T.P. Fetch

PARTE TEORICA

Investiga y responde las siguientes preguntas de manera detallada:

1. ¿Qué es el método `fetch` en JavaScript y cuál es su propósito en el desarrollo web?

2. Describe la sintaxis básica del método `fetch` y menciona los parámetros más

comunes que acepta.

3. Explica cómo se manejan las respuestas obtenidas mediante el método `fetch`,

incluyendo el uso de los métodos `json()`, `text()` y `blob()`.

4. ¿Cuál es la ventaja de utilizar `fetch` en comparación con enfoques tradicionales

como XMLHttpRequest?

5. Habla sobre la importancia de los encabezados (headers) en las solicitudes `fetch`.

¿En qué situaciones podrían ser útiles?

Desarrollo

1. El método fetch en JavaScript es una interfaz para hacer solicitudes de red y manipular partes del protocolo HTTP, como solicitudes y respuestas. Proporciona una forma fácil y lógica de obtener recursos de forma asíncrona a través de la red. A diferencia de XMLHttpRequest, que es una API basada en devoluciones de llamada, fetch es una API basada en promesas y proporciona una mejor alternativa que puede ser utilizada fácilmente en trabajadores de servicio.
2. La sintaxis básica del método fetch es fetch(url, [options]), donde url es la URL del recurso que se desea obtener y options es un objeto opcional que contiene cualquier configuración personalizada que se desee aplicar a la solicitud. Algunas de las opciones más comunes incluyen method, que especifica el método HTTP a utilizar (por ejemplo, “GET”, “POST”, etc.), headers, que especifica cualquier encabezado que se desee agregar a la solicitud, y body, que especifica cualquier cuerpo que se desee agregar a la solicitud.
3. Las respuestas obtenidas mediante el método fetch se manejan utilizando promesas. El método fetch devuelve una promesa que se resuelve con un objeto Response que representa la respuesta a la solicitud. El objeto Response no contiene directamente el cuerpo de la respuesta, sino que es una representación de toda la respuesta HTTP. Para extraer el contenido del cuerpo de la respuesta, se pueden utilizar métodos como json(), text() y blob().

json(): Este método se utiliza para extraer el contenido del cuerpo de la respuesta como un objeto JavaScript. Devuelve una promesa que se resuelve con el resultado de analizar el texto del cuerpo de la respuesta como JSON.

text(): Este método se utiliza para extraer el contenido del cuerpo de la respuesta como una cadena de texto. Devuelve una promesa que se resuelve con una cadena de texto que representa el contenido del cuerpo de la respuesta.

blob(): Este método se utiliza para extraer el contenido del cuerpo de la respuesta como un objeto Blob. Devuelve una promesa que se resuelve con un objeto Blob que representa el contenido del cuerpo de la respuesta.

1. Una ventaja de utilizar fetch en comparación con enfoques tradicionales como XMLHttpRequest es que fetch es más fácil de usar y proporciona una sintaxis más limpia y concisa para hacer solicitudes de red. Además, al ser una API basada en promesas, permite un mejor manejo de errores y un código más legible al evitar el “infierno de las devoluciones de llamada”. También integra conceptos avanzados de HTTP como CORS y otras extensiones a HTTP.
2. Los encabezados (headers) son importantes en las solicitudes fetch porque permiten enviar información adicional junto con la solicitud al servidor. Por ejemplo, los encabezados pueden utilizarse para especificar el tipo de contenido que se está enviando o solicitando, para enviar información de autenticación o para controlar el comportamiento del caché. Los encabezados pueden ser útiles en muchas situaciones, como cuando se necesita enviar datos en un formato específico o cuando se necesita autenticarse para acceder a un recurso protegido.