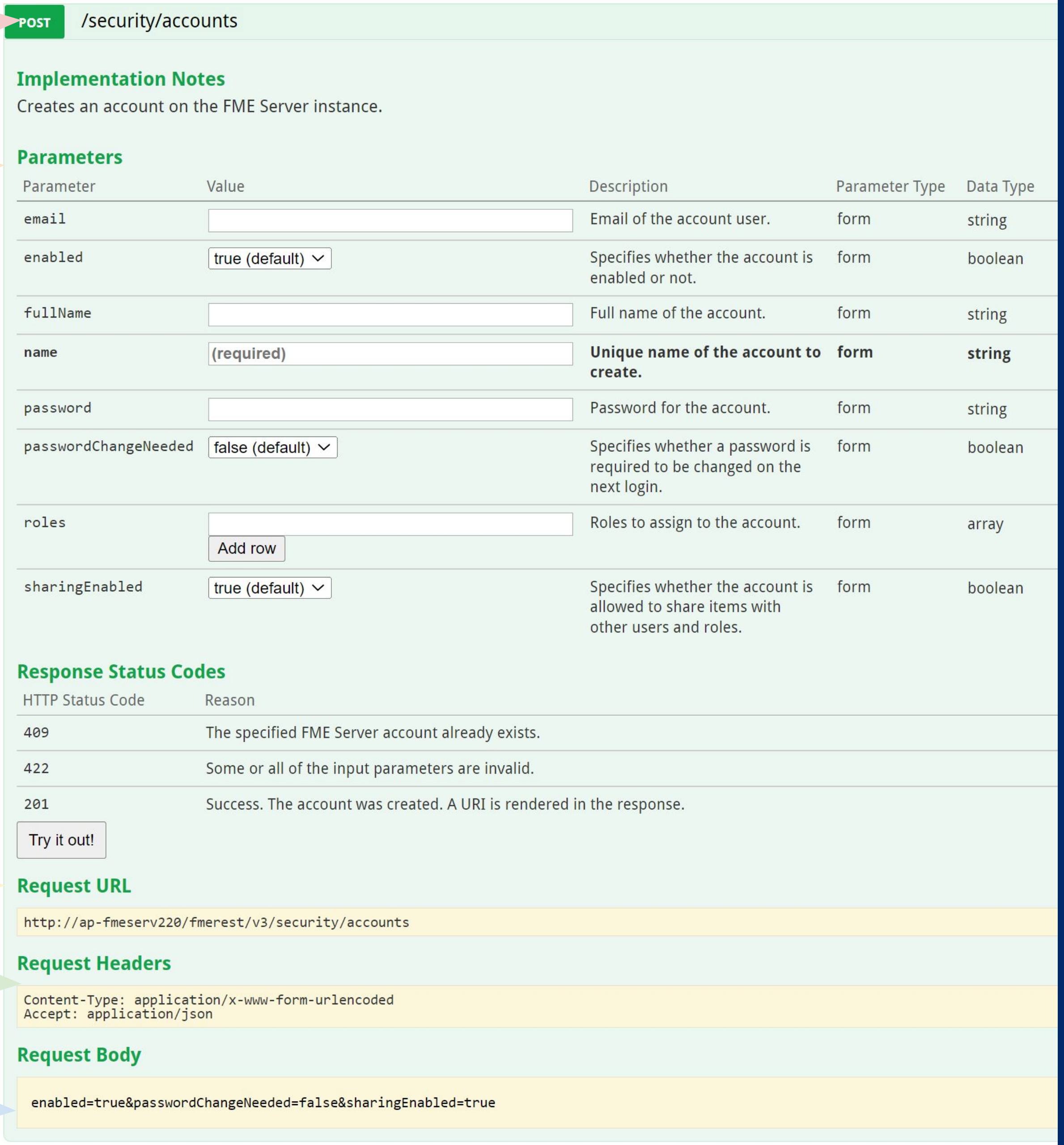
Method

Parameters

URL

Headers

Body



POST /security/accounts

Implementation Notes

Creates an account on the FME Server instance.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
email		Email of the account user.	form	string
enabled	true (default)	Specifies whether the account is enabled or not.	form	boolean
fullName		Full name of the account.	form	string
name	(required)	**Unique name of the account to create.**	**form**	**string**
password		Password for the account.	form	string
passwordChangeNeeded	false (default)	Specifies whether a password is required to be changed on the next login.	form	boolean
roles	Add row	Roles to assign to the account.	form	array
sharingEnabled	true (default)	Specifies whether the account is allowed to share items with other users and roles.	form	boolean

Response Status Codes

HTTP Status Code	Reason
409	The specified FME Server account already exists.
422	Some or all of the input parameters are invalid.
201	Success. The account was created. A URI is rendered in the response.

Try it out!

Request URL

```
http://ap-fmeserv220/fmerest/v3/security/accounts
```

Request Headers

```
Content-Type: application/x-www-form-urlencoded
Accept: application/json
```

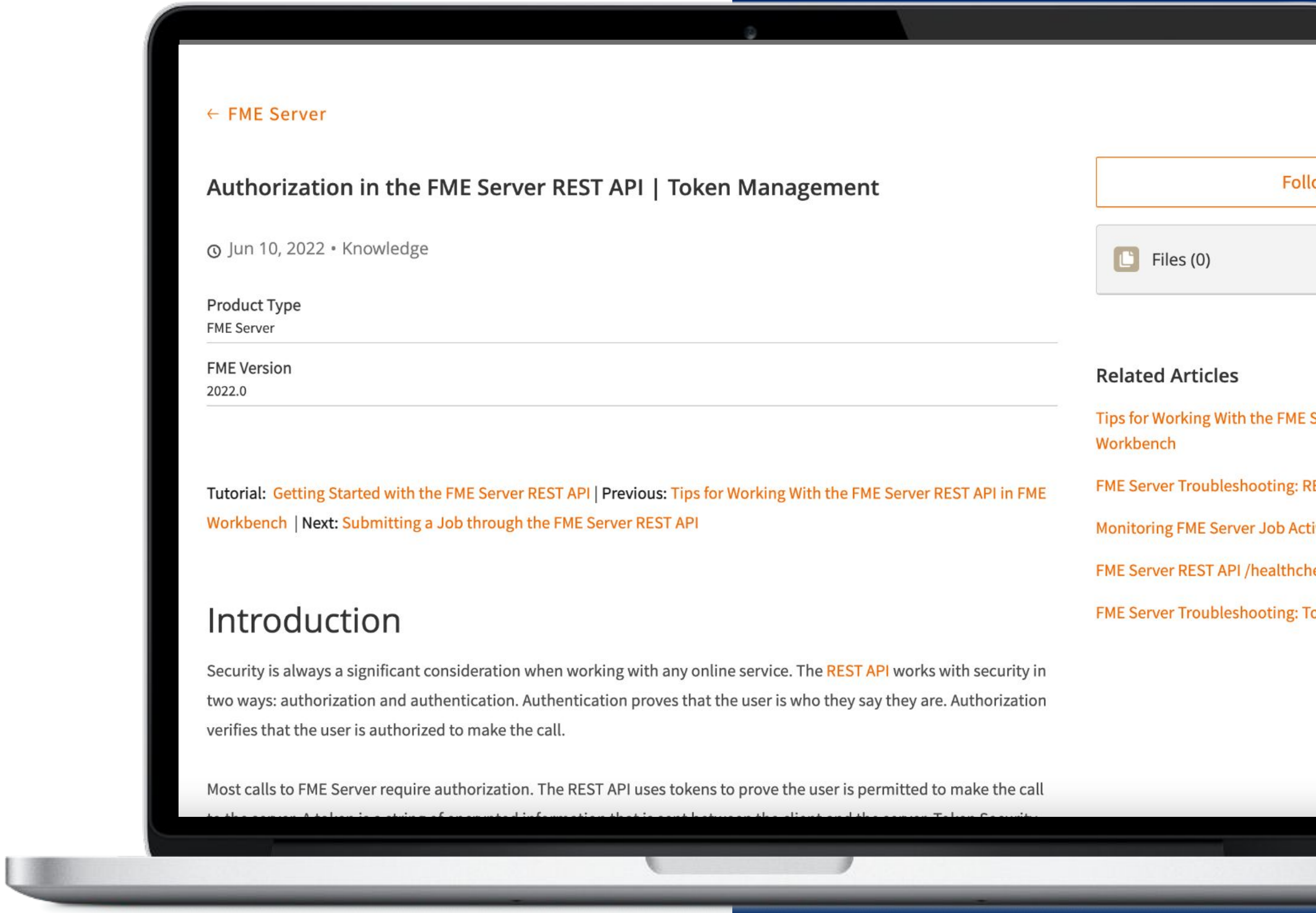
Request Body

```
enabled=true&passwordChangeNeeded=false&sharingEnabled=true
```

FME Server REST API: *Create an account*

Authorization in the FME Server REST API | Token Management

Exercise 1
Authorization in FME Server



← FME Server

Authorization in the FME Server REST API | Token Management

Jun 10, 2022 • Knowledge

Product Type
FME Server

FME Version
2022.0

Tutorial: Getting Started with the FME Server REST API | Previous: Tips for Working With the FME Server REST API in FME Workbench | Next: Submitting a Job through the FME Server REST API

Introduction

Security is always a significant consideration when working with any online service. The REST API works with security in two ways: authorization and authentication. Authentication proves that the user is who they say they are. Authorization verifies that the user is authorized to make the call.

Most calls to FME Server require authorization. The REST API uses tokens to prove the user is permitted to make the call

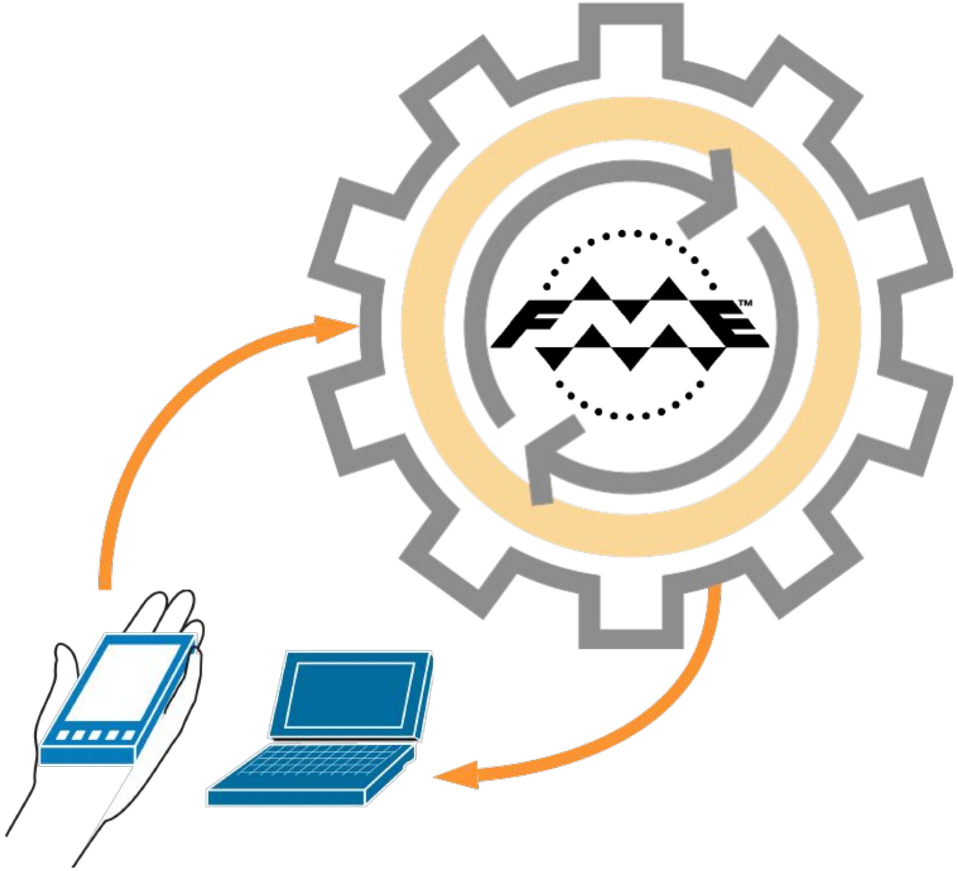
Follo
Files (0)
Related Articles
Tips for Working With the FME S
Workbench
FME Server Troubleshooting: RE
Monitoring FME Server Job Acti
FME Server REST API /healthche
FME Server Troubleshooting: To

Submitting a Job from a Third Party Application

Running Workflows from Anywhere

The FME Server REST API allows **systems and users** to submit jobs that **exchange and transform data** on-demand.

- Collect data from the field
- Power web applications
- Integrate with custom code
- Create tools for non-FME users
- Automate workflows between applications



There are multiple ways to submit jobs over the web to FME Server…

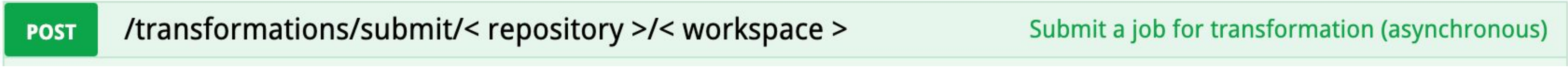
FME Server REST API

- REST Service
- POST method
- Runtime data is sent in the request *body*

Webhook URLs

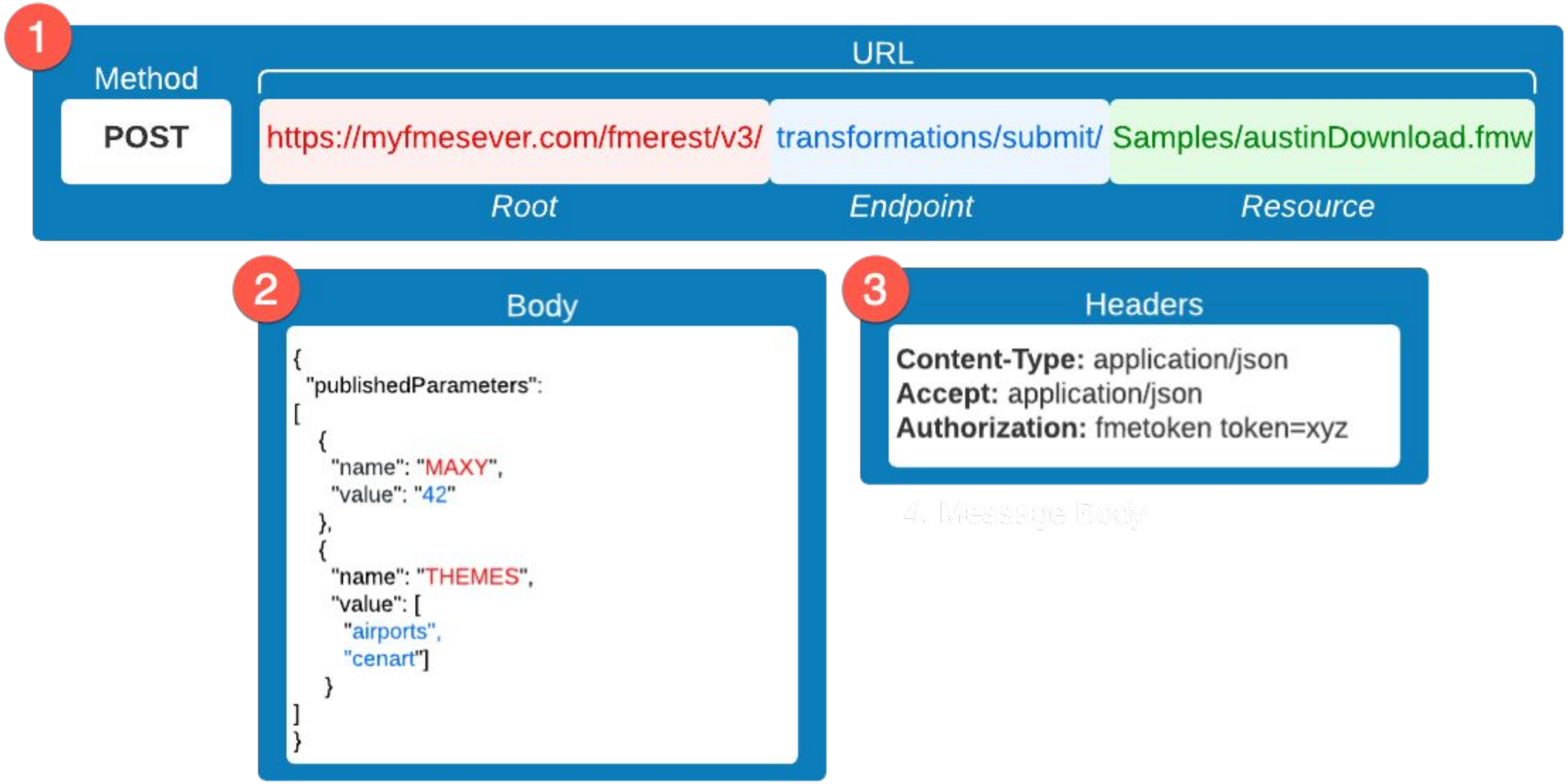
- Transformation Services (Data Download, Data streaming, etc.)
- GET or POST method
- Runtime data is sent in the request *URL parameters*

Building an API Job Request in 1, 2, 3…



/transformations/submit/< repository >/< workspace >

Submit a job for transformation (asynchronous)

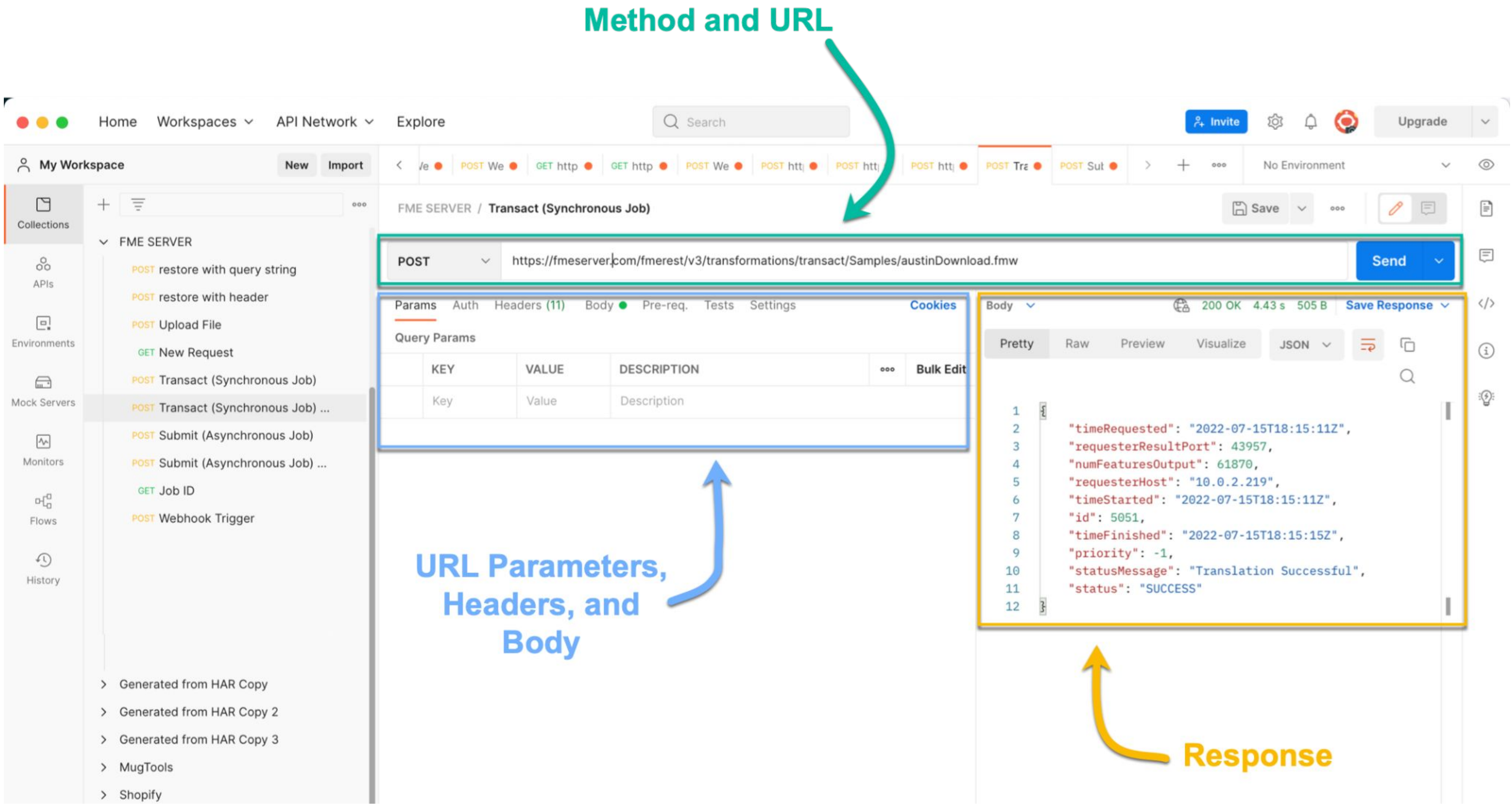


POST https://myfmesever.com/fmerest/v3/transformations/submit/Samples/austinDownload.fmw

```
{
 "publishedParameters":
[
  {
   "name": "MAXY",
   "value": "42"
  },
  {
   "name": "THEMES",
   "value": [
     "airports",
     "cenart"]
   }
]
}
```

Content-Type: application/json
Accept: application/json
Authorization: fmetoken token=xyz

Sending Requests from Postman



Method and URL

FME SERVER / Transact (Synchronous Job)

POST https://fmeserver.com/fmerest/v3/transformations/transact/Samples/austinDownload.fmw

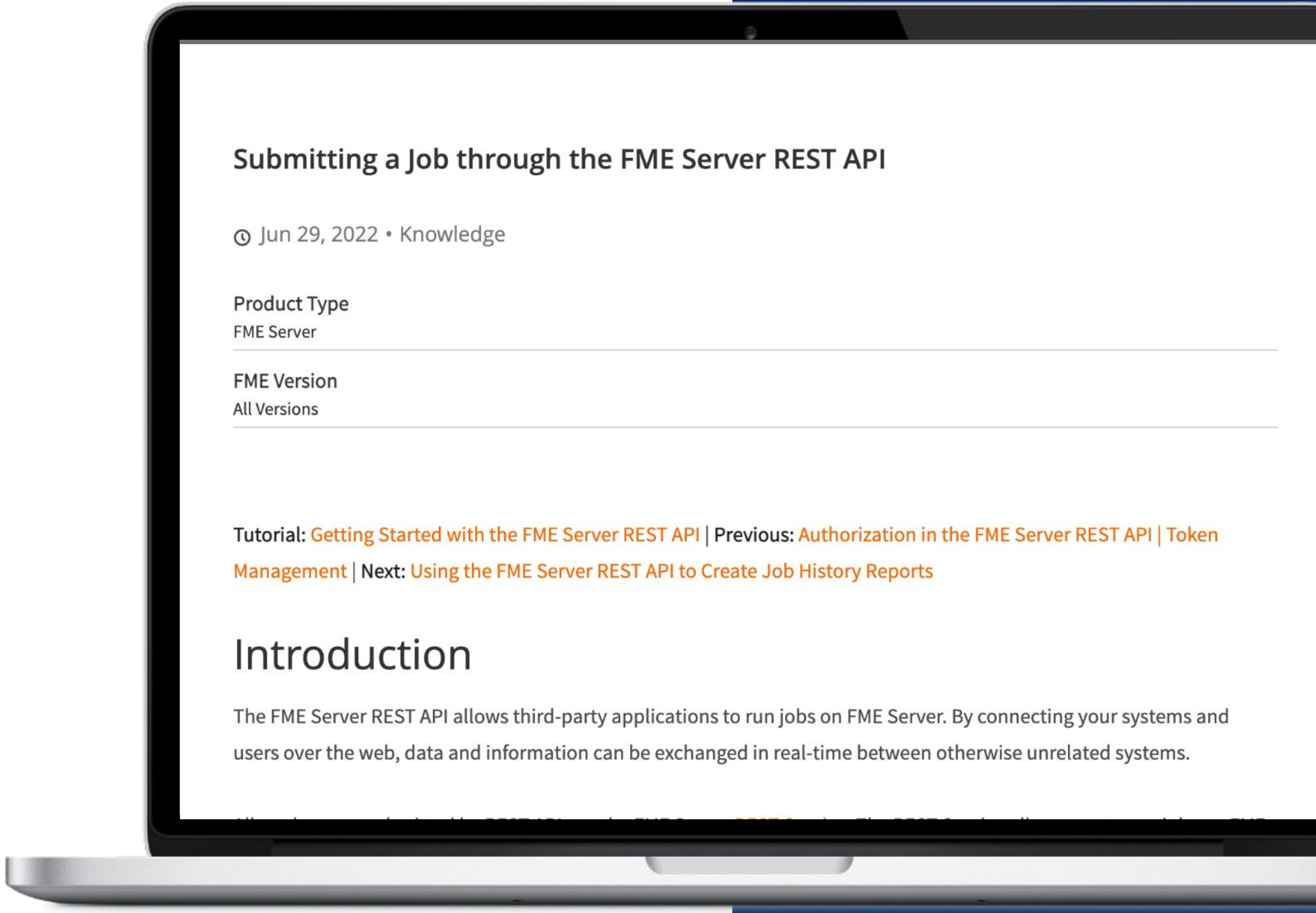
URL Parameters, Headers, and Body

Response

```
{
    "timeRequested": "2022-07-15T18:15:11Z",
    "requesterResultPort": 43957,
    "numFeaturesOutput": 61870,
    "requesterHost": "10.0.2.219",
    "timeStarted": "2022-07-15T18:15:11Z",
    "id": 5051,
    "timeFinished": "2022-07-15T18:15:15Z",
    "priority": -1,
    "statusMessage": "Translation Successful",
    "status": "SUCCESS"
}
```

Exercise 2

Submitting a Job through the FME Server REST API



Submitting a Job through the FME Server REST API

Jun 29, 2022 • Knowledge

Product Type
FME Server

FME Version
All Versions

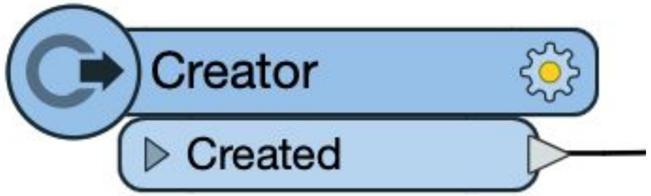
Tutorial: Getting Started with the FME Server REST API | Previous: Authorization in the FME Server REST API | Token Management | Next: Using the FME Server REST API to Create Job History Reports

Introduction

The FME Server REST API allows third-party applications to run jobs on FME Server. By connecting your systems and users over the web, data and information can be exchanged in real-time between otherwise unrelated systems.

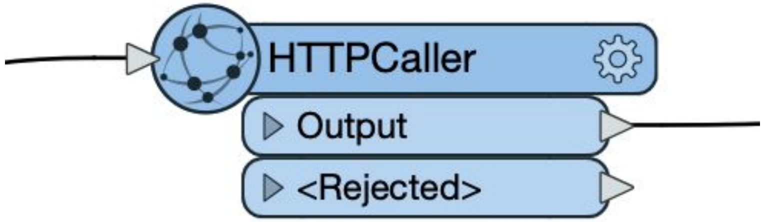
APIs in FME Desktop

Powerful API Transformers

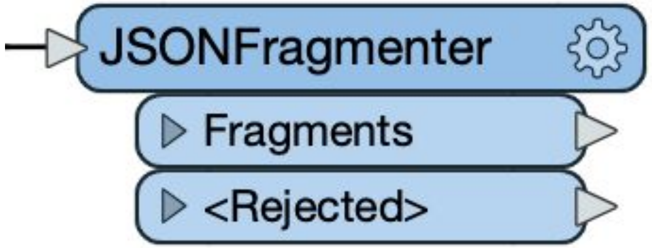


Typically most workflows in FME begin with a Reader that initiates the workflow.

However, since most APIs use the HTTPCaller as the Reader. The Creator is used to kick off the workflow.

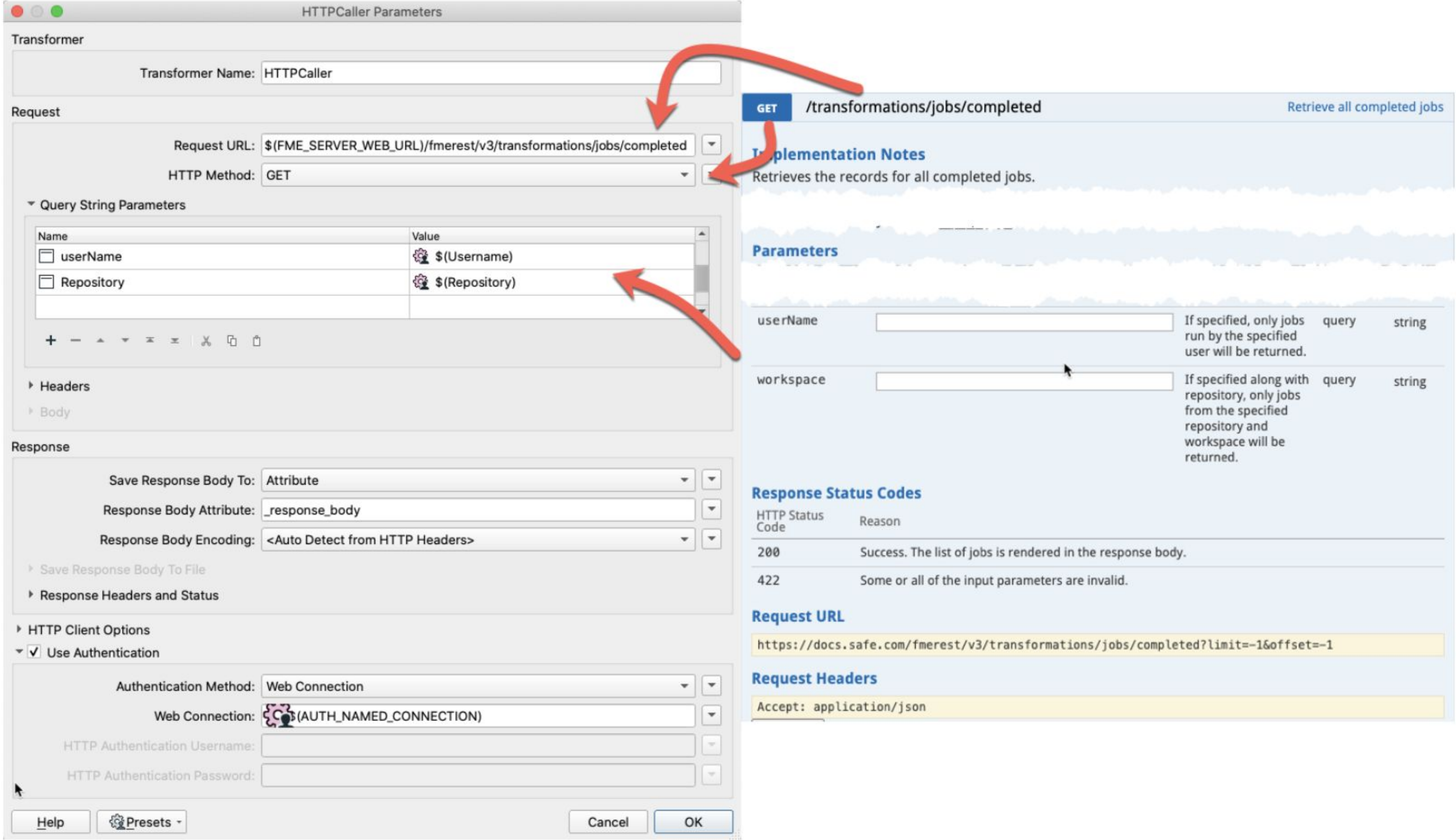


The HTTPCaller is where the call to the REST API is handled.



Most REST APIs return JSON. JSON can be parsed in the JSON fragmenter.

https://community.safe.com/s/article/Tips-for-Working-With-the-FME-Server-REST-API-in-FME-Workbench



HTTPCaller Parameters

Transformer

Transformer Name: HTTPCaller

Request

Request URL: $(FME_SERVER_WEB_URL)/fmerest/v3/transformations/jobs/completed

HTTP Method: GET

Query String Parameters

Name	Value
userName	$(Username)
Repository	$(Repository)

Headers

Body

Response

Save Response Body To: Attribute

Response Body Attribute: _response_body

Response Body Encoding: <Auto Detect from HTTP Headers>

Save Response Body To File

Response Headers and Status

HTTP Client Options

☑ Use Authentication

Authentication Method: Web Connection

Web Connection: $(AUTH_NAMED_CONNECTION)

HTTP Authentication Username:

HTTP Authentication Password:

GET /transformations/jobs/completed Retrieve all completed jobs

Implementation Notes

Retrieves the records for all completed jobs.

Parameters

userName		If specified, only jobs run by the specified user will be returned.	query	string
workspace		If specified along with repository, only jobs from the specified repository and workspace will be returned.	query	string

Response Status Codes

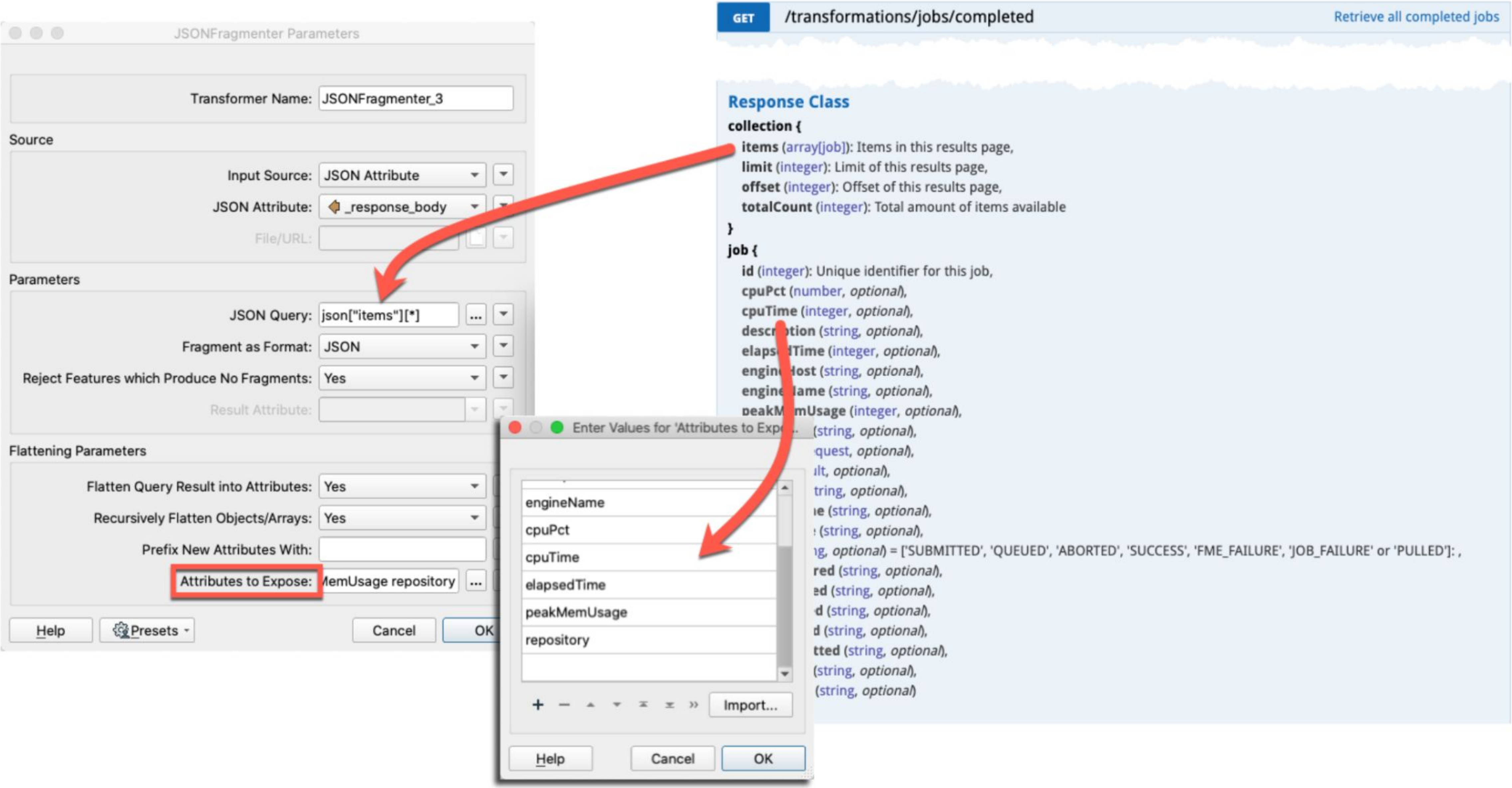
HTTP Status Code	Reason
200	Success. The list of jobs is rendered in the response body.
422	Some or all of the input parameters are invalid.

Request URL

```
https://docs.safe.com/fmerest/v3/transformations/jobs/completed?limit=-1&offset=-1
```

Request Headers

```
Accept: application/json
```



GET /transformations/jobs/completed

Retrieve all completed jobs

Response Class

```
collection {
  items (array[job]): Items in this results page,
  limit (integer): Limit of this results page,
  offset (integer): Offset of this results page,
  totalCount (integer): Total amount of items available
}
job {
  id (integer): Unique identifier for this job,
  cpuPct (number, optional),
  cpuTime (integer, optional),
  description (string, optional),
  elapsedTime (integer, optional),
  engineHost (string, optional),
  engineName (string, optional),
  peakMemUsage (integer, optional),
  (string, optional),
  quest, optional),
  ılt, optional),
  tring, optional),
  ie (string, optional),
  : (string, optional),
  ig, optional) = ['SUBMITTED', 'QUEUED', 'ABORTED', 'SUCCESS', 'FME_FAILURE', 'JOB_FAILURE' or 'PULLED']: ,
  red (string, optional),
  ed (string, optional),
  d (string, optional),
  d (string, optional),
  tted (string, optional),
  (string, optional),
  (string, optional)
```

JSONFragmenter Parameters

Transformer Name: JSONFragmenter_3

Source

Input Source: JSON Attribute

JSON Attribute: _response_body

File/URL:

Parameters

JSON Query: json["items"][*]

Fragment as Format: JSON

Reject Features which Produce No Fragments: Yes

Result Attribute:

Flattening Parameters

Flatten Query Result into Attributes: Yes

Recursively Flatten Objects/Arrays: Yes

Prefix New Attributes With:

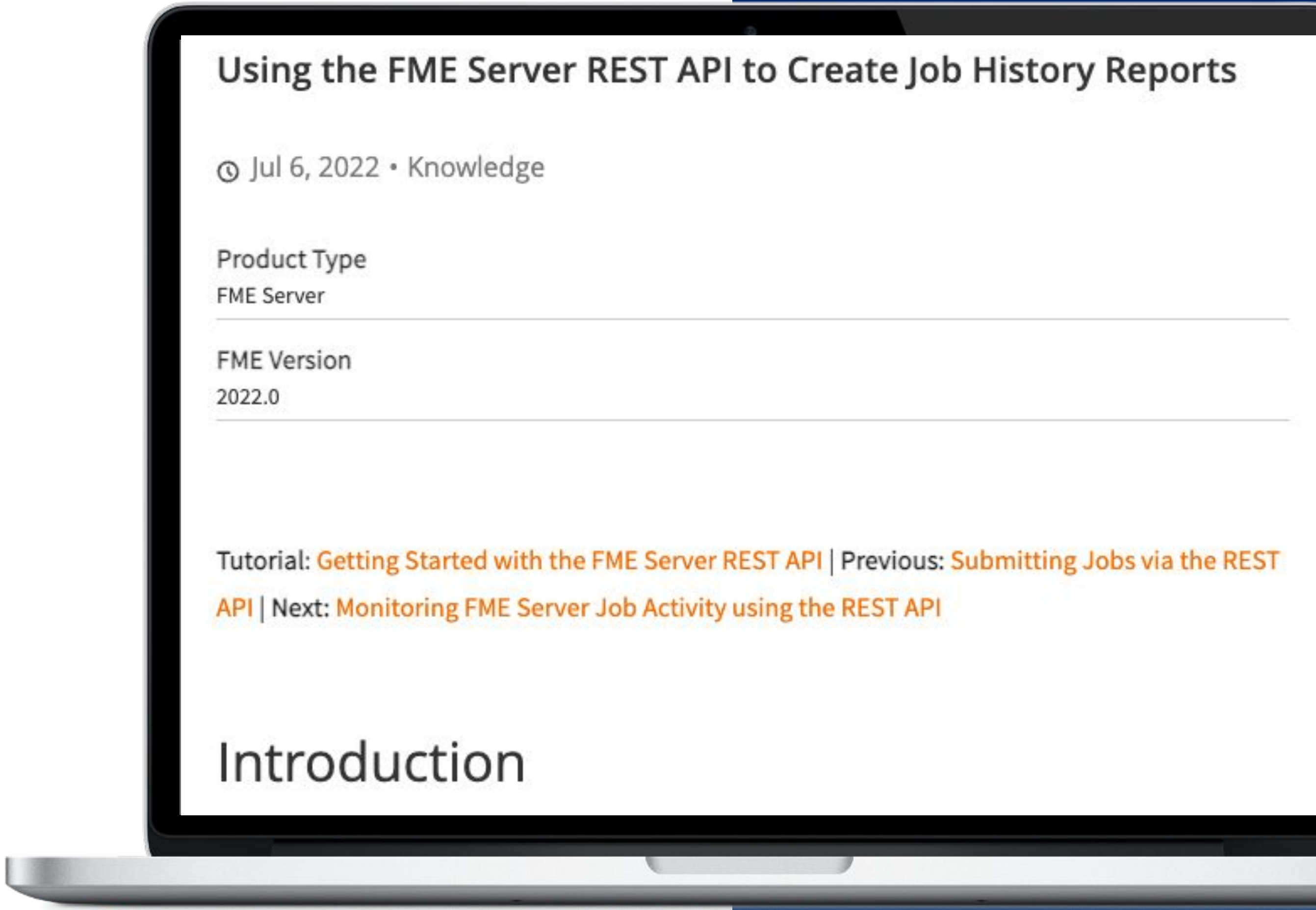
Attributes to Expose: lemUsage repository

Enter Values for 'Attributes to Expo...

- engineName
- cpuPct
- cpuTime
- elapsedTime
- peakMemUsage
- repository

Activity: Creating a Job History Report

Exercise 3
Using the FME Server REST API to Create Job History Reports



Using the FME Server REST API to Create Job History Reports

Jul 6, 2022 • Knowledge

Product Type
FME Server

FME Version
2022.0

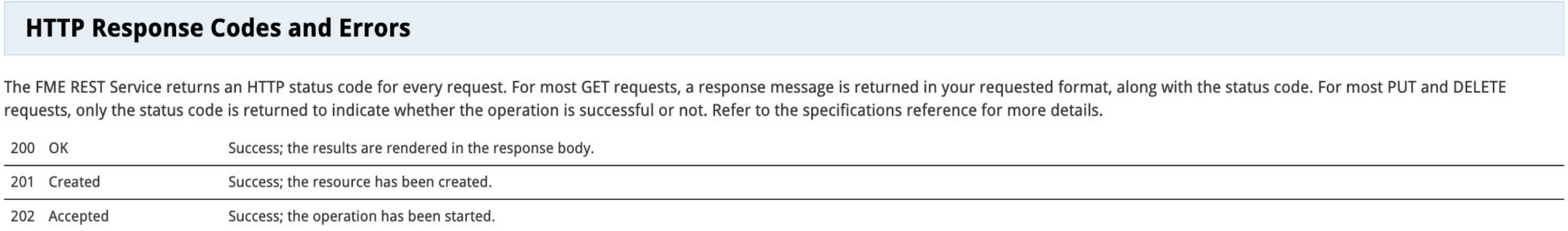
Tutorial: Getting Started with the FME Server REST API | Previous: Submitting Jobs via the REST API | Next: Monitoring FME Server Job Activity using the REST API

Introduction

Troubleshooting

Stuck? Try the following things:

- Test the call in Postman
- Review the error code. Cross reference it with the FME Server REST API Documentation.
- Ensure that the token used is valid. If you are having permissions issues, try granting more privileges temporarily to test.



HTTP Response Codes and Errors

The FME REST Service returns an HTTP status code for every request. For most GET requests, a response message is returned in your requested format, along with the status code. For most PUT and DELETE requests, only the status code is returned to indicate whether the operation is successful or not. Refer to the specifications reference for more details.

200	OK	Success; the results are rendered in the response body.
201	Created	Success; the resource has been created.
202	Accepted	Success; the operation has been started.

Resources

REST API Documentation

Getting Started with the FME Server REST API

Connect to APIs and Webhooks in no time [blog]

THE PEAK OF DATA INTEGRATION
2022 UC

Thank You!

sienna.emery@safe.com

sanae.mendoza@safe.com

kezia.yu@safe.com

SAFE SOFTWARE®