

Following points highlights Trello's confirmation to REST Design principles and constraints:

1. **Client Server Architecture:** The web application sends requests to server, as a result of action done on user interface by the user, and in response, server provides meaningful response containing HTTP Status codes, and response messages which client can consume and process accordingly.
2. **Layered:** Trello follows layered architecture with concrete separation of Presentation Layer, Application layer and Data Layer. This has helped Trello to achieve Separation of Concerns and modularity.
3. **Stateless:** A single request sent from Trello UI is independent and is self-contained. Server doesn't have to remember any previous requests made from browser. Trello Client sends cookies along with the requests as authentication information, which further proves that the implementation is stateless.
4. **Cacheable:** HTTP Response sent from server contains Cache-Control headers with different values of max-age. This proves that the resources responded by server are cacheable.
5. **Uniform Interface:** Requests are sent using common HTTP methods like GET, POST, PUT & DELETE, the resources are identified with URI's, the request and response bodies are written in JSON format. This format is consistent across the application.