



HITOBITO CHEATSHEET

Version 21-03-06

Hitobito besteht aus dem Core (hitobito) und organisations-spezifischen Wagons (hitobito_youth, hitobito_pbs, hitobito_cevi, ...).

Core

Normales Ruby on Rails Projekt. Hier sind alle gemeinsamen Funktionalitäten definiert.

Routen

config/routes.rb

Definieren, welche URLs in der Applikation erreichbar sind und verbinden diese mit Controllern und Actions darin (z.B. PeopleController#create).

Wagons

Organisations-spezifische Änderungen, wie Gruppen- und Rollentypen, Features und Exporte

Controllers

app/controllers/*.rb

Definieren die Logik, was passieren soll wenn eine URL aufgerufen wird, und rendern via Views HTML.

- ✓ Weitere Routen ergänzen
- ✓ Core-Routen überschreiben (einzelne Routen)

Views

app/views/*.haml

Sind mehr oder weniger HTML-Code mit Platzhaltern, welche mit Daten aus der DB und mit übersetzten Texten befüllt werden.

- ✓ Bei render_extensions einhängen
- ✓ Neue Views hinzufügen
- ✓ Core-Views überschreiben (nur ganze Datei)

Übersetzungen

config/locales/*.yml

Jede Übersetzung hat einen technischen Key, der entsprechend dem Verwendungsort aufgebaut ist (z.B. de.events.form_tabs.general)

- ✓ Core-Übersetzungen überschreiben (einzelne Keys)
- ✓ Neue Übersetzungen hinzufügen

Models

app/models/*.rb

Definieren, welche Objekte in hitobito gespeichert werden können, sowie Beziehungen (z.B. Person hat Rollen, Rolle gehört zu Gruppe) und Validierungen (z.B. Name darf nicht leer sein).

- ✓ Core-Models anpassen via lib/hitobito_*/wagon.rb
- ✓ Neue Models hinzufügen

Decorators

app/decorators/*.rb

“Hülle” um ein Model, enthält Model-spezifische Berechnungen (z.B. full_name = first_name + last_name)

- ✓ Core-Decorators anpassen via lib/hitobito_*/wagon.rb
- ✓ Neue Decorators hinzufügen

Helpers

app/helpers/*.rb

Wo zu viel Logik in den Views wäre, packen wir die in einen Helper (z.B. wiederverwendbare Rollen-Anzeige inklusive Ebene und Bezeichnung)

- ✓ Neue Helpers hinzufügen
- ✓ Core-Helpers überschreiben (nur ganze Methode)

Domain-Klassen

app/domain/*.rb

Für komplexere Berechnungen und Business-Logik (z.B. PDF-Export, Duplikats-Prüfung, Gruppen fusionieren)

- ✓ Core-Domain-Klassen anpassen via lib/hitobito_*/wagon.rb
- ✓ Neue Domain-Klassen hinzufügen

Specs (Unit Tests)

spec/*.rb

Automatisierte Tests für alles was denkbar wäre, in Zukunft mal kaputt zu gehen.

- ✓ Neue Specs hinzufügen

... Weiteres

Abilities (Berechtigungen) app/abilities/*.rb

CSS Styles app/javascript/stylesheets/*.css

JavaScript app/javascript/javascripts/*.coffee

Datenbank-Migrationen db/migrate/*.rb

Konfigurationsdateien config/*.rb

- ✓ Neue Abilities hinzufügen und bestehende überschreiben
- ✓ Neue CSS-Styles hinzufügen
- ✓ Neue JavaScript Scripts hinzufügen
- ✓ Neue Migrationen hinzufügen
- ✓ Vorgesehene Config überschreiben