

Codex Implementation Guide: Best Practices, Fehleranalyse & Verbesserungen

Diese Anleitung beschreibt, was Codex basierend auf der Analyse aller 13 Hauptfunktionen im Projekt `/try` umsetzen soll. Ziel: maximale Stabilität, Sicherheit, Performance, vollständige Internationalisierung und saubere Struktur.

1. Reaction-based Event Sign-up

- Auf Buttons oder Slash-Commands umstellen, Reaktionen ersetzen.
- An-/Abmeldung persistent in DB speichern.
- Bei Bot-Neustart aktuelle Reaktionen erneut einlesen.
- Teilnahme/Abmeldung mit sichtbarer Rückmeldung.
- Mehrsprachige Bestätigung implementieren.

2. Newsletter Opt-Out Command

- Als Slash-Command implementieren: `/newsletter optout` und `/newsletter optin`.
- Persistent speichern (DB); global oder serverabhängig klären.
- Mehrsprachige Rückmeldung + Wiederanmeldung ermöglichen.
- Tests: User wird zuverlässig abgemeldet und nicht mehr benachrichtigt.

3. Reminder Timing (1h / 10min)

- Reminder-Zeitpunkte dynamisch planbar machen.
- Erinnerung auch bei Bot-Restart triggern (persistente Jobs).
- @role mention optional, aber sauber konfigurierbar.
- Reminder in `t(...)` lokalisieren.

4. Daily Newsletter at 08:00 UTC

- Automatisierter Versand mit Zeitprfung.
- Inhalt ber eigene Funktion ``createDailyNewsletterContent()``.
- Rate-Limits beachten, personalisierte Inhalte mglich machen.
- Mehrsprachige Texte, Datum formatiert in Nutzer-Sprache.

5. `get_events_for()` & `format_events()` Helpers

- Effiziente DB-Queries in ``get_events_for()``.
- Lokalisierte Formatierung (z.B. Datum, Uhrzeit).
- ``format_events()`` mit Variationen (short/long).
- Tests fr leere Liste, Einzelfall, Pluralisierung.

6. Secure File Upload Route

- Whitelist fr Dateitypen, MIME-Check, Size-Limit.
- CSRF-Schutz, Pfadvalidierung, zufllige Dateinamen.
- Uploads nicht direkt aus Webroot ladbar machen.
- Antivirus-Scan optional bei ffentlichen Dateien.

7. Google Calendar Sync

- Nur notwendige Scopes, OAuth2 + Token Refresh.
- Google-Event-ID in DB speichern Duplikate vermeiden.
- Fehlerbehandlung mit Backoff, Sync-Tokens verwenden.
- Format RFC3339 + UTC-Sicherheit prfen.

8. WebSocket Live Updates

- Authentifizierung bei Verbindungsaufbau.

- Rume je nach Rolle oder Thema.
- Sicherheitschecks: CSRF, Origin, Rate-Limiting.
- Ping/Pong + Logging + Retry-System.

9. Role-Based UI Visibility

- Alle Rollen-Abfragen zentralisiert.
- Backend-Prfung zusätzlich zum UI-Hiding.
- Kein Zugriff auf geschützte Daten via direkte API-URL.
- Rollen konfigurierbar halten (nicht hardcoded).

10. Responsive Design

- Flexbox/Grid nutzen, Layout dynamisch auf mobile.
- Keine fixen Breiten, alles in % oder vw/vh.
- Viewport-Tag setzen, Mobile Breakpoints testen.
- Mobile UX (Buttons, Spacing) beachten.

11. Monthly Champion Selection

- Automatisierte Auswertung am Monatsende.
- Historie & Gründe sichtbar machen.
- Gleichstand behandeln, Rolle automatisch rotieren.
- Champion-Benachrichtigung via DM.

12. Resources & Guides Section

- Guides aus DB oder Markdown-Dateien laden.
- Suchfeld, Filter, Kategorien hinzufügen.
- Nicht-öffentliche Inhalte rollenbasiert absichern.
- Mehrsprachige Inhalte & Uploads absichern.

13. Vollständige Internationalisierung

- Keine Klartexte im Code (auch JS).
- Nur `t("...")`, `i18n.t(...)`, `{t("...")}` verwenden.
- Pluralformen beachten, Zeitformate lokalisieren.
- Pseudo-Translation zur Erkennung fehlender Keys.

Ziel:

Alle Punkte jetzt vollständig, konsistent, sicher und mehrsprachig umsetzen. Danach System auf Produktionsqualität testen.

> Codex: Beginne mit dem Umbau nach dieser Checkliste kein Punkt darf fehlen.

14. Reminder (öffentlich, sprachlich, sauber getaktet)

- Reminder wird **max. 1x pro Stunde pro Channel** gesendet.
- Reminder erfolgt **nur**, wenn für den Channel offene Aufgaben vorliegen.
- Reminder wird **nicht als DM**, sondern in einem öffentlichen Discord-Channel gepostet.
- Reminder-Text ist **immer auf Englisch**, da keine pro-User-Lokalisierung im Channel möglich ist.
- Sprachtext über festen Key `reminder_hourly` in `en.json`:

```
```json
```

```
"reminder_hourly": "Reminder: Its {time} UTC check your tasks!"
```

```
```
```

- Keine Verwendung von `t(...)` mit Channel-Sprache notwendig Reminder ist **systemisch englisch**.

Codex implementiert:

- Zeitprfung & Duplikatvermeidung (pro Stunde, pro Channel).
- Aufgabenprfung via `get_open_tasks(channel_id)`.
- Logging & Debug-Modus bei Reminder-Testlufen.