

Plan van aanpak chatroom API

Ewout Eggens

1 Introductie

In dit document zal ik de planning voor mijn API, die bij de voorheen gemaakte chatroom hoort, toelichten.

2 Opdracht

Bij deze opdracht dient er een API in PHP gebouwd te worden voor de chatroom die in de vorige weekopdracht gemaakt is. Bij deze opdracht zijn er drie verschillende "tiers" aan eindproduct: (in oplopende volgorde)

- **tier 1: fixed response**
De server stuurt een vaste response terug, ongeacht de parameters die je meegeeft in de url.
- **tier 2: flatfile datastructure**
De server reageert op de parameters die meegegeven worden in de url en slaat de serverside gegevens op in een tekstbestand.
- **tier 3: SQL database**
De server reageert op de parameters die meegegeven worden in de url en slaat de serverside gegevens op in een SQL database.

3 Aanpak

Gezien mijn niveau op het gebied van PHP en het feit dat ik niet bekend ben met SQL leek het mij verstandig op niveau 2 in te stromen. Ik wil een API maken die in grote lijnen hetzelfde functioneert als de API die we in week 1 van Codegorilla hebben gekregen. Dit houdt in:

- Berichten op de server zetten aan de hand van een 'key' en een 'id'.
- Een specifiek bericht uit de server halen die hij herkent aan het 'id' en verifiëert of deze bij de juiste 'key' hoort'.
- Een lijst uit de server halen waar alle 'ids' worden geleverd die bij een bepaalde 'key' horen.

Als ik dit 'tier 2' product af heb dan zal ik mijn tijd besteden om mijn kennis over SQL op het nodige pijl te krijgen om 'tier 3' te behalen. Daarna zal ik deze kennis implementeren.

Het voordeel van deze aanpak is dat ik niet veel aan de front end hoeft te sleutelen, ik kan dan zoveel mogelijk focusen op PHP en SQL.

Mocht ik zo ver komen dat ik een SQL database heb, dan kan ik kijken of ik de API, en daarmee de front-end kan uitbreiden. Mogelijke toevoegingen zijn:

- Users met hun eigen passwords.
- Chatrooms.
- Lijst met users die recent online waren.
- Eventuele andere toevoegingen.