**Design Decisions:**

1. **Choice of Technology:** I opted for Node.js and Express.js due to their lightweight nature, efficiency in handling HTTP requests, and the ease of building RESTful APIs. These technologies are widely used and well-supported in the developer community.
2. **Single Endpoint:** I designed the API to have a single endpoint **/facts** for simplicity and ease of use. This endpoint allows users to retrieve dog facts with optional query parameters.
3. **Data Storage:** To keep the implementation simple, I stored dog facts directly in the server code as an array. This approach eliminates the need for external databases and simplifies the setup process.
4. **Error Handling:** I included basic error handling to manage invalid requests, such as missing or invalid query parameters. The API responds with appropriate error messages to guide users in providing correct inputs.

**Challenges Faced:**

1. **Parsing Query Parameters:** Understanding how to parse and handle query parameters in Express.js was initially challenging. However, I overcame this by referring to the Express.js documentation and experimenting with different approaches until I found the most suitable solution.