Documentación API Reto Python CRM.

Registro de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Nombre | Versión | Acciones |
| 25/07/2024 | Gabriel Lopera M. | 1.0 | Creación |

Contenido

[Introducción 4](#_Toc172877381)

[Login: Web Service Rest tipo POST 5](#_Toc172877382)

[Registrar usuario: Web Service Rest tipo POST 6](#_Toc172877383)

[Listado total de vulnerabilidades – Opción filtro fixeadas: Web Service Rest tipo GET 7](#_Toc172877384)

[Registrar vulnerabilidades Fixeadas: Web Service Rest tipo POST 8](#_Toc172877385)

[Información sumarizada de vulnerabilidades por severidad: Web Service Rest tipo GET 9](#_Toc172877386)

# Introducción

Se han desarrollado varias API con el Framework de Django Rest Framework que tiene como objetivo obtener una lista de vulnerabilidades del NIST y permitir el almacenamiento de las vulnerabilidades que ya han sido corregidas.

A continuación, se describen los métodos de la API para una mejor comprensión de su uso y complejidad.

**Nota:** Las URL o IP:Puerto que se muestran en este documento pueden variar de acuerdo al entorno en el que la Api se encuentre desplegada.

## Login: Web Service Rest tipo POST

Todas las peticiones mencionadas en este documento están protegidas con un método de autenticación vía token JWT el cual deberá ser enviado en la cabecera de las peticiones.

Para obtener el token JWT se deberá tener un usuario establecido y hacer uso del método login del api, a continuación se describe dicho método.

Este método requiere de un usuario y contraseña que serán enviados en el cuerpo de la petición y como respuesta retornara datos del usuario junto con el JWT, **la vigencia del token es de 1 hora** a partir del momento de la petición, pasado este tiempo se deberá volver a hacer uso de este método para obtener un nuevo token.

* Método: POST
* URL: <http://localhost:5173/api/login/>
* Cabecera:
  + Nombre: Content-Type.
  + Valor: application/json.
* Parametros: formato JSON

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Descripción** | **Obligatoriedad** |
| username (string) | Nombre de usuario | Obligatorio |
| password (string) | Contraseña del usuario | Obligatorio |

Tabla 1. Parámetros método POST

* Esquema de la respuesta:

{

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiZXhwIjoxNzIxOTc0ODczLCJlbWFpbCI6ImFkbWluQHBydWViYS5jb20uY28ifQ.RfdVTWnMV7GTH0WUfYHCQ-PQ2eSikZzyYCHZaNb\_D4Y",

"user": {

"username": "admin",

"first\_name": "Admin",

"last\_name": "Admin",

"email": "admin@prueba.com.co",

"is\_superuser": true,

"id": 1

},

"permissions": [

"add\_logentry",

"change\_logentry"

],

"groups": [

{

"id": 1,

"group": "admin"

}

]

}

* Descripción de la respuesta:
  + **user:** objeto con información relacionada al usuario (nombre, apellido, usuario, email, id, si es super usuario, etc)
  + **permissions:** permisos de acceso del usuario.
  + **token:** token (JWT)
  + **groups:** Grupos asignados
* Códigos de respuesta a la petición http

|  |  |
| --- | --- |
| **CÓDIGO** | **DESCRIPCIÓN** |
| 200 OK | Petición recibida con éxito. |
| 400 Bad Request | Petición con formato invalido. |
| 500 internal server error | Error interno en el servicio. |

Tabla 2. Códigos de respuesta Http POST

## Registrar usuario: Web Service Rest tipo POST

* Método: \_POST
* URL: http://localhost:5173/api/admin-user/
* Parametros: formato JSON

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Descripción** | **Obligatoriedad** |
| username (string) | Nombre de usuario | Obligatorio |
| password (string) | Contraseña del usuario | Obligatorio |
| groups (id) | ID del grupo asociar | Obligatorio |
| email | Email | No obligatorio |
| first\_name | Nombre | No obligatorio |
| last\_name | Apellido | No obligatorio |

Tabla 3. Parámetros método POST

* Esquema de respuesta

{

"id": 14,

"username": "glopermaa",

"email": "",

"first\_name": "",

"last\_name": "",

"is\_active": true,

"is\_superuser": false,

"groups": [

1

],

"password": "pbkdf2\_sha256$390000$DSp4UFhEjwOY8nmPoAlWT0$IBuIuDGnasiCIKcV0nMcN2f2tti619n6O8lOub820mM="

}

* Codigos de respuesta a la petición http

|  |  |
| --- | --- |
| **CÓDIGO** | **DESCRIPCIÓN** |
| 200 OK | Petición recibida con éxito. |
| 400 Bad Request | Petición con formato invalido. |
| 500 internal server error | Error interno en el servicio. |

Tabla 4. Códigos de respuesta Http POST

## Listado total de vulnerabilidades – Opción filtro fixeadas: Web Service Rest tipo GET

Permite obtener todas las vulnerabilidades de Common Vulnerabilities and Exposures (CVEs) desde la API del NIST National Vulnerability Database (NVD).

Index será la posición en la lista de vulnerabilidades, page\_size será la cantidad de vulnerabilidades, con un máximo de 2000 por petición. filter\_fixed será la bandera si se desean obtener todas las vulnerabilidades exceptuando las fixeadas.

* Método: GET
* URL: <http://localhost:5173/api/vulnerabilities/all/>
* Query:
  + Nombre: index (obligatorio)
  + Valor: >=0
  + Nombre: page\_size (obligatorio)
  + Valor: >0
  + Nombre: filter\_fixed (Opcional, si se desea filtrar las fixeadas)
  + Valor: boleano
* Cabecera:
  + Nombre: Authorization.
  + Valor: JWT token (token: obtenido mediante el método login)
* Esquema de la respuesta:

{

"status": true,

"data": {

"resultsPerPage": 1,

"startIndex": 0,

"totalResults": 257995,

"format": "NVD\_CVE",

"version": "2.0",

"timestamp": "2024-07-26T05:35:36.247",

"vulnerabilities": [

…..

]

}

}

* Aclaración: recordar que este endpoint sirve para filtrar todas las vulnerabilidades exceptuando las fixeadas, con la bandera Filter en el query.
* Codigos de respuesta a la petición http

|  |  |
| --- | --- |
| **CÓDIGO** | **DESCRIPCIÓN** |
| 200 OK | Petición recibida con éxito. |
| 400 Bad Request | Petición con formato invalido, en el  cuerpo de la respuesta se detallara el error en formato json |
| 500 internal server error | Error interno en el servicio. |

Tabla 5. Códigos de respuesta Http GET

## Registrar vulnerabilidades Fixeadas: Web Service Rest tipo POST

Permite registrar las vulnerabilidades Fixeadas con diferentes valores

* Método: POST
* URL: http://localhost:5173/api/vulnerabilities/
* Cabecera:

|  |  |
| --- | --- |
| **Nombre** | **Valor** |
| Content-Type | application/json |
| Authorization | JWT token (token: obtenido mediante el  método login) |

Tabla 6. Headers petición Http POST

* Parametros: formato JSON

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Descripción** | **Obligatoriedad** |

|  |  |  |
| --- | --- | --- |
| cve\_id (string) | Identificador único de la vulnerabilidad | Obligatorio |
| name (string) | Nombre de la vulnerabilidad | Obligatorio |
| criticality (String) | Criticidad de la vulnerabilidad | Obligatorio |
| status (String) | Estado de la vulnerabilidad | Obligatorio |
| solution\_date (String en formato 2023-02-27T12:00:00Z) | Fecha de solución | Obligatorio |
| description\_vulnerabilitie | Descripción de la vulnerabilidad | No obligatorio |
| description\_solution | Descripción de la solución | No obligatorio |

Tabla 7. Parámetros método POST

* Esquema de la respuesta:

{

"status": true,

"data": {

"cve\_id": "CVE-1999-1474",

"name": "PRUEBA",

"criticality": "HIGH",

"status": "FIXED",

"created\_at": "2024-07-26T09:07:38.965157-05:00",

"last\_update\_at": "2024-07-26T09:07:38.965331-05:00",

"description\_vulnerabilitie": "attack DDoS",

"solution\_date": "2023-02-27T07:00:00-05:00",

"description\_solution": "radware"

}

}

* Códigos de respuesta a la petición http

|  |  |
| --- | --- |
| **CÓDIGO** | **DESCRIPCIÓN** |
| 201 Created | Petición recibida con éxito. |
| 400 Bad Request | Petición con formato invalido, en el  cuerpo de la respuesta se detallará el error en formato json |
| 500 internal server error | Error interno en el servicio. |

Tabla 8. Códigos de respuesta método POST

## Información sumarizada de vulnerabilidades por severidad: Web Service Rest tipo GET

* Método: GET
* URL: http://localhost:5173/api/vulnerabilities/summarized/
* Cabecera:
  + Nombre: Authorization.
  + Valor: JWT token (token: obtenido mediante el método login)
* Esquema de la respuesta:

{

"status": true,

"data": {

"MEDIUM": 64140,

"LOW": 2776,

"HIGH": 64581,

"CRITICAL": 23721,

"TOTAL": 131497

}

}

* Códigos de respuesta a la petición http

|  |  |
| --- | --- |
| **CÓDIGO** | **DESCRIPCIÓN** |
| 200 OK | Petición recibida con éxito. |
| 400 Bad Request | Petición con formato invalido, en el  cuerpo de la respuesta se detallará el error en formato json |
| 500 internal server error | Error interno en el servicio. |

Tabla 9. Códigos de respuesta Http GET