Integrating a front-end application with a Backend Server:-

→ Understanding RESTful APIs

- REST (Representational State Transfer) is an architectural style for designing networked applications.
- RESTful APIs use HTTP requests to perform CRUD (Create, Read, Update, Delete) operations on resources.

Key Characteristics:-

- Stateless- Each API call is independent, with no stored context on server b/w requests.
- Resource-Based: Each piece of data (resource) is identified by a URL
- Methods: Common HTTP methods include GET, POST, PUT, DELETE.

→ ~~Making~~ Examples of RESTful API Endpoints:-

- GET /users: Retrieve a list of users
- POST /users: Create a new user
- GET /users/%id%: Retrieve a specific user by ID.
- PUT /users/%id%: Update a specific user by ID
- DELETE /users/%id%: Delete a specific user by ID

→ Making API calls from the Frontend

- Tools & Libraries

- Fetch API
  A built in JavaScript method for making HTTP requests.

- Axios:
  A popular library for making HTTP requests, offering a more powerful & flexible API compared to fetch API.

Code example using Fetch API:-

```
fetch ('https://api.example.com/users')
  .then( response => response.json())
  .then( data => console.log(data))
  .catch (error => console.error('Error:', error));
```

Explanation:

The 'fetch' function makes a GET request to the specified URL. The response is then converted to JSON & logged to the console.

Code Example using Axios:-

```
import axios from 'axios';

axios.get ('https://api.example.com/users')
```

```
  .then ( response => {
      console .log ( response.data);
  })
  .catch ( error => {
      console.error( 'Error:', error);
  });
```

Explanation:

The 'axios.get' method makes a GET request to the specified URL.

The response data is logged to the console.