



# Enov8 DevOps Cheat Sheet

## REST API & Webhooks

Ashley Hosking / Rohit Gupta

## Contents

Introduction .....	4
Scope.....	4
REST-API Overview .....	4
REST-API Prerequisites .....	4
Postman Code Snippets Generator .....	4
Enov8 Postman Collection.....	5
Update Latest Build Version of System .....	6
Context.....	6
URL Structure .....	6
Body (Structure & Example) .....	6
Response (Structure & Example).....	6
cURL Example .....	7
PowerShell Example .....	7
Update Deployed Version of System Instance .....	8
Context.....	8
URL Structure .....	8
Body (Structure & Example) .....	8
Response (Structure & Example).....	8
cURL Example .....	9
PowerShell Example .....	9
Update Deployed Version of System Component .....	10
Context.....	10
URL Structure .....	10
Body (Structure & Example) .....	10
Response (Structure & Example).....	10
cURL Example .....	11
PowerShell Example .....	11
Update Health of a System Component.....	12
Context.....	12
URL Structure .....	12
Body (Structure & Example) .....	12
Response (Structure & Example).....	12
cURL Example .....	13
PowerShell Example .....	13
Update Health of a System Instance .....	14

## Enov8 DevOps Cheat Sheet

Context.....	14
URL Structure .....	14
Body (Structure & Example) .....	14
Response (Structure & Example).....	14
cURL Example .....	15
PowerShell Example .....	15
Update Tag Information .....	16
Context.....	16
URL Structure .....	16
Body (Structure & Example) .....	16
Response (Structure & Example) – On Creation of Tag [POST Request] .....	16
cURL Example – Create Tag.....	17
cURL Example – Update Tag.....	17
PowerShell Example – Update Tag.....	18
Update Environment Event Status .....	19
Context.....	19
URL Structure .....	19
Body (Structure & Example) .....	19
Response (Structure & Example).....	19
cURL Example.....	20
PowerShell Example .....	20
Webhooks .....	21
Context.....	21
Creating a Webhook.....	21
Sample Payload .....	21
Annex .....	23
Generating an API Key.....	23
Class Definition.....	24

# Enov8 DevOps Cheat Sheet

## Introduction

As part of transforming IT's capability, Enov8 is promoting the capture of key CICD metrics within the enov8 platform for the purpose of promoting better Environment & Release Management efficiencies and transparency across the organisation.

## Scope

This document is specifically focussed on supporting the software engineers within Enov8 in sending key CICD metrics to the enov8 platform.

## REST-API Overview

Enov8's REST-API allows you to automatically Create, Update & Read information. Which in turn allows you to integrate other platforms and automation scripts, for example CICD scripts to capture "version", or test automation scripts to capture "health".

Currently the following three types of requests are supported.

1. POST (Create)
2. PUT (Update)
3. GET (Read)

## REST-API Prerequisites

The following will be required before using the enov8 REST-API

- The URL of your enov8 instance  
e.g. <https://company.enov8.com/ecosystem/api/>
- An "API KEY" comprising of the following parameters
  - app\_id (Application ID)
  - app\_key (Application Key)
- An enov8 platform user account with the required permissions
  - user\_id

Note: That a dedicated "Service User Account" is typically created. Please consult [support.enov8.com](https://support.enov8.com) for more information.

Please refer to the Annex of this document for more information on how to create an API KEY or refer to [support.enov8.com](https://support.enov8.com) for more information.

## Postman Code Snippets Generator

For code snippets in alternative languages please refer to an HTTP client with a code generator such as Postman.

[https://learning.getpostman.com/docs/postman/sending\\_api\\_requests/generate\\_code\\_snippets/](https://learning.getpostman.com/docs/postman/sending_api_requests/generate_code_snippets/)

## Enov8 DevOps Cheat Sheet

### Enov8 Postman Collection

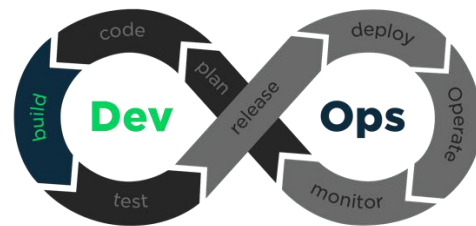
Sample Postman Examples for using REST API's can be imported into POSTMAN using the following link.

<https://www.getpostman.com/collections/d7942dfb7058b7e07052>

### Update Latest Build Version of System

#### Context

The benefit of updating a System within the enov8 platform with the latest Build Version is comparison reporting can be generated between the latest build version available for that particular system and the deployed version across all instances\* of the system (Environment Instances).



Note: Systems (Application Name) within the enov8 platform are conceptual (Non-physical) representations of the applications within an organisation.

\*Updating System Instances is covered in the next section "Update Deployed Version of System Instance".

#### URL Structure

```
{ecosystem api url}/System?user_id={user id}&app_id={app id}&app_key={app key}
```

URL Structure example:

```
https://demo.enov8.com/ecosystem/api/System?user_id=XYZ_USER&app_id=XYZ_APP&app_key=e4zevful7l5j1uaq5iq44
```

#### Body (Structure & Example)

```
{
  "Version": "1.2",
  "Resource Name": "Mulesoft"
}
```

#### Response (Structure & Example)

```
{
  "success": true,
  "total_attempted": 1,
  "total_updated": 1,
  "result": [
    {
      "success": true,
      "System ID": "ECO-000000013632"
    }
  ],
  "value passed": [
    {
      "Version": "1.2",
      "Resource Name": "Mulesoft",
      "System ID": "ECO-000000013632"
    }
  ]
}
```

## Enov8 DevOps Cheat Sheet

### cURL Example

Below is an example of the same call using cURL.

```
curl -X PUT \  
  
'https://demo.enov8.com/ecosystem/api/System?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcctxcaf' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: fcb48bef-6be3-4c79-82af-70b73b05beaf' \  
-H 'cache-control: no-cache' \  
-d '{  
  "Version": "1.2",  
  "Resource Name": "Mulesoft"  
}'
```

### PowerShell Example

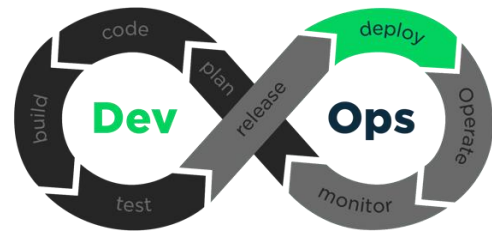
Below is an example of the same call using PowerShell.

```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/System?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcctxcaf'  
  
$jsonmessage = @{  
  'Resource Name'='Mulesoft'  
  'Version'='1.2'  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

## Update Deployed Version of System Instance

### Context

Updating a System Instance within the enov8 platform with the latest Deployed Version from your Deployment tool allows for automated comparison reporting to be generated. This allows greater visibility to the various environment consumers as to what version is currently deployed in the Instance they are using and allows for comparing across all environments.



Note: API Updates via the “environmentinstance” class will decipher the relevant system instance to update based on the connected System and Environment parameters passed as part of the body of the call.

### URL Structure

```
{ecosystem api url}/EnvironmentInstance?user_id={user id}&app_id={app id}&app_key={app key}
```

### URL Structure example:

```
https://demo.enov8.com/ecosystem/api/EnvironmentInstance?user_id=XYZ_USER&app_id=XYZ_APP&app_key=e4zevful7l5j1uaq5iq44
```

### Body (Structure & Example)

```
{  
  "System": "Mulesoft",  
  "Environment": "SIT",  
  "Version": "1.3"  
}
```

### Response (Structure & Example)

```
{  
  "success": true,  
  "total_attempted": 1,  
  "total_updated": 1,  
  "result": [  
    {  
      "success": true,  
      "System ID": "ECO-000000013686"  
    }  
  ],  
  "value passed": [  
    {  
      "Version": "1.3",  
      "System ID": "ECO-000000013686"  
    }  
  ]  
}
```



## Enov8 DevOps Cheat Sheet

### cURL Example

Below is an example of the same call using cURL.

```
curl -X PUT \  
  
'https://demo.enov8.com/ecosystem/api/EnvironmentInstance?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: dea9a16f-a057-4d4a-beb9-6dfdc9c55891' \  
-H 'cache-control: no-cache' \  
-d '{  
  "System": "Mulesoft",  
  "Environment": "SIT",  
  "Version": "1.3"  
}'
```

### PowerShell Example

Below is an example of the same call using PowerShell.

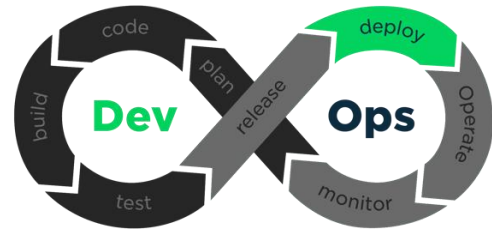
```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/EnvironmentInstance?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf'  
  
$jsonmessage = @{  
  'System'="Mulesoft"  
  'Environment'="SIT"  
  'Version'="1.3"  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

## Update Deployed Version of System Component

### Context

As some application teams may also capture versions on the component/server level, versions can also be passed to the System Component layer (e.g. App Server, Database, etc).

Note: For multiple versions on one System Component the technical specifications property should be used. Please refer to support.enov8.com for further information.



### URL Structure

```
{ecosystem api url}/SystemComponent?user_id={user id}&app_id={app id}&app_key={app key}
```

#### URL Structure example:

```
https://demo.enov8.com/ecosystem/api/SystemComponent?user_id={user id}&app_id={app id}&app_key={app key}
```

### Body (Structure & Example)

```
{  
  "Version": "1.3",  
  "Resource Name": "dev15862.service-now.com"  
}
```

### Response (Structure & Example)

```
{  
  "success": true,  
  "total_attempted": 1,  
  "total_updated": 1,  
  "result": [  
    {  
      "success": true,  
      "System ID": "ECO-000000158325"  
    }  
  ],  
  "value passed": [  
    {  
      "Version": "1.3",  
      "Resource Name": "dev15862.service-now.com",  
      "System ID": "ECO-000000158325"  
    }  
  ]  
}
```

## Enov8 DevOps Cheat Sheet

### cURL Example

Below is an example of the same call using cURL.

```
curl -X PUT \  
  
'https://demo.enov8.com/ecosystem/api/SystemComponent?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: 40fea0b7-a050-436f-9455-2d8d2f15b68d' \  
-H 'cache-control: no-cache' \  
-d '{  
  "Version": "1.3",  
  "Resource Name": "dev15862.service-now.com"  
}'
```

### PowerShell Example

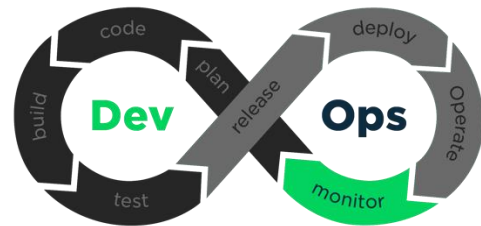
Below is an example of the same call using PowerShell.

```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/SystemComponent?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf'  
  
$jsonmessage = @{  
  'Resource Name'="dev15862.service-now.com"  
  'Version'="1.3"  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

### Update Health of a System Component

#### Context

As part of some CI/CD flows, automated health checks may be utilised to monitor the health of the system post a deployment. The following call can be utilised to update the relevant System Component's health if a health check were to indicate an outage or healthy state.



Note: The following statuses are accepted, "InOperation", "UnplannedOutage", "PlannedOutage".

#### URL Structure

```
{ecosystem api url}/SystemComponent?user_id={user id}&app_id={app id}&app_key={app key}
```

#### URL Structure example:

```
https://demo.enov8.com/ecosystem/api/SystemComponent?user_id={user id}&app_id={app id}&app_key={app key}
```

#### Body (Structure & Example)

```
{
  "Status": "UnplannedOutage",
  "Resource Name": "dev15862.service-now.com"
}
```

#### Response (Structure & Example)

```
{
  "success": true,
  "total_attempted": 1,
  "total_updated": 1,
  "result": [
    {
      "success": true,
      "System ID": "ECO-000000158325"
    }
  ],
  "value passed": [
    {
      "Status": "UnplannedOutage",
      "Resource Name": "dev15862.service-now.com",
      "System ID": "ECO-000000158325"
    }
  ]
}
```

## Enov8 DevOps Cheat Sheet

### cURL Example

Below is an example of the same call using cURL.

```
curl -X PUT \  
  
'https://demo.enov8.com/ecosystem/api/SystemComponent?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: f8d5d0a1-d505-4044-b89f-71bd6d58a25b' \  
-H 'cache-control: no-cache' \  
-d '{  
  "Status": "UnplannedOutage",  
  "Resource Name": "dev15862.service-now.com"  
}'
```

### PowerShell Example

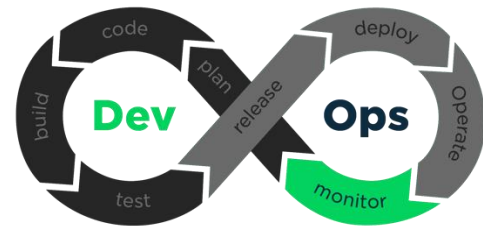
Below is an example of the same call using PowerShell.

```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/SystemComponent?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf'  
  
$jsonmessage = @{  
  'Resource Name'="dev15862.service-now.com"  
  'Status'="UnplannedOutage"  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

## Update Health of a System Instance

### Context

As part of some CI/CD flows, automated test suite can be run to check the application availability post a deployment. The following call can be utilised to update the relevant System Instance health if a health check were to indicate an outage or healthy state.



Note: The following statuses are accepted, "InOperation", "UnplannedOutage", "PlannedOutage".

### URL Structure

```
{ecosystem api url}/SystemInstance?user_id={user id}&app_id={app id}&app_key={app key}
```

### URL Structure example:

```
https://demo.enov8.com/ecosystem/api/SystemInstance?user_id={user id}&app_id={app id}&app_key={app key}
```

### Body (Structure & Example)

```
{  
  "Status": "UnplannedOutage",  
  "Resource Name": "Mulesoft SIT"  
}
```

### Response (Structure & Example)

```
{  
  "success": true,  
  "total_attempted": 1,  
  "total_updated": 1,  
  "result": [  
    {  
      "success": true,  
      "System ID": "ECO-000000158325"  
    }  
  ],  
  "value passed": [  
    {  
      "Status": "UnplannedOutage",  
      "Resource Name": "Mulesoft SIT",  
      "System ID": "ECO-000000158325"  
    }  
  ]  
}
```

## Enov8 DevOps Cheat Sheet

### cURL Example

Below is an example of the same call using cURL.

```
curl -X PUT \  
  
'https://demo.enov8.com/ecosystem/api/SystemInstance?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: f8d5d0a1-d505-4044-b89f-71bd6d58a25b' \  
-H 'cache-control: no-cache' \  
-d '{  
  "Status": "UnplannedOutage",  
  "Resource Name": "Mulesoft SIT"  
}'
```

### PowerShell Example

Below is an example of the same call using PowerShell.

```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/SystemInstance?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf'  
  
$jsonmessage = @{  
  'Resource Name'="dev15862.service-now.com"  
  'Status'="UnplannedOutage"  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

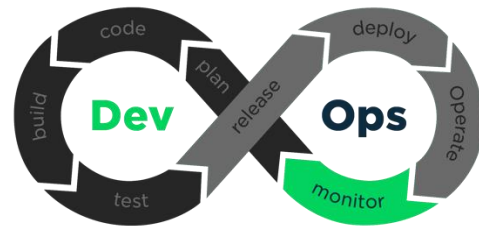
## Enov8 DevOps Cheat Sheet

### Update Tag Information

#### Context

As part of some CI/CD flows, updating information in the tags for a particular resource needs to be regularly updated for reporting and streamlined operations. The following examples can be used to change the value of any tag.

Tags can be created or updated for System, Environment, SystemInstance & SystemComponent classes.



#### URL Structure

```
{ecosystem api url}/Tag?user_id={user id}&app_id={app id}&app_key={app key}
```

#### URL Structure example:

```
https://demo.enov8.com/ecosystem/api/ Tag?user_id={user id}&app_id={app id}&app_key={app key}
```

#### Body (Structure & Example)

```
{
  "Key": "TagKeyName",
  "Value": "TagValue",
  "ResourceName": "GDW",
  "ClassName": "System"
}
```

#### Response (Structure & Example) – On Creation of Tag [POST Request]

```
{
  "success": true,
  "total_attempted": 1,
  "total_created": 1,
  "result": [
    {
      "success": true,
      "System ID": "ECO-000000231847"
    }
  ],
  "value passed": [
    {
      "Key": "TagKeyName",
```



## Enov8 DevOps Cheat Sheet

```
"Value": "TagValue",
"Status": "Active",
"Trend": "False",
"Resource": "ECO-000000013633",
"Source ID": "13633",
"Organisation": "ECO-000000003945"
}
]
}
```

### cURL Example – Create Tag

Below is an example of the same call using cURL.

```
curl -X POST \
'https://demo.enov8.com/ecosystem/api/Tag?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf' \
-H 'Content-Type: application/json' \
-H 'Postman-Token: f8d5d0a1-d505-4044-b89f-71bd6d58a25b' \
-H 'cache-control: no-cache' \
-d '{
    "Key": "TagKeyName",
    "Value": "TagValue",
    "ResourceName": "GDW",
    "ClassName": "System"
}'
```

### cURL Example – Update Tag

Below is an example of the same call using cURL.

```
curl -X PUT \
'https://demo.enov8.com/ecosystem/api/Tag?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcdtxcaf' \
-H 'Content-Type: application/json' \
-H 'Postman-Token: f8d5d0a1-d505-4044-b89f-71bd6d58a25b' \
-H 'cache-control: no-cache' \
-d '{
    "Key": "TagKeyName",
    "Value": "TagValue",
    "ResourceName": "GDW",
    "ClassName": "System"
}'
```

## Enov8 DevOps Cheat Sheet

### PowerShell Example – Update Tag

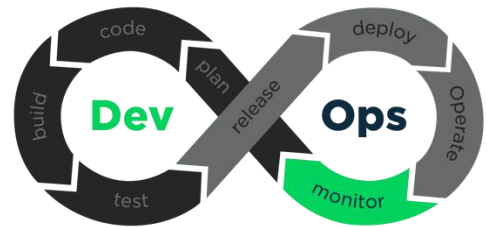
Below is an example of the same call using PowerShell.

```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/Tag?user_id=ENOV8_API&app_id=ENOV8_API&app_key=  
=cc0t5zjnkabbyvcdtxcaf'  
  
$jsonmessage = @{  
    'Key'="TagKeyName",  
    'Value'="TagValue",  
    'ResourceName'="GDW",  
    'ClassName'="System"  
  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

## Update Environment Event Status

### Context

As part of some CI/CD flows, deployment or environment event status would need to be regularly updated for reporting and streamlined operations. The following examples can be used to change the status of any env event



Note: The following statuses are accepted, "Planned", "InProgress", "Passed" ", "Failed" ", "Cancelled" ", "Closed".

### URL Structure

```
{ecosystem api url}/EnvEvent?user_id={user id}&app_id={app id}&app_key={app key}
```

### URL Structure example:

```
https://demo.enov8.com/ecosystem/api/ EnvEvent?user_id={user id}&app_id={app id}&app_key={app key}
```

### Body (Structure & Example)

```
{  
  "Status": "Passed",  
  "System ID": "ECO-000000192052"  
}
```

### Response (Structure & Example)

```
{  
  "success": true,  
  "total_attempted": 1,  
  "total_updated": 1,  
  "result": [  
    {  
      "success": true,  
      "System ID": "ECO-000000192052"  
    }  
  ],  
  "value passed": [  
    {  
      "System ID": "ECO-000000192052",  
      "Status": "Passed"  
    }  
  ]  
}
```

## Enov8 DevOps Cheat Sheet

```
}  
]  
}
```

### cURL Example

Below is an example of the same call using cURL.

```
curl -X PUT \  
  
'https://demo.enov8.com/ecosystem/api/EnvEvent?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcctxcaf' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: f8d5d0a1-d505-4044-b89f-71bd6d58a25b' \  
-H 'cache-control: no-cache' \  
-d '{  
  "Status": "Passed",  
  "System ID": "ECO-000000192052"  
}'
```

### PowerShell Example

Below is an example of the same call using PowerShell.

```
$reqURL =  
'https://demo.enov8.com/ecosystem/api/EnvEvent?user_id=ENOV8_API&app_id=ENOV8_API&app_key=cc0t5zjnkabbyvcctxcaf'  
  
$jsonmessage = @{  
  'System ID'=" ECO-000000192052"  
  'Status'="Passed"  
}  
  
$json = $jsonmessage | ConvertTo-Json  
$resp = Invoke-RestMethod -Uri $reqURL -Method Put -Body $json -ContentType 'application/json'  
Write-Host $resp
```

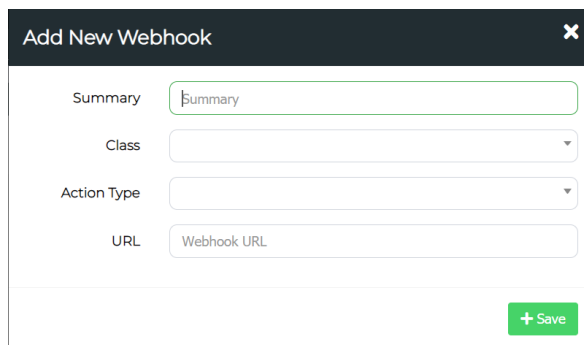
## Webhooks

### Context

Webhooks are automates JSON messages sent out by Enov8 Platform when an object is created or updated. These messages can be consumed by third party applications to take action based on the message.

### Creating a Webhook

As a System Admin user login to the enov8 platform and navigate to **Configuration Management > System Administration > Webhook Management**.

A screenshot of a form titled 'Add New Webhook' with a close button (X) in the top right corner. The form contains four input fields: 'Summary' with the text 'Summary', 'Class' with a dropdown arrow, 'Action Type' with a dropdown arrow, and 'URL' with the text 'Webhook URL'. At the bottom right of the form is a green button with a white plus sign and the text '+ Save'.

**Summary** – Short Title for the Webhook

**Class** – Class for which the webhook will be created

**Action Type** – Event on which webhook will be triggered.  
For Example – Create or Update of object

**URL** – URL to which the message will be sent

### Sample Payload

The following is the sample json payload sent by ecosystem when a system component status is changed.

```
{
  "className": "SystemComponent",
  "webhookType": "ObjectUpdate",
  "System ID": "ECO-000000138499",
  "System User": "demo (DEMO)",
  "System Time": "24-04-2017 17:31:38",
  "Blueprint Name": null,
  "Resource Name": "DB2_09",
  "Status": "InOperation",
  "Description": "",
  "Type": "Database",
  "Sub Type": "DB2",
  "End of life": "/",
  "Cost Center": null,
```

```
"Host Name": null,
"Network Domain": "",
"SystemInstance": [
  "ECO-0000000043688",
  "ICS_MF_LPAR09"
],
"Component Connection": "",
"Data Connection": "",
"RackSpace": "",
"Business Location": "",
"Booking": "",
"Monitored": false,
"Version": "10.2",
"Build No": null,
"Vendor": "",
"Service Provider": "",
"Licence": "",
"Support Group": "",
"Assigned To": [
  [
    "ECO-000000004225",
    "Enterprise IT (DEMO)"
  ],
  [
    "ECO-000000004182",
    "Test Environment Management (DEMO)"
  ]
],
"Tag": "",
"Knowledge Base": "",
"Orchestration": "",
"Technical Specification": "",
"Integration": "",
"Port Monitor": "",
"ClassType": "SystemComponent",
"Organisation": [
  "ECO-000000003945",
  "Organisation (DEMO)"
]
}
```

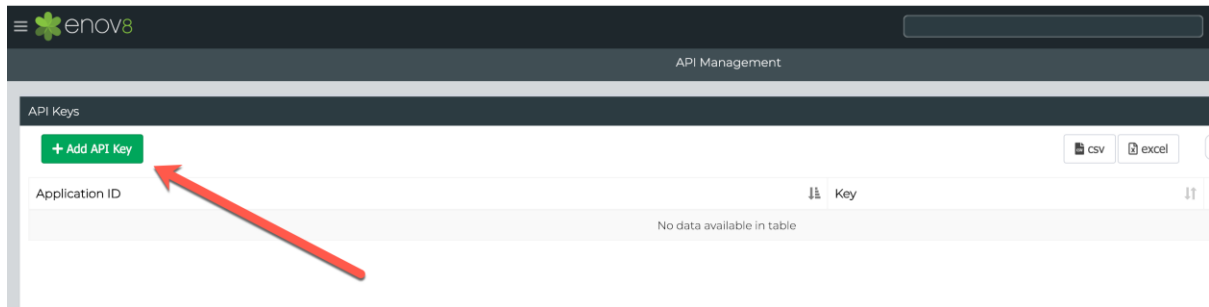
## Annex

### Generating an API Key

The following snapshots have been taken from the EcoSystem support portal.

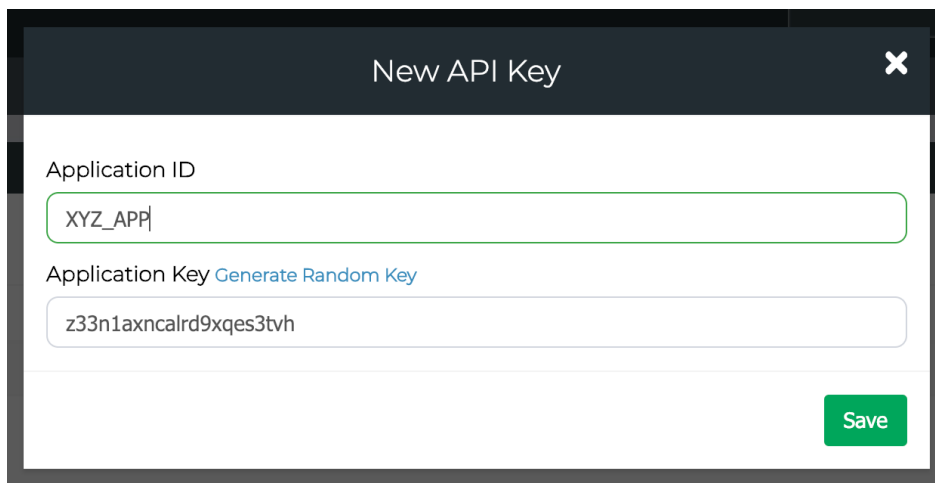
As a System Admin user login to the enov8 platform and navigate to **Configuration Management > System Administration > API Management**.

Click Add API Key



Enter your Application ID and click Generate Random Key for your Application Key.  
Note: Retain these details for future use.

Click Save.

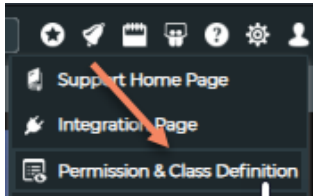
A screenshot of a modal window titled 'New API Key'. The window contains two input fields. The first field is labeled 'Application ID' and contains the text 'XYZ\_APP'. The second field is labeled 'Application Key' and contains the text 'z33n1axncalrd9xqes3tvh'. To the right of the 'Application Key' label is a blue link that says 'Generate Random Key'. At the bottom right of the form is a green 'Save' button.

You should receive a message saying API Key has been created successfully.

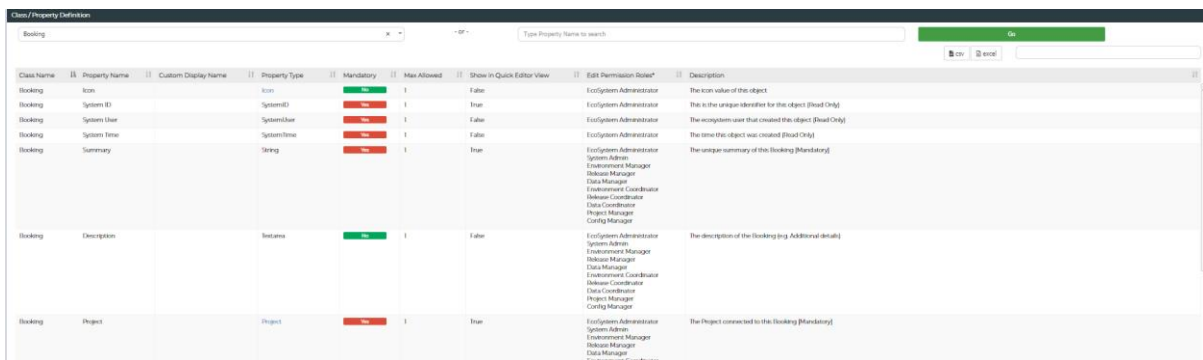
# Enov8 DevOps Cheat Sheet

## Class Definition

Permission & Class Definition module in ecosystem can be used to get details of the properties of a call for API calls and to under user permissions. Login to the enov8 platform and click on the help icon on top bar to open Permission & Class Definition



In Class/Property Definition widget select the class for which you want to see the property definition



Class Name	Property Name	Custom Display Name	Property Type	Mandatory	Max Allowed	Show in Quick Editor View	Edit Permission Group*	Description
Booking	Icon		Icon	Yes	1	False	Ecogystem Administrator	The icon value of the object
Booking	System ID		SystemID	Yes	1	True	Ecogystem Administrator	This is the unique identifier for this object (Read Only)
Booking	System User		SystemUser	Yes	1	False	Ecogystem Administrator	The ecogystem user that created this object (Read Only)
Booking	System Time		SystemTime	Yes	1	False	Ecogystem Administrator	The time this object was created (Read Only)
Booking	Summary		String	Yes	1	True	Ecogystem Administrator System Admin Environment Manager Release Manager Data Manager Environment Coordinator Release Coordinator Data Coordinator Project Manager Config Manager	The unique summary of this Booking (Mandatory)
Booking	Description		System	Yes	1	False	Ecogystem Administrator System Admin Environment Manager Release Manager Data Manager Environment Coordinator Release Coordinator Data Coordinator Project Manager Config Manager	The description of the Booking (eg. Additional details)
Booking	Project		Project	Yes	1	True	Ecogystem Administrator System Admin Environment Manager Release Manager Data Manager Environment Coordinator	The Project connected to this Booking (Mandatory)