Wieso ein gutes API Design wichtig ist

Salame, Ahmed
Hochschule Mannheim
Fakultät für Informatik
Paul-Wittsack-Str. 10, 68163 Mannheim

***Zusammenfassung*—Dieses Paper stellt Kriterien für ein gutes Application Programming Interface (API) Design. Es zeigt ebenso was schlechtes Design ausmacht anhand von paar Beispielen. Es werden verschiedene Ansätze aufgezeigt, mit dessen Hilfe die usability einer API verbessern kann.**
—TODO—

Inhaltsverzeichnis

1. Einleitung

[1] Es soll hier erstmal erklärt werden, wie wichtig eine API ist für den Entwickler.

2. Application Programming Interface

2.1. Die Definition einer API

Hier wird erstmal grob erklärt was eine API ist und wo es überall genutzt wird. Es soll die unterschiedlichen Typen einer API aufzeigen.

2.2. Die Vorteile einer API

Hier werden die Vorteile einer API aufgezeigt.

3. Probleme mit APIs

Hier soll es eine Erklärung geben für einige Negativ Beispiele.

3.1. Mangelnde Dokumentation

Hier soll es einige Beispiele geben, die erklären sollen welche Probleme es gibt. Dabei soll Bezug genommen werden auf bekannte Probleme wie unglücklich gewählte Methodennamen ‚Parameter etc. Dabei werden hier Beispiele aus GWT und LibGDX genommen. Danach wird auf potenzielle Lösungen Eingegangen (siehe Abschnitt 5).

3.2. Falsches Nutzen der API

Hier soll aufgezeigt werden, welche folgen es haben kann, falls eine API nicht richtig genutzt wird, weil es für den Nutzer nicht ersichtlich war, was nun genau vom Entwickler vorgesehen war, wie man die Methoden richtig nutzen soll. Es soll auf die Sicherheitslücken hinweisen.

4. Ansätze zur Verbesserung allgemeiner APIs

Hier sollen potenzielle Lösungen aufgezeigt, wie man allgemein eine Bessere API bauen kann.

4.1. Umdenken der Menschen

Hier soll es einen Ansatz geben, der sich damit befasst, dass man die usability einer API nicht unterschätzen sollte. Viele nehmen das nicht ernst und entwickeln nicht sorgfältig genug.

4.2. Nutzer freundlicheres Design

Hier soll es eine erklärung geben, was genau die meisten Nutzer etwas als besonders benutzerfreundlich betrachten. Dabei soll auch Beispiele aufgezeigt werden, wie man sowas ermittelt

5. Ausgewählte API Entwurfs Beispiele

Hier soll es einen Auszug geben, welche möglichen Lösungen es zu ausgewählten Probleme gibt. Hier soll es unter anderem einige Beispiele geben, die typisch sind

für clean code (Methodennamen, Parameter, namens Konventionen etc). Um zu verdeutlichen was genau gemeint ist, wird dabei Java/C++ code als Beispiel genommen. Dieser Abschnitt soll sich mehr mit der Technischen Seite befassen.

5.1. Methodennamen und ihre Parameter

Hier soll es erst mal ein Beispiel geben, welche Probleme auftreten könnten, wenn man sich bei dem schreiben einer Methode keine Gedanken macht, z.B. falsche Parameter übergaben oder zu viele Parameter.

5.2. Keine redundanten Methoden

Hier soll es aufzeigen, wieso man nicht für alles eine Methode schreiben sollte, wie z.B. beim einlesen von einem Images.

6. Fazit

Hier soll es nochmal einen Bezug auf die vorherigen Lösungsbeispielen geben und was für Vorteile es einem bringt.

Abkürzungen

API Application Programming Interface

Literatur

[1] Joshua Bloch. *A Brief, Opinionated History of the API*. 19. Nov. 2017. URL: https://www.youtube.com/watch?v=ege-kub1qtk.