



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA EN SISTEMAS
INGENIERIA EN SOFTWARE**

Nombre: Jhonny Moreira

Fecha: 13/12/2025

Materia: Aplicaciones Web

Clase 008 – Clase 009

Uso de Bruno para Probar Métodos HTTP en JSONPlaceholder

Objetivo de la práctica

El objetivo de esta práctica fue familiarizarse con la herramienta Bruno para crear, gestionar y probar peticiones HTTP, utilizando distintos métodos HTTP (GET, POST, PUT, PATCH, DELETE) en una API pública de prueba: JSONPlaceholder. Durante la práctica, se creó una colección con múltiples peticiones y se validaron los resultados de cada una.

Pasos seguidos

1. Instalación de Bruno

La primera etapa consistió en instalar la herramienta Bruno desde su página oficial. Esta herramienta permite realizar pruebas de APIs RESTful de manera visual y sencilla, soportando peticiones HTTP con diferentes métodos y mostrando respuestas en tiempo real.

2. Creación de una colección en Bruno

Después de instalar Bruno, se creó una nueva colección denominada "Clase 008 - Clase 009". A través de esta colección, se organizaron todas las peticiones que se iban a realizar a JSONPlaceholder. Cada petición fue creada siguiendo el formato .bru que Bruno utiliza para gestionar las solicitudes.

3. Definición de las peticiones

Cada solicitud HTTP (GET, POST, PUT, PATCH, DELETE) fue definida para interactuar con los siguientes recursos de JSONPlaceholder:

Posts: Obtener, crear, actualizar y eliminar publicaciones.

Comments: Obtener y crear comentarios para las publicaciones.

Albums y Photos: Obtener álbumes y fotos relacionadas.

Todos: Obtener y crear tareas (todos).

Users: Obtener usuarios y sus detalles.

Cada uno de estos recursos fue asociado con su respectivo método HTTP según su función:

GET: Obtener información de la API (ej. GET /posts, GET /users/1).

POST: Crear nuevos recursos (ej. POST /posts).

PUT/PATCH: Actualizar recursos existentes (ej. PUT /posts/1).

DELETE: Eliminar recursos (ej. DELETE /posts/1).



ESCUELA POLITÉCNICA NACIONAL FACULTAD DE INGENIERÍA EN SISTEMAS INGENIERIA EN SOFTWARE

4. Configuración de los parámetros y cuerpos de las solicitudes

Para cada petición, se configuraron los parámetros adecuados, como Path Params y Query Params (ej. ?postId=1 para obtener comentarios de un post específico). En las peticiones POST y PUT, también se configuró el Body con los datos necesarios en formato JSON.

5. Prueba de las peticiones

Con todas las peticiones definidas y configuradas, se procedió a realizar las pruebas. Cada solicitud fue ejecutada y los resultados fueron observados en tiempo real:

Los códigos de respuesta HTTP fueron verificados (por ejemplo, 200 OK para GET, 201 Created para POST).

Las respuestas JSON de la API fueron visualizadas y comparadas con lo esperado (por ejemplo, los detalles de los usuarios al hacer un GET /users/1).

Se utilizó la pestaña Params para los parámetros de consulta y los Headers para configuraciones adicionales como Content-Type.

Resultados y Observaciones

Métodos HTTP: Se probó cada uno de los métodos HTTP de forma efectiva, utilizando JSONPlaceholder como entorno de prueba. Todos los métodos respondieron como se esperaba, proporcionando las respuestas correctas.

Parámetros y Body: La correcta utilización de query params (ej. postId=1) y body JSON para los métodos POST, PUT y PATCH permitió validar que los datos fueron enviados correctamente y que las respuestas contenían la información esperada.

Respuestas HTTP: Se obtuvieron respuestas adecuadas en cada caso. Los códigos de estado HTTP como 200 OK, 201 Created, y 204 No Content fueron correctamente devueltos por la API, lo que indica que las solicitudes fueron procesadas exitosamente.

Conclusiones

La práctica permitió una comprensión profunda sobre el uso de Bruno para gestionar peticiones a APIs RESTful. Aprendimos a configurar solicitudes HTTP con distintos métodos, manejar parámetros y cuerpos de solicitudes, y entender cómo interactuar con una API real para obtener y manipular datos. Además, se validaron los códigos de respuesta HTTP y las respuestas JSON de manera efectiva.

Esta experiencia también fortaleció el uso de herramientas como Bruno en el contexto de pruebas de APIs y es esencial para cualquier desarrollador que busque realizar pruebas locales de manera sencilla y rápida.