APIs

Cami Gomez - Backend developer

Temas

- Qué son las APIs.
- Clasificación.
- Protección.
- Consumo de APIs Web.
- Creando una API.
- Malas prácticas.

API (Application Programming Interface).

Es un conjunto de características y reglas que existen dentro de un programa de software que permiten la interacción con otros sistemas.

Propósitos

1. **APIs de Datos:**

Las APIs de datos proporcionan a varios bancos de datos o proveedores SaaS (Software as a Service o Software como Servicio) acceso CRUD (Create, Read, Update, Delete) a conjuntos de datos subyacentes, permitiendo la comunicación entre una aplicación y un sistema de gestión de bases de datos.

2. **APIs de Funciones:**

Ofrecen funcionalidades específicas, como procesamiento de pagos, conversiones de moneda, o generación de mapas, que se pueden integrar en otras aplicaciones.

3. **APIs de Sistema Operativo:**

Facilitan la interacción entre el software y el sistema operativo subyacente, permitiendo a las aplicaciones utilizar funciones como manejo de archivos, gestión de memoria o comunicaciones de red.

API Web.

En el desarrollo web, una API es generalmente un conjunto de características de código (por ejemplo, métodos, propiedades, eventos y URL) que un desarrollador puede usar en sus aplicaciones para interactuar con sitios web y servicios de terceros.

APIs para pagos

A través de este tipo de API tu negocio puede integrar sistemas dentro del sitio web o la app de tu empresa con la intención de ampliar los métodos de pago para tus productos y servicios.

Ejemplos:

Redes sociales

Las redes sociales también ofrecen innovaciones con APIs para enriquecer la experiencia del usuario e incorporar funcionalidades para obtener información sobre los visitantes o crear usuarios o perfiles en tu site desde cuentas de Facebook, Google, entre otros.

Ejemplos:

Ubicación

Algunas de las APIs más populares y que ya mencionamos son las que permiten ofrecer información y servicios específicos para los usuarios en una localización determinada, potenciando la experiencia.

Ejemplos:

Google Maps Platform

Geoapify

Amazon Location Service

Clasificaciones.

Según el nivel de acceso o disponibilidad.

1. **APIs Públicas (o Abiertas):**

Están disponibles para cualquier desarrollador y tienen como objetivo ampliar el alcance de una organización, ofreciendo servicios o datos a una amplia gama de usuarios.

2. **APIs Privadas (o Internas):**

Son utilizadas internamente por una organización para mejorar sus propios productos y servicios, sin estar disponibles para desarrolladores externos.

3. **APIs de Socio (o Partner):**

Se comparten con socios comerciales específicos y ofrecen un acceso más restringido que las APIs públicas, permitiendo colaboraciones entre empresas.

Según la tecnología o protocolo.

APIs REST (RESTful):

Utilizan HTTP como protocolo de comunicación, siguiendo los principios del estilo arquitectónico REST (Transferencia de Estado Representacional) para ofrecer servicios web ligeros y eficientes.

```
curl --location 'https://api.thecatapi.com/v1/images/upload' \
--header 'x-api-key: DEMO-API-KEY' \
--form 'file=@"/Users/aden/Dropbox/Mac/Downloads/bl4.jpeg"' \
--form 'sub_id="my-user-1"'
```

https://documenter.getpostman.com/view/5578104/RWgqUxxh

Según la tecnología o protocolo.

APIs SOAP (Simple Object Access Protocol):

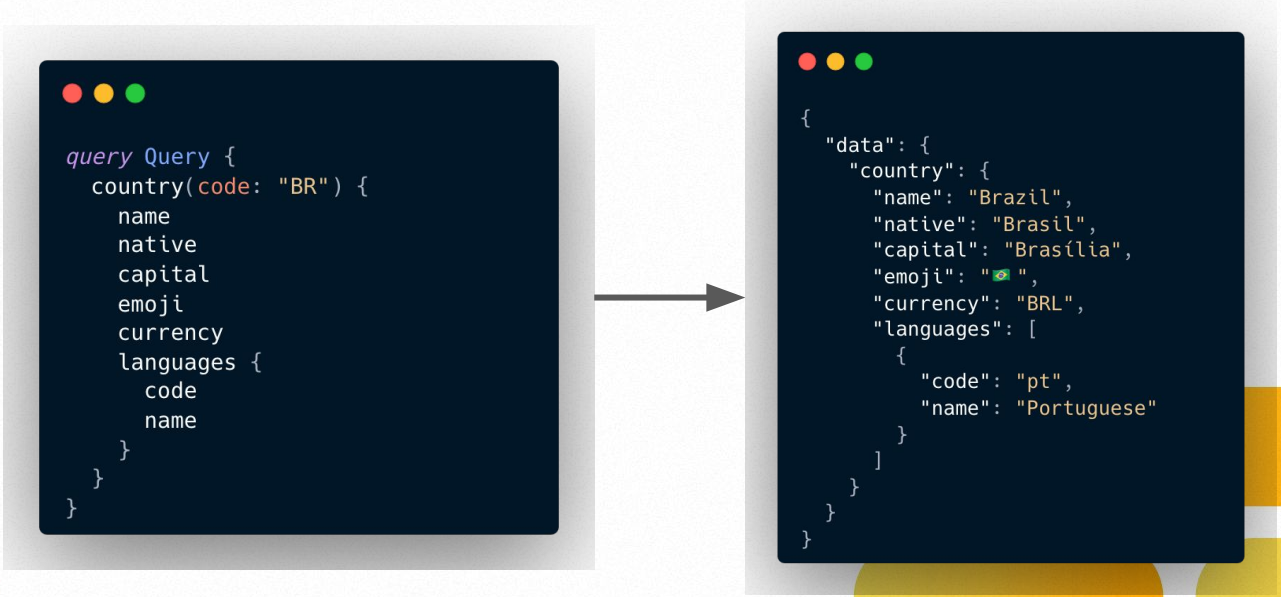
Basadas en el protocolo SOAP, estas APIs ofrecen un método de comunicación más estricto y seguro, utilizando XML para el intercambio de mensajes.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
        xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns1:RequestHeader
         soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
         soapenv:mustUnderstand="0"
         xmlns:ns1="https://www.google.com/apis/ads/publisher/v202402">
      <ns1:networkCode>123456</ns1:networkCode>
      <ns1:applicationName>DfpApi-Java-2.1.0-dfp_test</ns1:applicationName>
    </ns1:RequestHeader>
  </soapenv:Header>
  <soapenv:Body>
    <getAdUnitsByStatement xmlns="https://www.google.com/apis/ads/publisher/v202402">
      <filterStatement>
        <query>WHERE parentId IS NULL LIMIT 500</query>
      </filterStatement>
    </getAdUnitsByStatement>
  </soapenv:Body>
</soapenv:Envelope>
```

Según la tecnología o protocolo.

APIs GraphQL:

Permiten a los clientes solicitar exactamente los datos que necesitan y nada más, lo que puede mejorar la eficiencia de las aplicaciones al reducir la cantidad de datos transferidos.



```
query Query {
  country(code: "BR") {
    name
    native
    capital
    emoji
    currency
    languages {
      code
      name
    }
  }
}
```

```
{
  "data": {
    "country": {
      "name": "Brazil",
      "native": "Brasil",
      "capital": "Brasília",
      "emoji": "🇧🇷",
      "currency": "BRL",
      "languages": [
        {
          "code": "pt",
          "name": "Portuguese"
        }
      ]
    }
  }
}
```

https://studio.apollographql.com/public/countries/variant/current/explorer

Según el formato de intercambio de datos.

1. **APIs basadas en JSON:**

Utilizan JSON (JavaScript Object Notation) como formato para el intercambio de datos, favoreciendo la ligereza y la facilidad de uso, especialmente en aplicaciones web.

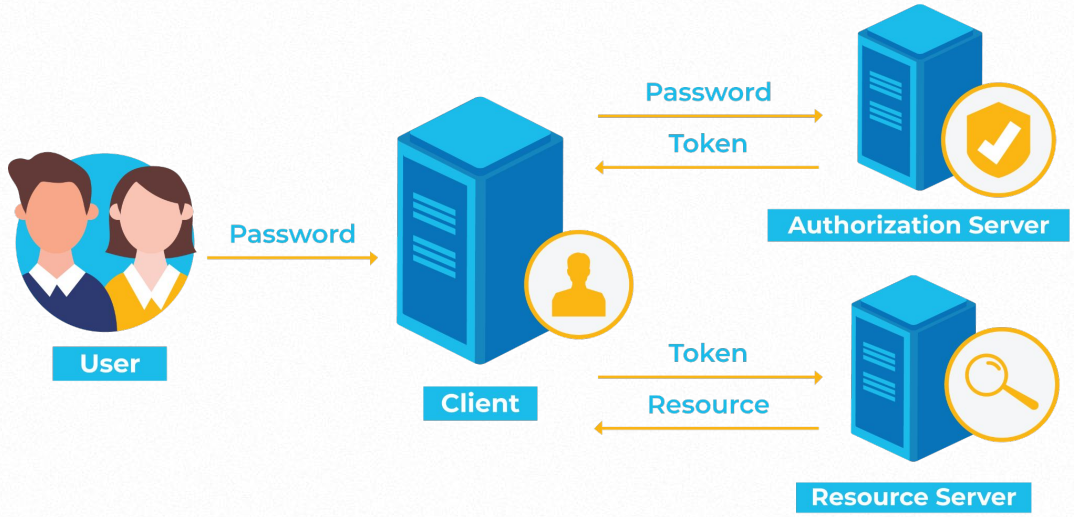
2. **APIs basadas en XML:**

Emplean XML (eXtensible Markup Language) para el intercambio de datos, ofreciendo un formato más estructurado y estricto, común en entornos empresariales y para integraciones complejas.

Protección.

Tokens de autenticación.

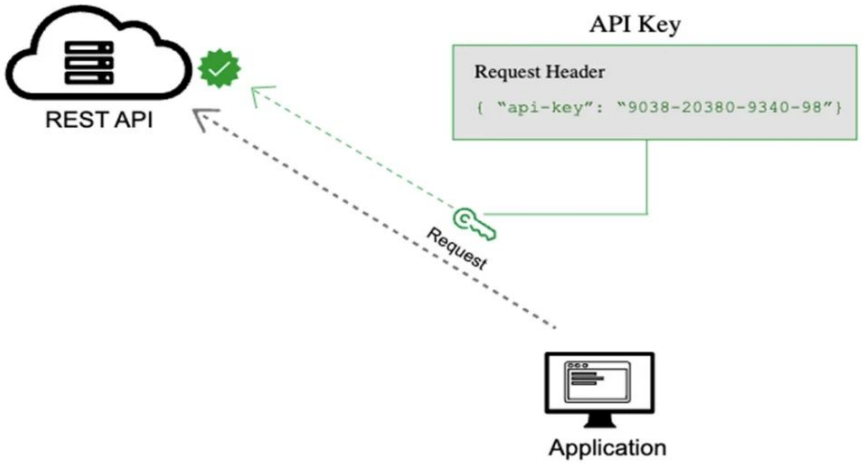
Se utilizan para autorizar a los usuarios a hacer la llamada a la API. Los tokens de autenticación comprueban que los usuarios son quienes dicen ser y que tienen los derechos de acceso para esa llamada concreta a la API.



okta

API key.

Las claves de API verifican el programa o la aplicación que hace la llamada a la API. Identifican la aplicación y se aseguran de que tiene los derechos de acceso necesarios para hacer la llamada a la API en cuestión. Las claves de API no son tan seguras como los tokens, pero permiten supervisar la API para recopilar datos sobre su uso.



Consumir una API

APIs públicas

https://github.com/public-apis/public-apis

https://github.com/graphql-kit/graphql-apis

https://publicapis.dev/category/art-and-design

Cómo diseñar una API

Definir el propósito y los objetivos de la API

Antes de empezar, es crucial entender qué problema va a resolver tu API, quiénes serán los usuarios y cómo se espera que interactúen con ella.

Definir el propósito y los objetivos de la API

API sobre Sakura Card Captor.

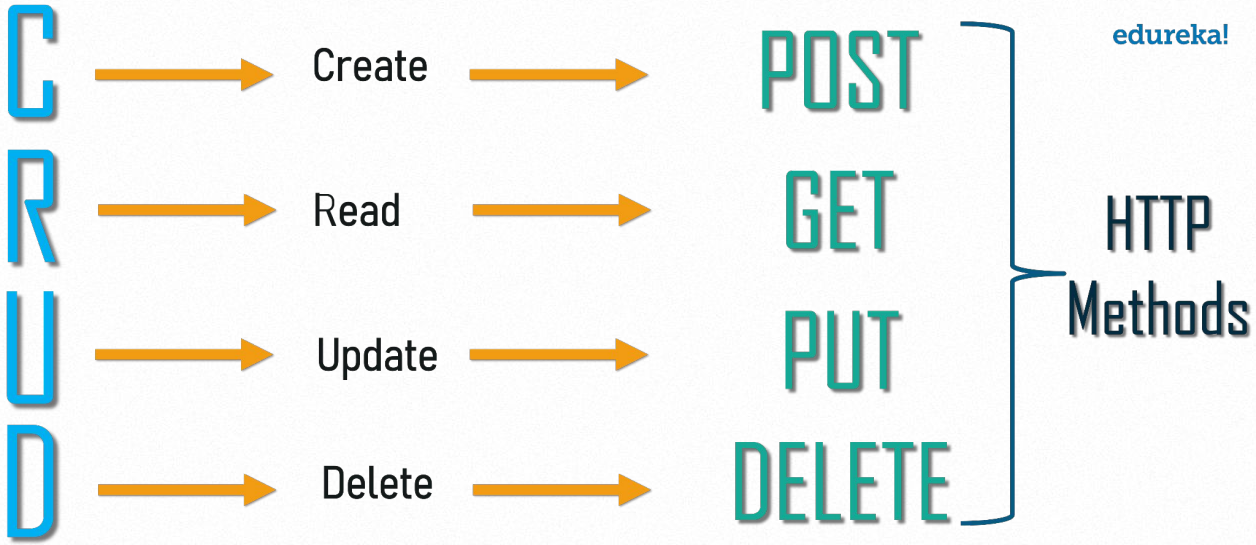
- Obtener información de cada uno de los personajes.
- Obtener información sobre las cartas.
- Crear nuevas cartas y personajes.
- Listar, crear y actualizar los episodios.

Definir las rutas y los métodos HTTP

Asigna las acciones a realizar sobre los recursos a métodos HTTP específicos (CRUD).

- GET
- POST
- PUT/PATCH
- DELETE

Métodos.


C
Create
POST
R
Read
GET
U
Update
PUT
D
Delete
DELETE
HTTP
Methods
edureka!

Definir las rutas y los métodos HTTP

- Usa nombres de recursos en plural.
- Sigue una estructura jerárquica lógica.
- Incluye parámetros a los recursos cuando sea necesario.
- Mantén la simplicidad y la claridad.
- Facilita la filtración, ordenación y paginación a través de parámetros de consulta.

https://www.sakura.com/api/

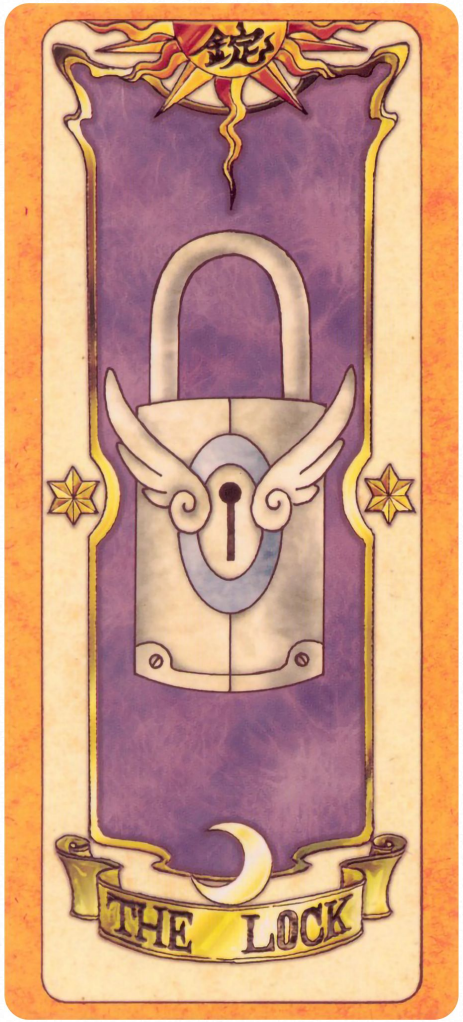
Method	API endpoint	Description
GET	/episodes	Lista de episodios.
GET	/episodes/<episode_id>/comments	Lista de comentarios a un episodio.
GET	/episodes/<episode_id>/comments/<comment_id>	Info de un comentario específico.
POST	/episodes/<episode_id>/comments	Crear un nuevo comentario.
PUT	/episodes/<episode_id>/comments/<comment_id>	Actualizar un comentario
PATCH	/episodes/<episode_id>/comments/<comment_id>	Actualizar parcialmente un comentario.
DELETE	/episodes/<episode_id>/comments/<comment_id>	Eliminar un comentario.

Method	API endpoint	Description
GET	/cards/	Lista de cartas clow.
GET	/cards/<card_id>/	Info de una carta específica.
GET	/cards?captured_by=sakura&limit=20	Lista de cartas capturadas por Sakura.

Considerar la seguridad desde el principio

Planifica cómo autenticar y autorizar a los usuarios de tu API. Considera el uso de tokens, OAuth, o cualquier otro mecanismo adecuado de seguridad.

API	Description	Auth
Sakura API	API para mostrar información de Sakura Card Captor.	apiKey / OAuth



Implementar el versionado.

Decide cómo versionar tu API para gestionar cambios futuros sin interrumpir a los usuarios existentes. Una práctica común es incluir el número de versión en la URL.

Method	API endpoint	Description
GET	/v2/characters/<character_id>/	Lista de personajes.
GET	/v2/characters/<character_id>/images	Lista de imágenes un personajes.
GET	/v2/characters/<character_id>/images/<image_id>	Info de una imagen específica.

v1

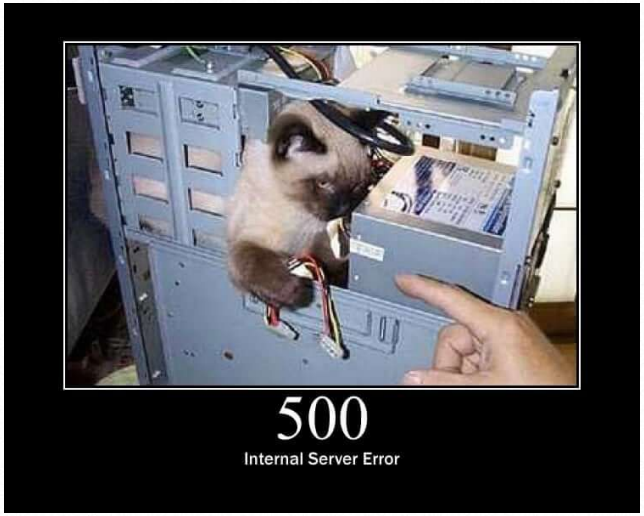
```
{
  name: "Shaoran",
  image: "shaoran.jpg"
}
```

v2

```
{
  name: "Shaoran",
  "thumbnail": {
    "path": "shaoran_thu.jpg",
    "extension": "jpg"
  },
  images: [
    {
      "extension": "jpg",
      "path": "shaoran.jpg"
    },
    {
      "extension": "png",
      "path": "shaoran.png"
    }
  ]
}
```

Definir los códigos de estado HTTP

Usa códigos de estado HTTP para comunicar el resultado de las solicitudes de manera estándar y entendible.



Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases:

- Respuestas informativas (100–199)
- Respuestas satisfactorias (200–299)
- Redirecciones (300–399)
- Errores del cliente (400–499)
- Errores del servidor (500–599)

https://datatracker.ietf.org/doc/html/rfc2616#section-10

POST *http://www.sakura.com/api/episodes/*

Resultados posibles:

- *403* - Acceso prohibido.
- *400* - petición incorrecta, p.ej. falta un campo o su valor no es válido.
- *500* - Error del lado del servidor al intentar crear el recurso, p.ej. se ha caído la BD.
- *201* - Recurso creado correctamente.

DELETE *http://www.sakura.com/api/episodes/1*

Resultados posibles:

- *403* - Acceso prohibido.
- *500* - Error del lado del servidor al intentar eliminar el recurso.
- *204* - El recurso fue eliminado correctamente (No content).

Documentar la API

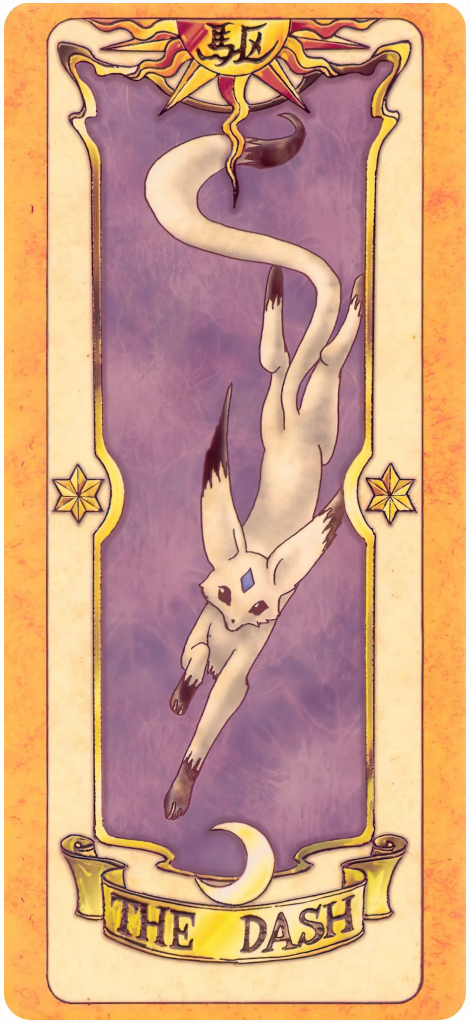
Crea una documentación clara y detallada que incluya información sobre los endpoints, métodos, parámetros de solicitud, formatos de respuesta, códigos de estado y ejemplos de uso.

Swagger

Postman

Pruebas

Diseña y ejecuta pruebas unitarias, pruebas de integración, y pruebas de carga para asegurarte de que tu API funciona como se espera y puede manejar la carga prevista.



Obtener retroalimentación e iterar

Comparte tu API con un grupo de usuarios beta y solicita su retroalimentación. Usa esta información para hacer ajustes y mejoras.

Malas prácticas.

Malas prácticas.

- Evita incluir verbos o acciones en los paths.
- Sobrecarga de endpoints.
- Ignorar los códigos de estado HTTP.
- Inconsistencias en el diseño.
- Falta de versionado.
- Seguridad por oscuridad (Confiar en que los detalles de implementación no sean conocidos).
- Datos expuestos innecesariamente.
- Falta de limitación de tasa.
- Documentación insuficiente o desactualizada.
- No manejar errores de manera adecuada.

Gracias.