Mockiato Quick Start Guide

[Mockiato Quick Start Guide 1](#_Toc6908878)

[What is Mockiato? 2](#_Toc6908879)

[**Architecture** 3](#_Toc6908880)

[**Data Models** 4](#_Toc6908881)

[**Getting Started** 5](#_Toc6908882)

[Groups 5](#_Toc6908883)

[Owner 5](#_Toc6908884)

[Important Links 5](#_Toc6908885)

[Access to Mockiato 5](#_Toc6908886)

[**Admin Panel** 6](#_Toc6908887)

[Creating a group in Mockiato 6](#_Toc6908888)

[Add User to Group 6](#_Toc6908889)

[**Create A New Service** 7](#_Toc6908890)

[Create a SOAP service using a request/response pair 7](#_Toc6908891)

[Create a REST service using a request/response pair 8](#_Toc6908892)

[Create an MQ service using a request/response pair 9](#_Toc6908893)

[Recording 10](#_Toc6908894)

[Live Invocation 12](#_Toc6908895)

[Bulk upload 13](#_Toc6908896)

[OPEN API 14](#_Toc6908897)

[WSDL 15](#_Toc6908898)

[**Existing Service Options** 16](#_Toc6908899)

[Update 16](#_Toc6908900)

[Export 16](#_Toc6908901)

[Stop 16](#_Toc6908902)

[Delete 16](#_Toc6908903)

[Restore 17](#_Toc6908904)

[**Match Templates** 18](#_Toc6908905)

[**Troubleshooting** 19](#_Toc6908906)

## What is Mockiato?

Mockiato is a web-based platform for API virtualization. Mockiato was developed at Optum to enable test automation, and can simulate REST APIs, SOAP services, and message-oriented middleware.

Mockiato can generate realistic data for testing, and export it to JSON, XML, or CSV. Mockiato creates virtual endpoints that simulate your production APIs. These virtual services are ideal for testing, sandboxing, knowledge transfer, and driving rapid development.

Mockiato is built on open-source technologies like Node.js and MongoDB, and was designed API-first with cloud readiness in mind. It exposes a RESTful API to programmatically interact with your services, as well as a modern web interface and command-line client.

**Architecture**

Mockiato is comprised of 3 basic architectural components: a web-based user interface, a REST API for managing services, and a Mongo database.

**Web UI**

Mockiato provides a simple, intuitive interface for managing virtual services. Built on AngularJS, this single-page application acts as a client to a Mockiato server.

**REST API**

In Mockiato, virtual services are considered resources. A REST API is exposed to facilitate CRUD (create, read, update, delete) operations on these resources.

For more information on the methods available in the API, please see our Swagger documentation.

**NoSQL**

All of the data that comprises a virtual service (base path, request data, response data, etc.) is stored in a Mongo database.

Please see the next section for more information on the data models behind Mockiato.

**Data Models**

Mockiato structures data according to 4 basic models: a service, a request / response pair, a group, and an owner.

**Service**

The service model is the primary entity, and the remaining 3 are sub-components of it. The service is comprised of a base path, type (e.g. SOAP, REST), name, owner, group, and a set of request / response pairs.

**RR Pair**

A request / response pair holds all information necessary for request matching (headers, status codes, HTTP methods, relative paths, request bodies, response bodies, etc.) At least one RR pair should be associated with a service for matching to occur, but many can run on a single service. For example, one service running on base path /v2/pets could have 2 RR pairs: one for creating a pet (e.g. a POST with some request data), and one for retrieving the pet (e.g. a GET with the pet ID as relative path).

**Group (SUT)**

Formerly known as a "system under test", a group is a convenient way to organize services. Think of it like a tag; it's just a way to say "these services belong together". It has only 2 fields: a generated ID and a name. The name is prepended to the base path of your virtual service. For example, a service with basepath /v2/pets in the group "test" will run in Mockiato on the base path /virtual/test/v2/pets.

## Getting Started

### Groups

General Groups – Group used for virtual service calls that are executed using the Mockiato URL as the endpoint.

Domain Groups – Group used by EEPS/EAIP domains for virtual service calls that are executed using a proxy or frontend aggregate service that redirects via API Engine Console config setting to Mockiato virtual backend services.

### Owner

The service owner is the person who created the service. If Mockiato is running with the LDAP authentication strategy, then the non-ID fields for the owner model are simply a username and email address. These are pulled from AD automatically on your first login.

### Important Links

- Mockiato Prod: <https://mockiato-prod.origin-ctc-core.optum.com/>

- Mockiato Performance: <https://mockiato-perf.origin-elr-core.optum.com/>

- Splunk: <https://phi-splunk.optum.com/en-US/account/login?>

-

### Access to Mockiato

Raise a request in Secure to be added to the mockiato\_users group.

## Admin Panel

This page allows you to manage groups and group users. Functions include create/delete group and add/remove users from groups.

1. Group names cannot include spaces.
2. If you create a group you are automatically the owner of that group.
3. Any member of a group may add/remove other members from that group.
4. Groups may only be deleted if they have no services associated with them.
5. If you want to be added to a group you can go to Browse->Services and filter by group. The owners MS ID will be displayed next to any service that has been created under that group. Send that individual a request to be added to the group.

### Creating a group in Mockiato

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Select Admin button from navigation bar.
4. Type the name of the group you wish to create in the field under "Create a new group".
5. Click the Create Group Button.
6. You should now be able to see your group in the drop down box labeled “Select a group to view/edit users”

### Add User to Group

1. User **MUST** sign in to Mockiato before they can be added to a group or their MS ID will not show up in the User dropdown menu.
2. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
3. Enter MS credentials to login to the application.
4. Select Admin button from navigation bar.
5. Select the Group name you wish to add a user to from the Group dropdown.
6. Select the MS ID of the user you wish to add and click the + Add Member button.
7. User will now be part of the group.

## Create A New Service

### Create a SOAP service using a request/response pair

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Click on Req/Res Pair button on the home page.
4. Select the radio button for 'SOAP' Service Type and select the group you just created from the Group dropdown box.
5. Enter valid Service name(eg. MockiatoSOAPService),Base Path(eg. /consumer/path/MockiatoSOAPService), and Response Delay(Optional).
6. In the Request/Response Pair panel you can add a label for your pair (Optional).
7. Add request headers (eg. Content-Type application/xml).
8. Enter your request in the Request Payload field.
9. Enter your response Status (eg. 200 OK).
10. Add response headers (eg. Content-Type application/xml).
11. Enter your response in the Response Payload field.
12. Click on 'Publish' button.
13. A pop up should appear showing your virtual service URL.

**Testing direct SOAP service call**

1. Open any SOAP client (eg. SoapUI, Postman, etc.) to test the service.
2. Use the Base URL which was displayed in the success pop up message when you created your service. Alternately you can click the View Endpoints button in the upper right corner when viewing the service or you can select the Service info button on the services screen.
3. Use request body from your virutal service in the request field of your SOAP client.
4. Submit request.
5. Verify response payload returns expected result.

### Create a REST service using a request/response pair

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Click on Req/Res Pair button on the home page.
4. Select the radio button for ‘REST’ Service Type and select the group you just created from the Group dropdown box.
5. Enter valid Service name(eg. Mockiato/rest/service/read),Base Path(eg. /Testing/Mockiato/rest/service/read), and Response Delay(Optional).
6. In the Request/Response Pair panel you can add a label for your pair (Optional).
7. Select your HTTP Method.
   1. Add relative path if needed.
   2. Add query key value pairs if needed.
8. Select Payload Type from dropdown.
9. Add request headers (eg. Content-Type application/json).
10. Enter your request in the Request Payload field.
11. Enter your response Status (eg. 200 OK).
12. Add response headers (eg. Content-Type application/xml).
13. Enter your response in the Response Payload field.
14. Click on 'Publish' button.
15. A pop up should appear showing your virtual service URL.

**Testing direct REST service call**

1. Open any REST client (eg. SoapUI, Postman, etc.) to test the service.
2. Use the Base URL which was displayed in the success pop up message when you created your service. Alternately you can click the View Endpoints button in the upper right corner when viewing the service or you can select the Service info button on the services screen.
3. Select payload type.
4. Add Payload, relative path, or query parameters depending on payload type.
5. Use request body from your virtual service in the request field of your REST client.
6. Submit request.
7. Verify response payload returns expected result.

### Create an MQ service using a request/response pair

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Click on Req/Res Pair button on the home page.
4. Select the radio button for 'MQ' Service Type.
5. Select your group from the Group dropdown box.
6. Enter a valid service name (eg. F5633POP\_FF2\_V1).
7. In the Request/Response Pair panel you can add a label for your pair (Optional).
8. Enter your request in the Request Payload field.
9. Enter your response in the Response Payload field.
10. Click on 'Publish' button.

**Testing direct MQ service call**

1. An application (ex. NeoLoad) will be needed to send a request to MQ. This application must be set up using the same queue information as the group used in Mockiato to create the virtual service.
2. Use request body from your virutal service in the request field.
3. Execute your request.
4. Verify response payload returns expected result.

### Recording

**Set up a recorder for a SOAP/REST service**

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. On the home page, Click on the Record Button.
4. Select service type (SOAP or REST).
5. Select the Group you wish to record your service to from the drop down menu.
6. Enter the name you would like to record your service under (Service name and version are recommended for this value).
7. Provide the Live APIs Host name, Port number, and Base Path.
   1. If Live api url is <https://api6.optum.com:8443/api/beta/benefits/member/v4.5/profile> then host is “api6.optum.com” ,Port is 8443 , Base Path is “/api/beta/benefits/member/v4.5/profile”.
8. If host url contains https check the Use SSL button.
9. Add any headers you would like to keep. This is optional.
10. Select Filter response checkbox if you wish to filter based on status codes, strings, or header values.
    1. Add any status codes, strings, or header values that you would like to omit from your recording.
11. Click the Publish button to create the recorder. A popup window will appear with Recording Live url (Ex : http://mockiato-dev.ocp-elr-core-nonprod.optum.com/recording/live/PerfTestgroup/TestService ) the Live service. This Recording Live URL is where you direct your request to record the live service.

**Set up request and record SOAP/REST service.**

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Open any SOAP/REST client (eg. SoapUI, Postman, etc.) to set up your request.
4. The Recording LiveURL from the recorder you set up is used as your service endpoint.
5. Add required headers, queries, base paths, and/or request payloads, etc required to run your request.
6. Run your request(s).
7. Log in to Mockiato and locate your recorder by navigating to Browse->Recorders on the menu bar and selecting your service on the Recorders page.
8. You should be able to see all of your requests and responses in your recorder.
9. Click the Finalize and Publish button at the bottom of the recorder page to complete the creation of the virtual service in Mockiato.
10. After successful creation of the service in Mockiato you will get a popup with the Mockiato virtual service URL (Ex: http://mockiato-dev.ocp-elr-core-nonprod.optum.com/PerfTestgroup/TestService ), Use this URL to test the virtual service.

### Live Invocation

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Select Browse->Services to locate the service on which you wish to enable live invocation.
4. Check the “**Perform Live Invocation”** check box to display the options.
5. Select “**Invoke Pre or Post Virtual”** option as per your need
   1. **Use Live Service First –** This option will tell Mockiato to call the live service first to get the response. If there is an error in calling the live service then Mockiato will automatically check the request and provide the corresponding virtual response to user.
   2. **Use Virtual Service First -** This option will tell Mockiato to check the Mockiato service Repository first and will return the virtual response. If the Mockiato service does not hav a match in its repository then Mockiato will automatically call the live service and provide the live response.
6. Set up live service details.
   1. Provide live service host name, port number, and base path.
      1. ex.<https://api6.optum.com:8443/api/beta/benefits/member/v4.5/profile> than host is “api6.optum.com” ,Port is 8443 , Base Path is “/api/beta/benefits/member/v4.5/profile”.
   2. Filter responses **–** Is used to filter out any unwanted responses based on status code, header value, or body string value.
   3. Select Use HTTPS checkbox if your service has HTTPS in its url.
   4. Select the Record Live Services checkbox if you would like to record the live request and response and add it to Mockiato.
   5. Click on Update button to add any changes.
7. Test your service and check response header value for “\_mockiato-is-live-backend”. If the value is true your response was returned from the live service. If the value is false your response was returned by Mockiato.

### Bulk upload

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Select the Bulk Upload button on the home page.
4. Select the radio button that matches the service type you are going to upload.
5. Select the group you wish to upload your file to from the Group dropdown menu.
6. Enter the name you would like to use for this service in the Name field. Actual name of the service being virtualized and version number are recommended.
7. If you are creating a SOAP or REST service, enter the base path in the base path field.
8. Click the Upload File button and select the appropriate file from your computer.
   1. File must be a zip file containing a request and a response file for each test case.
   2. Request and response files can end in .txt, .json,, or .xml file extensions but they must match.
   3. Request files MUST end using –req preceeding the file extension or Mockiato will ignore them.
   4. Response files MUST end using –rsp preceeding the file extension or Mockiato will ignore them.
   5. Request and response pairs MUST have the same name prior to the –req or –rsp designation to be associated with one another.
   6. If you are adding several test cases for a service you can use the same name and increment them numerically.
      1. Ex. soapServiceV3-01-req.xml

soapServiceV3-01-rsp.xml

soapServiceV3-02-req.xml

soapServiceV3-02-rsp.xml

* 1. Select the Publish button.
  2. You can locate your service by going to Browse->Services in the tool bar.

### OPEN API

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Select the OpenAPI button on the home page.
4. Select the group you would like to use from the Group drop down menu.
5. Enter the name of the service you are going to virtualize in the Name field. Actual service name and version number are recommended for this value.
6. Upload a file using the Upload File button or enter the URL for the spec in the Spec Info field. We currently support Open API v3.0
7. Select the Overwrite Base Path checkbox if you need a value that differs from what is in the spec and enter the new base path in the Base Path field.
8. Click the Publish button.
9. You can locate your service by going to Browse->Services in the tool bar.

### WSDL

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Select the WSDL button on the home page.
4. Select the group you would like to use from the Group drop down menu.
5. Enter the name of the service you are going to virtualize in the Name field. Actual service name and version number are recommended for this value.
6. Upload a wsdl file using the Upload File button or enter the URL for the wsdl in the Spec Info field.
7. Select the Overwrite Base Path checkbox if you need a value that differs from what is in the spec and enter the new base path in the Base Path field.
8. Click the Publish button.
9. You can locate your service by going to Browse->Services in the tool bar.

## Existing Service Options

### Update

1. Find the service you wish to update by selecting Browse->Services from the menu bar. Filter by Group/Owner if you do not own the service.
2. Click on the name of the service in the left hand column.
3. Update any field (Req/Res Pair, Label etc) or add/remove a pair using the Add or Delete buttons.
4. Click on 'Update'.
5. Test updated service and verify response payload returns expected result.

### Export

1. Find the service you wish to export by clicking Browse on the menu bar and selecting Services, Drafts, or Archive depending on the location of the service. Filter by Group/Owner if you do not own the service.
2. Click the Export button to the right side of your service.

### Stop

1. Find the service you wish to stop by clicking Browse on the menu bar and selecting Services or Recorders depending on the location of the service. Filter by Group/Owner if you do not own the service.
2. Click the Stop button to the right side of your service.

### Delete

1. Find the service you wish to delete by clicking Browse on the menu bar and selecting Services, Recorders, Drafts, or Archive depending on the location of the service. Filter by Group/Owner if you do not own the service.
2. Click the trash can icon to the right side of your service.

### Restore

1. Find the service you wish to restore by clicking Browse on the menu bar and selecting Archive depending on the location of the service. Filter by Group/Owner if you do not own the service.
2. Click the Restore button to the right side of your service.

## Match Templates

Matching templates allow the user to select which fields Mockiato should attempt to match on, and/or provide specific conditions for whether a field is considered to have matched. For detailed information on Match Templates click on the help button [?] located next to the Match Template field in your virtual service.

1. Using Firefox or Chrome browser navigate to <https://mockiato-prod.origin-ctc-core.optum.com/>
2. Enter MS credentials to login to the application.
3. Navigate to your service by selecting Browse->Services on the tool bar and filtering for the appropriate owner/group on the Services page.
4. Select your service from the left hand column.
5. You can create your basic match template either by:
   1. making a copy of your request and removing name/value pairs from .json or elements from xml leaving only the information you wish to match on as well as removing any data inside xml elements
   2. clicking the Create Match Template button located below your Request Payload field which will create an empty representation of your request. You can then edit the resulting generated template to suit your needs.

Best Practices for creating Match Templates

1. Remove ALL data from the template.
2. Only include the name value fields in .json or elements in .xml that you want to match on in your template. Remember, you must maintain the structure of the request for it to be valid.
3. Do not include security header information in your xml Match Templates.
4. Do not include comments in your Match Templates.

## Troubleshooting

Common issues

1. My virtual service doesn’t work.
   1. Verify that your service has been virtualized.
   2. Verify that you are sending your request to the correct url.
   3. Check for typos in both the url and in the request payload you are sending.
   4. Verify you are using the correct http method.
   5. Verify you are sending your request using the correct content type.
2. My test case isn’t working/doesn’t match/returns the wrong result.
   1. Verify that your test case has been virtualized.
   2. Double check request data against the virtualized test case for accuracy/typos.
   3. Ensure that your request aligns with current Match Templates
   4. Ensure there are no Match Templates that are too inclusive.