

Resumen Header y Request RESTful over HTTP

Headers

cualidad	Header	Msg Type
Compresión	<code>Accept-encoding : gzip</code>	client request
Compresión	<code>Content-Encoding : gzip</code>	server response
<i>Básico</i> : encoding	<code>Content-type : application/json</code>	server response
Cacheabilidad	<code>Expires : Fri, 19 Nov 2021 19:20 EST</code>	server response
Basic Authentication	<code>Authorization: Basic</code> : /clave y contraseña encodeados, no encriptados/	client request
API Keys	<code>X-API-Key</code> : /puede ser aleatoria o JWT/	client request
Bearer Authentication /Token Auth	<code>Authorization: Bearer</code> </token JTW por ejemplo/>	client request
Versionado (tambien se puede usar un custom header o extender la UIR con ella versión en particular)	- <code>Accept:</code> application/vnd.example.v1+json - <code>Accept-version-custom</code> : v1	client request
Paginado	<code>Pagination-Count</code> : 100 (total de elementos) <code>Pagination-Page</code> : 5 (num de pagina) <code>Pagination-Limit</code> : 20 (elementos per page)	server response

Query String: URL

quality	Ejemplo	Msg type	Interpretación
filtrado	GET /users? age= 18 &country= AR &active= true	URL con parámetros dinámicos definidos por el cliente	filtra según los parámetros indicados. El recomendado para GET
Ordenamiento	GET /events? sort= criterio1 , - criterio2descendente , criterio2	URL con parámetros dinámicos definidos por el cliente	usa sort= para definir ordenamiento y agrega ',' para agregar criterios de ordenamiento
versionado	GET [url]/api/ v1 /resource	URL a la que accede un cliente para interactuar con otra versión	se agrega el versionado

Request Body

se suelen seguir siendo estandares como por ejemplo JSON:API que define los campos en una request:

- "data" : recursos que el cliente solicita y el serv responde
- "errors" : cuando hay errores se usa en vez de data
- "metadata" : se puede usar para incluir timestamp o para paginado.
- "links" : se agregan URLs de interés

Quality	Label en el body del server response	emissor
filtrado	"filters": /mapa de los campos por los cuales filtrar y los contenidos específicos/	client (en post, no estandarizado)
ordenamiento	"sort" /array de los campos pos los cuales ordenar ('-' si es descendente)	client (en post, no estandarizado)
paginado	"meta": {"page": , "per_page": , "page_count": , "total_count": }	server response (estandarizado), más común que con headers

Ejemplo didáctico

Descripción: Permite que los usuarios inicien sesión.

Request:

- Headers: Authorization: Basic {base64(email:password)}
- POST /api/v1/users/token

Response:

Código HTTP: 200 OK

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh_token": "m9obiBEb2UiLCJpYXQiOjE1MTYyMzkwMj..."
}
```

Código HTTP: 401 Unauthorized si las credenciales son incorrectas.

descripción: permite que los usuarios inicien sesión

request:

- headers: Authorization: Bearer XHUIAHLSFAJP
- POST /api/v1/users/token

response:

código HTTP: 200 OK

```
{
```

```
"access_token": "hjhadakljdjakljaaklskh546ja"
```

```
"refresh_token": "dhasjkdhajadsdas"  
}  
código HTTP: 400 Unauthorized (credenciales incorrectas)
```

Ejemplo real

```
// Métodos auxiliares (simulados con printlns)
private String readFile(String fileName) {
    return ""; // simulado
}
```

```
// Métodos auxiliares simulados  
private String readFile(String fileName) {  
    } return ""; // simulado
```

FIUBA Departamento de Computación
Nombre Completo:
Rodríguez, Daniel

Fecha: 30/6/2025
Tiempo: 120 minutos

1. Un equipo de desarrollo de tres integrantes se encuentra trabajando en un proyecto ágil, utilizando la técnica de poker planning y utilizando la siguiente secuencia de Fibonacci (1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987) para asignar peso a las tareas. Al comenzar la estimación de tareas se encuentran con el siguiente conflicto: Uno vota que una tarea debe recibir el peso de 144, sin embargo, los otros dos uno vota que debe ser de 233 y el otro 89. En la siguiente les pasa lo mismo, pero con los valores 3, 5 y 8. Eso les pasa en casi todas las tareas, y les cuesta llegar a un consenso por lo que pasan mucho tiempo analizando y discutiendo las estimaciones. Más a o menos están alineados con la dificultad, pero no suelen asignar todos el mismo peso. ¿Qué les recomendaría?

2. Imaginen el contexto de una app mobile, donde los usuarios se pueden bajar una aplicación móvil Android o iOS, y que también pueden acceder vía web a una aplicación online.

S) Dentro de muchas otras funcionalidades se incluyen las siguientes:
La app permite buscar eventos de tipo deportivo relacionados.

Permite buscar eventos de tipo deportivo relacionados a running y ciclismo. Se pueden buscar y filtrar los eventos por nombre, tipo, fecha, lugar. La lista de eventos puede ser muy extensa. También permite crear su cuenta de usuario y loguearse a la app con sus credenciales. Un usuario debe tener por lo menos, nombre, apellido, y opcionales foto avatar, fecha de nacimiento y género. Estando logueado puede marcar y desmarcar como favorito los eventos que le interesan.

También existe un rol administrador, que desde la web, puede dar de alta nuevos eventos, indicando nombre, tipo, fecha, y lugar así como un detalle y una foto para el mismo. También puede editarlos.

Para el problema indicado:

a- Se pide definir a un máximo detalle la interfaz de una API rest que soporte las funcionalidades pedidas para la app y web. Utilizar JWT como bearer token, y un token simple como refresh token. Incluya también los servicios que requeriría un administrador para cargar contenido.

b- Nosotros podemos decidir cuándo desplegar la API y las apps en el store, pero no podemos garantizar que los usuarios se instalen la última versión de la app. Explique qué problemas puede haber a partir de esto y proponga posibles soluciones a los mismos.

Plantee para el caso detallado de eventos deportivos, un caso de un cambio en la API y app que son compatibles hacia atrás y un caso de cambios que no son compatibles. (defina un nuevo endpoint para cada cambio y explique por qué es o no compatible) Y explique cómo solucionaría que una app vieja que usa una versión vieja de la API cuando llegan cambios que no son compatibles.

3) a. Analice el siguiente código y comente si encuentra algún "code smell" o mala práctica en su diseño e implementación.

b. Se desea agregar soporte para archivos CSV (Comma Separated Values) Agregue dicha funcionalidad respetando las buenas prácticas, y usando principios o patrones y técnicas vistas en la materia.

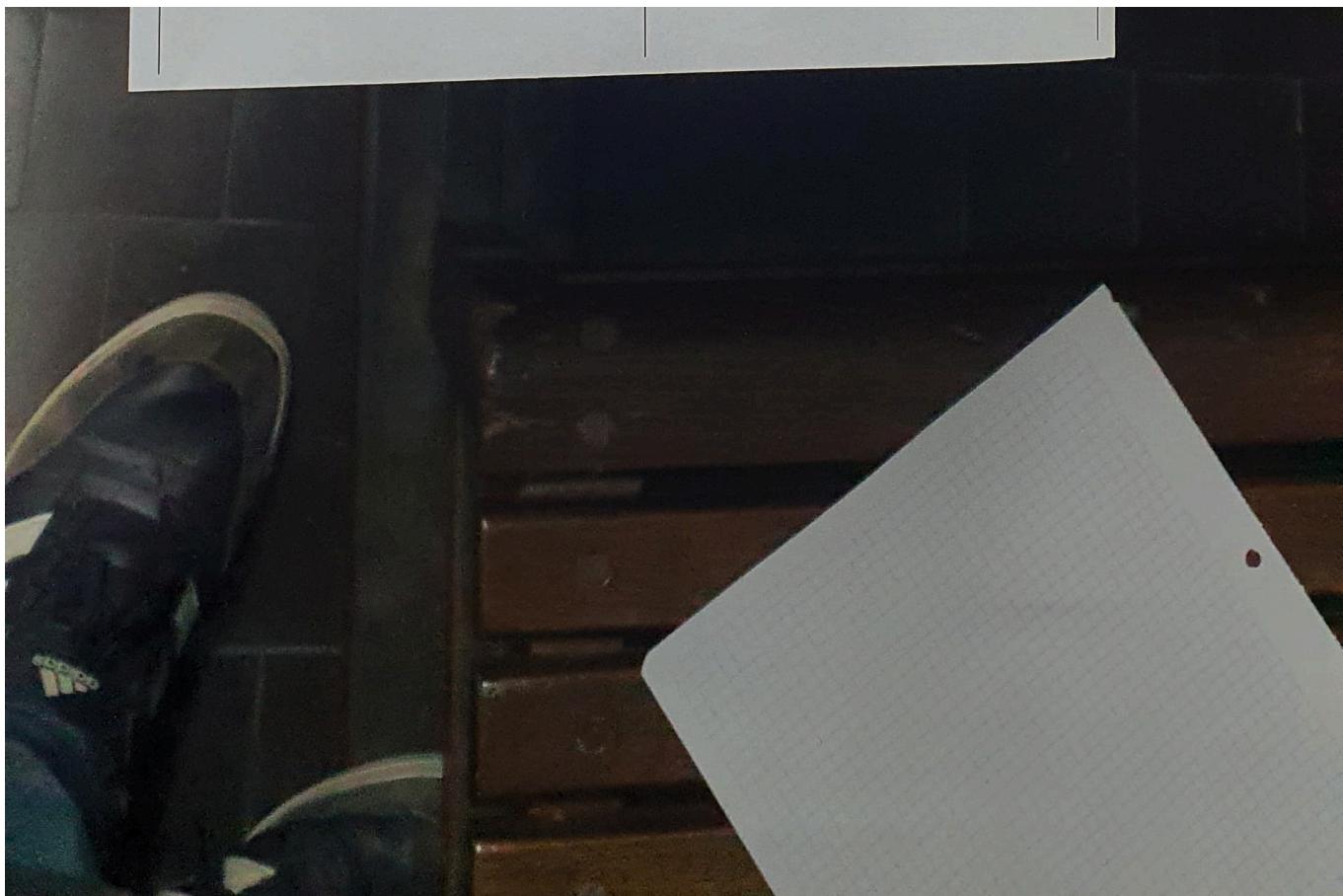
```
class JSONDataProcessor {  
  
    public void process(String fileName) {  
        System.out.println("Loading data...");  
        String content = this.readFile(fileName);  
        JSONObject jsonContent = this.parseJson(content);  
        if (this.validateJson(jsonContent)) {  
            Object extracted =  
this.extractFromJson(jsonContent);  
            this.processContent(extracted);  
        } else {  
            throw new IllegalArgumentException("Contenido  
inválido");  
        }  
        this.saveToJson(jsonContent, fileName);  
        this.saveMetadata(jsonContent);  
        System.out.println("Processing complete.");  
    }  
}
```

```

class XMLDataProcessor {

    public void process(String fileName) {
        System.out.println("Loading data...");
        String content = this.readFile(fileName);
        XMLObject xmlContent = this.parseXML(content);
        if (this.validateXML(xmlContent)) {
            Object extracted = this.extractFromXml(xmlContent);
            this.processContent(extracted);
        } else {
            throw new IllegalArgumentException("Contenido
inválido");
        }
        this.saveToXML(xmlContent, fileName);
        this.saveMetadata(xmlContent);
        System.out.println("Processing complete.");
    }
}

```



los usuarios pueden bajar una app móvil android o iOS que también se puede acceder vía web en una app online.

Las funcionalidades son:

- buscar eventos de tipo deportivo relacionados a running y ciclismo.
- buscar y filtrar eventos por nombre, tipo, fecha, lugar (paginado)
-
-