



## **Manual API Mayoristas**

Versión 2.4

**PROVISIONAL**



## Contenido

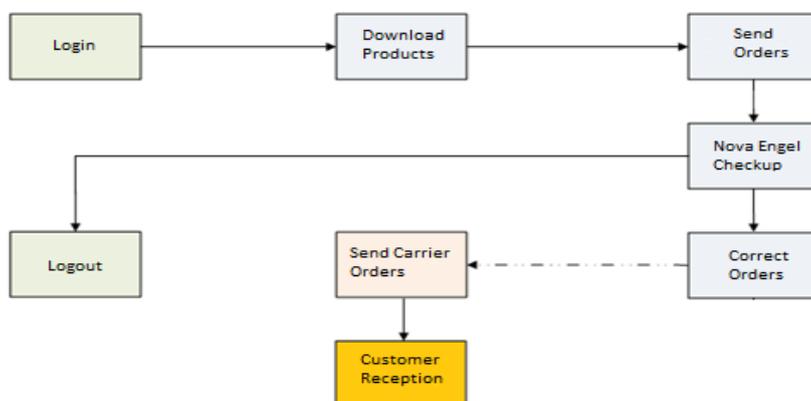
Introducción .....	3
Proceso .....	3
La API (REST).....	4
Login .....	4
Logout.....	5
ListaProductos .....	6
ListaProductosMarca.....	7
ListaProductosPaginado .....	8
ImagenProducto.....	9
Stock .....	9
PedidosEntreFechas .....	10
EnviarPedidos.....	11
ObtenerTracking.....	13
Los Datos .....	14
LoginModel.....	14
ProductModel .....	14
PrecioCantidad .....	15
StockProductModel.....	15
ClientOrderModel .....	16
ClientOrderLineModel.....	16
OrderModel.....	17
OrderLineModel .....	18
OrdersInfoModel.....	18
TrackingModel.....	19
Observaciones .....	20

## Introducción

Nova Engel ofrece esta API, que ofrece una forma de enviar y recibir información a sus clientes: La API utiliza servicio web (Ideal para la cantidad voluminosa de datos), esto se hace a través del protocolo http.

## Proceso

El proceso que se sigue viene resumido en la figura siguiente:



1. **Login:** Identifica a los clientes en el sistema, cada cliente tiene un nombre de usuario y password, que permite el acceso a la página web y los servicios de la API.
2. **Download Products:** Permite actualizar la información sobre los productos ofrecidos por Nova Engel a sus clientes. Nosotros le recomendamos descargar con frecuencia los datos de productos para estar informado sobre las novedades y para conocer el stock en todo momento.
3. **Send Orders:** Esta opción permite enviar a Nova Engel todos los pedidos que queramos para ser entregados al cliente final. Cada pedido deberá estar identificado por el cliente de una sola forma. Por limitaciones del transportista será solo **NUMÉRICO**.
4. **Logout:** Nos desconecta del sistema.



## La API (REST)

Los clientes trabajarán sobre el estándar HTTP por métodos POST y GET. Es una forma fácil de trabajar y permite el formato de datos [JSON](#) y [XML](#).

Los métodos disponibles en el servicio web, se explican en las siguientes páginas, la conexión es: <https://b2b.novaengel.com>. La sintaxis exacta para las llamadas que se van a realizar depende de la plataforma del cliente.

### Login

Esta llamada nos identifica en el sistema. Acepta un nombre de usuario y un password. Si todo ha ido bien, recibiremos una cadena con el token de autenticación que necesitaremos posteriormente para realizar las llamadas a los siguientes métodos. Si por alguna razón falla, no recibiremos datos.

**Método:** *Post*  
**Llamada:** *https://b2b.novaengel.com/api/login*  
**Parámetros (body):** *{user: "prueba", password: "prueba"}*  
**Valor Devuelto:** *LoginModel*

The screenshot shows a REST client interface with the following details:

- Environment:** No environment
- URL:** http://localhost:34203/api/login/
- Method:** POST
- Headers:** Content-Type: application/json, Accept: application/json
- Body:** {user: "prueba", password: "prueba"} (JSON format)
- Status:** 200 OK
- Time:** 55 ms
- Response Body:** {"Token": "67047ece-dcd0-4d28-aaaa-204eefcec804"}



## Logout

Esta llamada invalida el token que estamos usando y nos desconecta del Sistema. No es esencial, pero es aconsejable por motivos de seguridad

<b>Método:</b>	<i>Post</i>
<b>Llamada:</b>	<i><a href="https://b2b.novaengel.com/api/logout/token">https://b2b.novaengel.com/api/logout/token</a></i>
<b>Parámetros (URI):</b>	<i>token</i>
<b>Valor Devuelto:</b>	<i>No devuelve valor</i>

## ListaProductos

Esta llamada nos devuelve los productos disponibles. Para el correcto funcionamiento, es necesario pasarle el token obtenido en el Login. Tenemos la posibilidad de elegir el idioma (es, en, fr...) para traducir información relevante (descripción del producto, contenido lote y categorías). Si por alguna razón falla, recibiremos una lista vacía de ProductModel.

<b>Método:</b>	<i>Get</i>
<b>Llamada:</b>	<i>https://b2b.novaengel.com/api/products/availables/token/lenguaje</i>
<b>Parámetros (URI):</b>	<i>token, lenguaje</i>
<b>Valor Devuelto:</b>	<i>List&lt;ProductModel&gt;</i>

```
[
  {
    "Id": 101028,
    "EANs": [
      "8410412000659"
    ],
    "Description": "SOLAR ACEITE LUMINOSO bronceador SPF0 200 ml",
    "SetContent": null,
    "Price": 4.31,
    "PVR": 8,
    "Stock": 81,
    "BrandId": "1004",
    "BrandName": "Babaria",
    "Gender": "Unisex",
    "Families": [
      "Solar"
    ],
    "Kgs": 0.2,
    "Width": 67,
    "Height": 165,
    "Depth": 46,
    "Vat": 21,
    "Date": "2018-05-21T15:00:38.28",
    "Content": "200 ml",
    "ProductLine": "00000001058",
    "PriceQuantity": [
      {
        "Quantity": 1,
        "Price": 4.31
      },
      {
        "Quantity": 3,
        "Price": 4.07
      },
      {
        "Quantity": 12,
        "Price": 3.95
      }
    ],
    "ItemId": "1004-00659"
  },
],
```

## ListaProductosMarca

Esta llamada nos devuelve una lista de los productos de una marca que hayamos solicitado. Tenemos la posibilidad de elegir el idioma (es, en, fr...) para traducir información relevante (descripción del producto, contenido lote y categorías). Para un correcto funcionamiento, es necesario proporcionar el token obtenido, el Id de la marca deseada para ver sus productos y el idioma. En caso de no especificar el idioma, el programa nos devolverá uno por defecto. Si por alguna razón falla, recibiremos una lista vacía de ProductModel.

**Método:** *Get*  
**Llamada:** *https://b2b.novaengel.com/api/products/brand/token/brandId/lenguaje*  
**Parámetros (URI):** *token, brandId, lenguaje*  
**Valor Devuelto:** *List<ProductModel>*

```
[
  {
    "Id": 87061,
    "EANs": [
      "8410190613430"
    ],
    "Description": "AGUA FRESCA edt vapo 230 ml",
    "SetContent": null,
    "Price": 29.76,
    "PVR": 57,
    "Stock": 71,
    "BrandId": "10",
    "BrandName": "Adolfo Dominguez",
    "Gender": "Hombre",
    "Families": [
      "Perfumes"
    ],
    "Kgs": 0.55,
    "Width": 107,
    "Height": 142,
    "Depth": 99,
    "Vat": 21,
    "Date": "2017-03-09T06:16:39.697",
    "Content": "230 ml",
    "ProductLine": "5200",
    "PriceQuantity": [
      {
        "Quantity": 1,
        "Price": 29.76
      },
      {
        "Quantity": 3,
        "Price": 29.01
      },
      {
        "Quantity": 12,
        "Price": 28.15
      }
    ],
    "ItemId": "10-13430"
  }
]
```

## ListaProductosPaginado

Esta llamada nos devuelve una página de los artículos disponibles. Para que funcione correctamente proporcionaremos el token obtenido, el número de página que queremos descargar (empezando por la 0) y el número de elementos de cada página. En el momento que el número de artículos devueltos sean inferiores a los elementos pedidos significa que hemos descargado todos los datos. Tenemos la posibilidad de elegir el idioma (es, en, fr...) para traducir información relevante (descripción del producto, contenido lote y categorías). En caso de no especificar el idioma, el programa nos devolverá uno por defecto.

**Método:** *Get*  
**Llamada:** *https://b2b.novaengel.com/api/products/paging/token/pages/elements/languaje*  
**Parámetros (URI):** *token, pages, elements, languaje*  
**Valor Devuelto:** *List<ProductModel>*

```
[
  {
    "Id": 101028,
    "EANs": [
      "8410412000659"
    ],
    "Description": "SOLAR ACEITE LUMINOSO bronceador SPF0 200 ml",
    "SetContent": null,
    "Price": 4.31,
    "PVR": 8,
    "Stock": 81,
    "BrandId": "1004",
    "BrandName": "Babaria",
    "Gender": "Unisex",
    "Families": [
      "Solar"
    ],
    "Kgs": 0.2,
    "Width": 67,
    "Height": 165,
    "Depth": 46,
    "Vat": 21,
    "Date": "2018-05-21T15:00:38.28",
    "Content": "200 ml",
    "ProductLine": "0000001058",
    "PriceQuantity": [
      {
        "Quantity": 1,
        "Price": 4.31
      },
      {
        "Quantity": 3,
        "Price": 4.07
      },
      {
        "Quantity": 12,
        "Price": 3.95
      }
    ],
    "ItemId": "1004-00659"
  },
],
```

## ImagenProducto

Esta llamada nos proporciona una URL para poder acceder temporalmente a la imagen del producto solicitado. Para el correcto funcionamiento, es necesario proporcionar el token obtenido por el login y el id del producto deseado. Si por algún motivo falla, recibiremos una cadena vacía.

**Método:** *Get*  
**Llamada:** *https://b2b.novaengel.com/api/products/image/token/productId*  
**Parámetros (URI):** *token, productId*  
**Valor Devuelto:** *string*

## Stock

Esta llamada nos devuelve el stock de todos los artículos disponibles. La información transmitida es la mínima por lo que es un método ideal para mantener actualizado el stock de los productos

**Método:** *Get*  
**Llamada:** *https://b2b.novaengel.com/api/stock/update/token*  
**Parámetros (URI):** *token*  
**Valor Devuelto:** *List<StockProductModel>*

```
[
  {
    "Id": 106118,
    "Stock": 0
  },
  {
    "Id": 106119,
    "Stock": 0
  },
  {
    "Id": 68540,
    "Stock": 5
  },
  {
    "Id": 68545,
    "Stock": 194
  },
  {
    "Id": 68562,
    "Stock": 3
  },
  {
    "Id": 68572,
    "Stock": 477
  }
]
```

## PedidosEntreFechas

Con esta llamada podemos saber los pedidos entre las fechas que elijamos y su estado general. Necesitaremos el token, la fecha de inicio y final. Si por alguna razón falla, recibiremos una lista vacía de OrderModel

**Método:** *Get*  
**Llamada:** *https://b2b.novaengel.com/api/orders/betweendates/token/initiation/final*  
**Parámetros (URI):** *token, initiation (date), final (date).*  
**Valor Retornado:** *List<OrderModel>*

Normal	Basic Auth	Digest Auth	OAuth 1.0	 No environment ▾
--------	------------	-------------	-----------	--

http://localhost:34203/api/orders/betweendates/eeabe531-53d8-421f-90c2-7cb6afd32af9/2014-01-01/2015-02-09

Send Preview Add to collection

Body	Cookies (7)	Headers (10)	STATUS 200 OK	TIME 203 ms
------	-------------	--------------	---------------	-------------

Pretty Raw Preview   JSON XML

```

1  [
2    {
3      "OrderNumber": "M140000426",
4      "Invoice": null,
5      "Date": "2014-12-17T10:25:14.61",
6      "Discount": 0,
7      "Charge1": 10,
8      "Charge2": 0,
9      "Total": 210.1,
10     "Valoration": 0,
11     "ClientId": 5970,
12     "Status": 0,
13     "CarrierNotes": null,
14     "SendInfo": {
15       "ClientDocument": "M140000426",
16       "Tracking": null,
17       "Expedition": "NEEXP513246",
18       "Carrier": "SEUR"
19     },
20     "Lines": [
21       {
22         "ProductId": 29255,
23         "Units": 5,
24         "Price": 16.23,
25         "Discount": 0
26       },
27       {
28         "ProductId": 2977,
29         "Units": 5,
30         "Price": 23.79,
31         "Discount": 0
32       }
33     ],
34     "Name": "Cristian",
35     "SecondName": "Alvarez Muñoz",
36     "Telephone": "968446850",
37     "Mobile": "699854712",
38     "Street": "Pelayo nº 5 5A",
39     "City": "Palma",
40     "County": "Balears",
41     "PostalCode": "96632",
42     "Country": "es"
43   },

```



## EnviarPedidos

Nos permite notificar los pedidos que queremos que sean enviados. Acepta como parámetros el token recibido por el Login, y una lista de pedidos a ser tramitados. Obtendremos información de cada pedido por si hemos encontrado errores. Para que el envío sea correcto, el envío del pedido será en formato JSON:

JSON format, from Body:

```
[
  {
    "OrderNumber": "568746",
    "Valoration": 0,
    "CarrierNotes": "cerrado lunes",
    "Lines": [
      {
        "ProductId": 2977,
        "Units": 5
      },
      {
        "ProductId": 15162,
        "Units": 5
      }
    ],
    "Name": "Cristian",
    "SecondName": "Alvarez Muñoz",
    "Telephone": "968446850",
    "Mobile": "699854712",
    "Street": "Pelayo nº 5 5A",
    "City": "Palma",
    "County": "Balears",
    "PostalCode": "96632",
    "Country": "ES",
  }
]
```



La información recibida será información de los posibles errores que se han encontrado en el pedido, y los pedidos que hayan sido correctos.

**Método:** Post  
**Llamada:** <https://b2b.novaengel.com/api/orders/send/token>  
**Parámetros (URI):** token  
**Parámetros (body):** orderList (json format)  
**Valor Retornado:** List<OrdersInfoModel>

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▼

<http://localhost:34203/api/orders/send/03a96a4c-ca49-405b-b14d-92acf650f5a9/> POST ▼

form-data x-www-form-urlencoded raw JSON ▼

```
1 [
2   {
3     "OrderNumber": "568746",
4     "Valoration": 0,
5     "TransportNotes": "cerrado lunes",
6     "Lines": [
7       {
8         "ProductId": 29255,
9         "Units": 5
10      },
11      {
12        "ProductId": 2977,
13        "Units": 5
14      }
15    ],
16    "Name": "Cristian",
17    "SecondName": "Alvarez Muñoz",
18    "Telephone": "968446850",
19    "Mobile": "699854712",
20    "Street": "Pelayo nº 5 5A",
21    "City": "Palma",
22    "County": "Baleares",
23    "PostalCode": "96632",
24    "Country": "ES",
25  }
26 ]
```

Send Preview Add to collection

Body Cookies (10) Headers (10) STATUS 200 OK TIME 247 ms

Pretty Raw Preview JSON XML

```
1 [
2   {
3     "Message": "Ordens booked 1"
4   }
5 ]
```



## ObtenerTracking

Este método nos permite obtener la información de localización de nuestros pedidos que el transportista nos da. Si por alguna razón falla, recibiremos una lista vacía de TrackingModel.

**Método:** *Get*  
**Llamada:** *https://b2b.novaengel.com/api/trackings/orders/token/orders*  
**Parámetros (URI):** *token, orders*  
**Parámetros (Body):** *orderList (xml format)*  
**Valor Retornado:** *List<TrackingModel>*

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▼

http://localhost:34203/api/trackings/orders/d14e9a61-cc15-4897-aaed-21b4f31678c9/orders?orders=M140000427&orders=M140000427 GET ▼

Send Preview Add to collection

---

Body Cookies (12) Headers (10) STATUS 200 OK TIME 93 ms

Pretty Raw Preview JSON XML

```
1 [
2   {
3     "ClientDocument": "M140000427",
4     "Tracking": "NEEXP478708",
5     "Expedition": "1076462",
6     "Carrier": "SEUR"
7   },
8   {
9     "ClientDocument": "M140000427",
10    "Tracking": "NEEXP478708",
11    "Expedition": "1076462",
12    "Carrier": "SEUR"
13  }
14 ]
```

## Los Datos

A continuación, explicaremos la estructura de los datos usados en la API:

### LoginModel

Asigna un token que necesitaremos posteriormente para realizar las siguientes llamadas.

```
public class LoginModel
{
    public string Token { get; set; }
}
```

### ProductModel

Contiene la información sobre los productos con los que Nova Engel trabaja. El formato es:

```
public class ProductModel
{
    public int Id { get; set; }
    public List<string> EANs { get; set; }
    public string Description { get; set; }
    public string SetContent { get; set; }
    public decimal Price { get; set; }
    public decimal PVR { get; set; }
    public int Stock { get; set; }
    public string BrandId { get; set; }
    public string BrandName { get; set; }
    public string Gender { get; set; }
    public List<string> Families { get; set; }
    public decimal Kgs { get; set; }
    public decimal Vat { get; set; }
    public int Height { get; set; }
    public int Width { get; set; }
    public int Depth { get; set; }
    public List<PrecioCantidad> PriceQuantity { get; set; }
    public string Date { get; set; }
    public string Content { get; set; }
    public string ProductLine { get; set; }
    public string ItemId { get; set; }
}
```

- **Id:** Clave única que identifica el producto.
- **EANs:** Lista de EANs que tiene el producto.
- **Description:** Descripción del producto.
- **SetContent:** Contenido del lote (en el caso que lo sea).
- **Price:** Precio de Nova Engel
- **PVR:** Precio recomendado de este producto.
- **Stock:** Actual stock de Nova Engel
- **BrandId:** Identificador de la Marca del producto.
- **BrandName:** Nombre de la marca del producto.
- **Gender:** Sexo del producto. Puede ser Unisex, Mujer u Hombre.
- **Families:** Lista de las familias a las que pertenece.



- **Vat:** IVA del producto.
- **Kgs:** Peso del producto en Kilogramos.
- **Width:** Ancho del producto en milímetros.
- **Height:** Alto del producto en milímetros.
- **Depth:** Fondo del producto en milímetros.
- **PriceQuantity:** Lista de precios por cantidad.
- **Date:** Fecha de creación de los datos del producto en nuestro sistema.
- **Content:** Contenido del producto.
- **ProductLine:** Gama del producto.
- **ItemId:** Identificador secundario del producto.

### PrecioCantidad

Contiene los precios del producto si la cantidad adquirida en el pedido es igual o superior a la indicada. El formato es:

```
public class PrecioCantidad
{
    public decimal Price { get; set; }
    public string Quantity { get; set; }
}
```

- **Price:** precio del producto.
- **Quantity:** cantidad del producto.

### StockProductModel

Contiene la información actualizada sobre el stock de los productos con los que Nova Engel trabaja. El formato es:

```
public class StockProductModel
{
    public int Id { get; set; }
    public int Stock { get; set; }
}
```

- **Id:** Clave única que identifica el producto.
- **Stock:** Actual Stock de Nova Engel.

## ClientOrderModel

Contiene el pedido para enviar a Nova Engel, la estructura es:

```
public class ClientOrderModel
{
    public string OrderNumber { get; set; }
    public decimal Valoration { get; set; }
    public string CarrierNotes { get; set; }
    public IList<ClientOrderLineModel> Lines { get; set; }
    public string Name { get; set; }
    public string SecondName { get; set; }
    public string Telephone { get; set; }
    public string Mobile { get; set; }
    public string Street { get; set; }
    public string City { get; set; }
    public string County { get; set; }
    public string PostalCode { get; set; }
    public string Country { get; set; }
}
```

- **OrderNumber:** Número de pedido asignado por el cliente. Puede tener una longitud máxima de 15 caracteres. Por limitaciones del transportista, será solo numérico.
- **Valoration:** Parámetro opcional con la valoración.
- **CarrierNotes:** Notas transportista.
- **Lines:** Lista de productos y cantidad que contiene el pedido.
- **Name:** Nombre del cliente final.
- **SecondName:** Apellido del cliente final.
- **Telephone:** Teléfono para que el transportista pueda llamar en caso de necesitarlo (es opcional).
- **Mobile:** Número de móvil para que el transportista pueda llamar en caso de necesitarlo (es opcional).
- **Street:** Calle de envío.
- **City:** Ciudad de envío.
- **County:** Provincia de envío.
- **PostalCode:** Código postal.
- **Country:** País de envío.

## ClientOrderLineModel

Detalle de cada pedido. Indica el id del producto y la cantidad:

```
public class ClientOrderLineModel
{
    public int ProductId { get; set; }
    public int Units { get; set; }
}
```

- **ProductId:** Id del producto.
- **Units:** Número de unidades del producto.

## OrderModel

Información del pedido.

```
public class OrderModel
{
    public string OrderNumber { get; set; }
    public string Invoice { get; set; }
    public DateTime Date { get; set; }
    public decimal Discount { get; set; }
    public decimal Charge1 { get; set; }
    public decimal Charge2 { get; set; }
    public decimal Total { get; set; }
    public decimal Valoration { get; set; }
    public int ClientId { get; set; }
    public enum Status { get; set; }
    public EstadoPedido Status { get; set; }
    public string CarrierNotes { get; set; }
    public TrackingModel SendInfo { get; set; }
    public IList<OrderLineModel> Lines { get; set; }
    public string Name { get; set; }
    public string SecondName { get; set; }
    public string Telephone { get; set; }
    public string Mobile { get; set; }
    public string Street { get; set; }
    public string City { get; set; }
    public string County { get; set; }
    public string PostalCode { get; set; }
    public string Country { get; set; }
}
```

- **OrderNumber:** Número de pedido asignado.
- **Invoice:** Factura del pedido.
- **Date:** Fecha del pedido.
- **Discount:** Descuento del pedido.
- **Charge1:** Cargo 1.
- **Charge2:** Cargo 2.
- **Total:** Total del pedido.
- **Valoration:** Parámetro opcional con la valoración.
- **ClientId:** Id del cliente.
- **Status:** Estado del pedido.
- **CarrierNotes:** Notas transportista.
- **SendInfo:** Información del tracking del pedido.
- **Lines:** Lista de productos y cantidad.
- **Name:** Nombre del cliente final.
- **SecondName:** Apellido del cliente final.
- **Telephone:** Teléfono para que el transportista pueda llamar en caso de necesitarlo (es opcional).
- **Mobile:** Número de móvil para que el transportista pueda llamar en caso de necesitarlo (es opcional).
- **Street:** Calle de envío.



- **City:** Ciudad de envío.
- **County:** Provincia de envío.
- **PostalCode:** Código postal.
- **Country:** País de envío.

**Status** puede tener los siguientes valores: { PendienteEnvio= 1, Procesado = 2, Enviado = 3, ErrorStock = 4, ErrorTransportista = 5, Cancelado = 6, Recibido = 7}

## OrderLineModel

Información de las líneas del pedido. Incluye el descuento y el precio asociado:

```
public class OrderLineModel
{
    public int ProductId { get; set; }
    public int Units { get; set; }
    public decimal Price { get; set; }
    public decimal Discount { get; set; }
}
```

- **ProductId:** Identificador único del producto.
- **Units:** Unidades asignadas.
- **Price:** Precio aplicado.
- **Discount:** Descuento aplicado.

## OrdersInfoModel

Muestra la información de los errores producidos de importación de pedidos. La estructura es:

```
public class OrdersInfoModel
{
    public string Message { get; set; }
    public string OrderNumber { get; set; }
}
```

- **Message:** Mensaje que explica la incidencia.
- **OrderNumber:** Número de Pedido.



## TrackingModel

Indica la información relatada sobre el tracking del pedido.

```
public class TrackingModel
{
    public string ClientDocument { get; set; }
    public string Tracking { get; set; }
    public string Expedition { get; set; }
    public string Carrier { get; set; }
}
```

- **ClientDocument:** Número de pedido asignado.
- **Tracking:** Número de tracking que el transportista asigna al envío.
- **Expedition:** Número de expedición del envío en nuestro sistema
- **Carrier:** Transportista que realiza el envío.



## Observaciones

Todos los métodos mencionados previamente han sido probados con el navegador Google Chrome del cual lo puedes obtener [aquí](#).

También se ha usado la aplicación [POSTMAN](#).

Para realizar pruebas hay que recordar que tenemos dos cabeceras:

Content-Type	application/json
Accept	application/json

The screenshot shows the Postman interface for configuring a request. At the top, there are tabs for authentication: "Normal", "Basic Auth", "Digest Auth", and "OAuth 1.0". A dropdown menu shows "No environment". To the right, there are icons for a keyboard, a close button, and a counter "0". Below this, there is a text input field for the request URL. To its right, a dropdown menu is set to "POST". Further right are buttons for "URL params" and "Headers (2)". Below the URL field, there is a "Manage presets" button. The main area contains a table for headers:

Content-Type	application/json	✕
Accept	application/json	✕
Header	Value	

At the bottom, there are buttons for "form-data", "x-www-form-urlencoded", and "raw". A dropdown menu is set to "JSON".