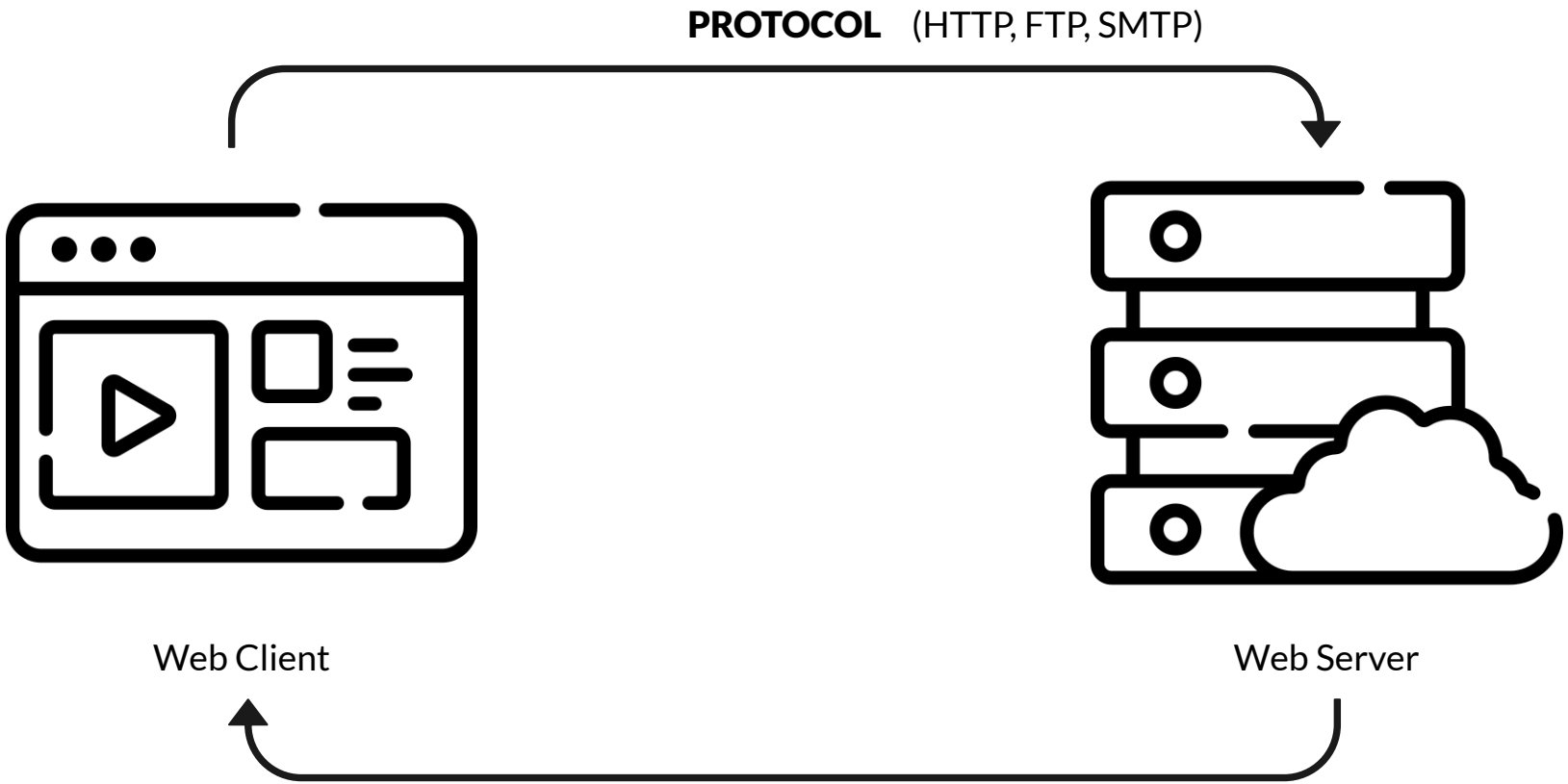
Aprendiendo Fetching y Axios

@KattyaCuevas

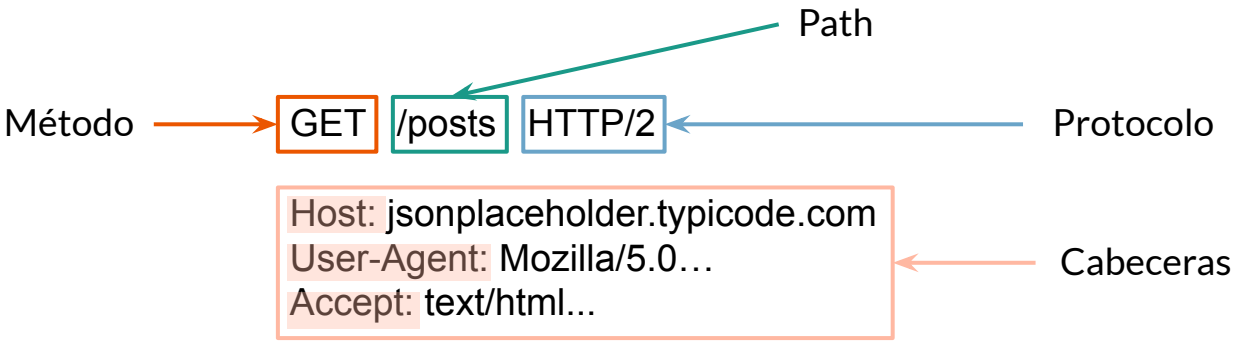
¿Cómo funciona la WEB?


PROTOCOL (HTTP, FTP, SMTP)
Web Client
Web Server

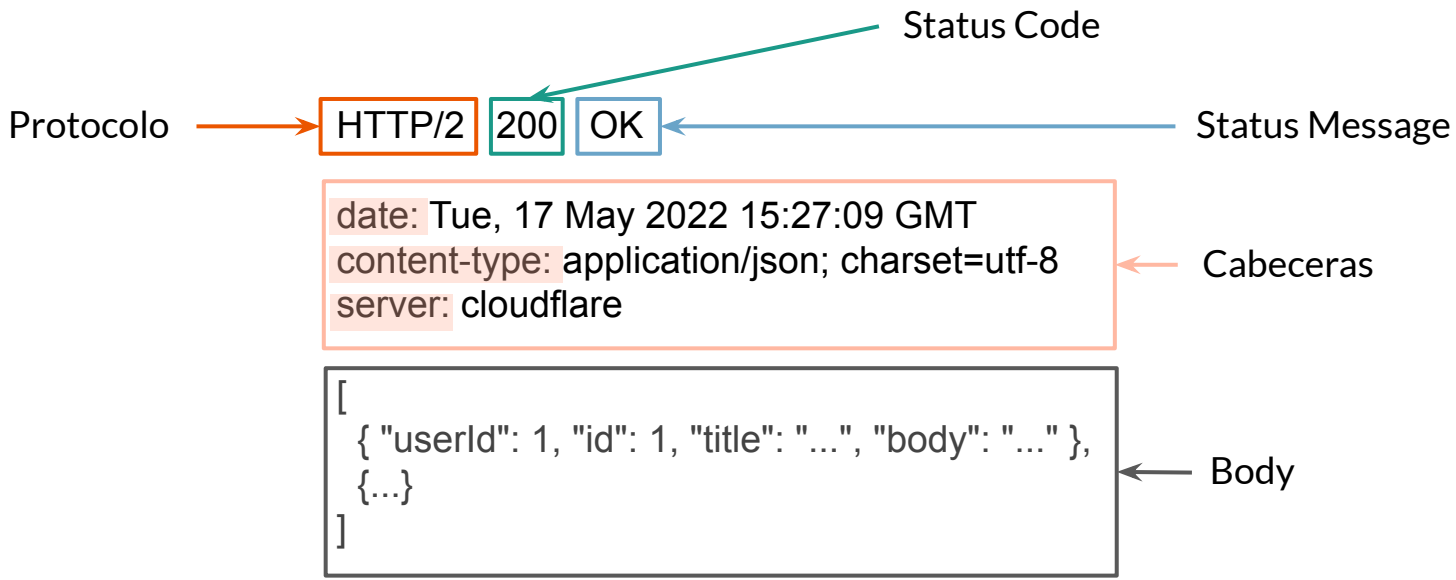
HTTP: HyperText Transfer Protocol

- Transferencia / intercambio de datos en la web a través de documentos
- Protocolo de estructura cliente - servidor
- **Cliente:** Request / Petición
- **Servidor:** Response / Respuesta

REQUEST / Petición



RESPONSE / Respuesta



Fetching

fetch()

- Método para iniciar el proceso de hacer una búsqueda de un recurso en la Web.
- Retorna una promesa (objeto que produce un valor en el futuro)

SINTAXIS

```
fetch(resource)

fetch(resource, options)
```

OPTIONS:

- Method, headers, body, mode, credentials

GET: Get all posts

```
const [state, setState] = useState("loading");
const [posts, setPosts] = useState([]);

useEffect(() => {
  async function loadPosts() {
    try {
      const response = await fetch(
        "https://jsonplaceholder.typicode.com/posts"
      );
      const newPosts = await response.json();
      setPosts(newPosts);
      setState("success");
    } catch (error) {
      console.error(error);
      setState("error");
    }
  }

  loadPosts();
}, []);
```

```
if (state === "loading") return <div>Loading</div>;
if (state === "error") return <div>Error</div>;

return (
  <section>
    {posts.map((post) => (
      <article key={post.id}>
        {post.id} - {post.title}
      </article>
    ))}
  </section>
);
```

```jsx
const [isLoading, setIsLoading] = useState(false)

async function createPost(event) {
  event.preventDefault()
  setIsLoading(true)

  try {
    const form = event.currentTarget
    const newPost = {
      title: form.title.value,
      body: form.body.value,
      userId: 1,
    }

    await fetch('https://jsonplaceholder.typicode.com/posts', {
      method: 'POST',
      body: JSON.stringify(newPost),
      headers: {
        'Content-type': 'application/json; charset=UTF-8',
      },
    })
    form.reset()
  } finally {
    setIsLoading(false)
  }
}
```

POST: Create a new post

```jsx
return (
  <form onSubmit={createPost}>
    <div>
      <label htmlFor="title">Título</label>
      <input id="title" type="text" name="title" />
    </div>
    <div>
      <label htmlFor="body">Contenido</label>
      <textarea id="body" name="body" />
    </div>
    <button disabled={isLoading}>Crear</button>
  </form>
)
```

Ejemplos de Autenticación

Code Sandbox

Axios

Axios

- Promise based HTTP client for the browser and node.js
- Automatic transforms for JSON data

SINTAXIS

```
axios.get(URL)

axios.post(URL, object)
```

Global Configuration:

```
axios.defaults.baseURL = 'https://api.example.com';
axios.defaults.headers.common['Authorization'] = AUTH_TOKEN;
axios.defaults.headers.post['Content-Type'] = 'application/x-www-form-urlencoded';
```

Ejemplos

Code Sandbox

Popular Alternatives

- SWR (swr.vercel.app)
- React Query (react-query.tanstack.com)