

1. **Projekt erstellen:** rails new <projektname>
2. **Gem Foundation einbinden:**

```

Ordner von der App/Gemfile
gem 'foundation-rails', '~> 5.5'
group :development do
  gem 'rails_layout'
end
Optional:
gem 'foundation-icons-sass-rails'

```
3. **Dannach** einen „bundle install“ ausführen
4. **Foundation aktivieren:** rails generate layout:install foundation5 --force
5. **Model und Controller anlegen:** rails g resource <Name-Mehrzahl> title:string description:text start_date:datetime
Resource besteht aus Model und Controller (keine Actions im Controller)
6. **In Datenbank schreiben:** rake db:migrate
7. **Index-Action im Controller anlegen:**

```

def index
  @projects = Project.all
end

```
8. **Index.html.erb im View/Project erstellen:**

```

<h1>
  <%if @projects.size == 1 %>
    <%= "Projekt" %>
  <% else %>
    <%= "Projekte" %>
  <% end %>
</h1>

<table>
  <thead>
    <tr>
      <th>Title</th>
      <th>Description</th>
      <th>Start Date</th>
      <th>Tasks</th>
    </tr>
  </thead>
  <tbody>
    <% @projects.each do | var | %>
      <tr>
        <td><%= var.title %></td>
        <td><%= var.description %></td>
        <td><%= var.start_date %></td>
        <td><%= link_to "Show", project_path(var.id) %></td>
        <td><%= link_to "Edit", edit_project_path(var.id) %></td>
        <td><%= link_to "Destroy", project_path(var.id), method: :delete, data: {confirm: "Sind sie sicher?"} %></td>
      </tr>
    <% end %>
  </tbody>
</table>

```
9. **Route anlegen (routes.rb):**

```

root "projects#index"
resources :projects

```
10. **Model "Datenbank" Tabelle "Project" mit Inhalt füllen:**

```

Project.create(title: "Test", description: "Test", start_date: "2016-03-03")
oder
a = Project.new(title: "Test", description: "Test", start_date: "2016-03-03")
a.save

```
11. **New-Action, Create-Action und new.html.erb in View/Project anlegen!**
12. **Projekt_controller eintrag:**

```

before_action :set_project, only: [:show, :edit, :update, :destroy]
def new
  @project = Project.new
end

def create
  @project = Project.new(project_params)
  if @project.save
    flash[:notice] = "Eintrag erfolgreich erstellt :)"
    redirect_to projects_url
  else
    flash[:alert] = "Leider ist etwas schief gegangen :("
    render :new
  end
end

```

```
private
def project_params
  params.require(:project).permit(:title, :description, :start_date)
end

def set_project
  @project = Project.find(params[:id])
end
```

- 13. new.html.erb ergänzen:**

Neues Projekt

<%= render "form" %>

14. index.html.erb ergänzen:

```
<%= link_to "Neues Project anlegen", new_project_path %>
```

- 15. Formular in View/Project anlegen (_form.html.erb):**

[illegible]

16. Validations eintragen (Im Model/project.rb):

```
validates :title, length: { minimum: 5, message: "Der Titel muss mindestens 5 Zeichen enthalten!" }
validate :start_date_cannot_be_in_the_past

def start_date_cannot_be_in_the_past
  errors.add(:start_date, "Das Datum muss in der Zukunft liegen!") if !start_date.blank? && start_date
    < Date.today
end
```

Der Befehl "validate" ruft die Methode auf die hinter ihm steht, in diesem Falle ":start date cannot be in the past"

17. Im View/Project/_form.html.erb die Fehler anzeigen lassen:

```
<% if @project.errors.any? %>
<h3>Opps! Der <%= @project.class.name.titleize.downcase %> konnte nicht gespeichert werden.</h3>
Korrigieren sie folgenden <%= pluralize(@project.errors.count, "error") %>:
<ul>
<% @project.errors.full_messages.each do |message| %>
<li><%= message %></li>
<% end %>
</ul>
<% end %>
<div class=.....
```

- 18. Error-Anzeige auslagern:**

Neuen Ordner in den Views -> "shared"

19. Neue Dateien errors.html.erb:

```
<% if object.errors.any? %>
  <h3>Oops! Der <%= object.class.name.titleize.downcase %> konnte nicht gespeichert werden.</h3>
  Korrigieren sie folgenden <%= pluralize(object.errors.count, "error") %>:
  <ul>
    <% object.errors.full_messages.each do |message| %>
      <li><%= message %></li>
    <% end %>
  </ul>
```

```
<% end %>
```

20. **_errors.html.erb in _form.html.erb Rendern:**

```
...form_for(@project) do |f| %>
  <%= render "shared/errors", object: @project %>
</div class=.....
```

21. **Show, Destroy, Update und Edit im Controller Einrichten:**

```
def edit
end

def update
  if @project.update(project_params)
    flash[:notice] = "Update erfolgreich :)"
    redirect_to projects_url
  else
    flash[:alert] = "Leider ist etwas schief gegangen :("
    render :edit
  end
end

def destroy
  @project.destroy
  redirect_to projects_url
end

def show
end
```

22. **edit.html.erb im View/Project erstellen:**

```
<h2>Editieren von <%= @project.title %></h2>
<%= render "form" %>
```

23. **show.html.erb im View/Project erstellen:**

```
<h1><%= @project.title %></h1>

<p><%= @project.description %></p>

<p><%= @project.start_date %></p>

<p>
  <ul class="button-group">
    <li>
      <%= link_to "Alle Projekte anzeigen", projects_path, class: "button" %>
    </li>
  </ul>
</p>
```

24. **Verlinkungen in der View/Project/Index.html.erb einfügen, wenn nicht bereits vorhanden!**

```
<% @projects.each do | var | %>
  <tr>
    <td><%= var.title %></td>
    <td><%= var.description %></td>
    <td><%= var.start_date %></td>
    <td></td>
    <td><%= link_to "Show", project_path(var.id) %></td>
    <td><%= link_to "Edit", edit_project_path(var.id) %></td>
    <td><%= link_to "Destroy", project_path(var.id), method: :delete, data: {confirm: "Sind sie sicher?"} %></td>
  </tr>
<% end %>
```

25. **Per Scaffold die verschachtelte und referenzierte Resource anlegen:**

```
rails g scaffold tasks title description:text start_date end_date project:references

rake db:migrate
```

26. **In der routes.rb die Ressourcen verschachteln:**

```
resources :projects do
  resources :tasks
end
```

27. **In der Show von Projects einen Link zu den Tasks hinzufügen:**

```
<%= link_to "Tasks", project_tasks_path(@project.id), class: "button" %>
```

28. **Im Tasks Controller eine neue before_action anlegen:**

```
before_action :set_project

private
def set_project
  @project = Project.find(params[:project_id])
end
```

29. **Index.html.erb von View/Tasks anpassen:**

```
<p id="notice"><%= notice %></p>
```

```

<h1>Listing Tasks for <%= @project.title %></h1>

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
<th>Start date</th>
<th>End date</th>
<th colspan="3"></th>
</tr>
</thead>

<tbody>
<% @tasks.each do |task| %>
<tr>
<td><%= task.title %></td>
<td><%= task.description %></td>
<td><%= task.start_date %></td>
<td><%= task.end_date %></td>
<td><%= link_to 'Show', project_task_path(@project.id, task.id) %></td>
<td><%= link_to 'Edit', edit_project_task_path(@project.id, task.id) %></td>
<td><%= link_to 'Destroy', project_task_path(@project.id, task.id), method: :delete, data: { confirm: 'Are you sure?' } %></td>
</tr>
<% end %>
</tbody>
</table>

<br>

<%= link_to 'New Task', new_project_task_path(@project.id) %>

```

30. Im Model/Project.rb anpassen:

```
has_many :tasks, dependent: :destroy
```

31. Im Model/Tasks.rb anpassen:

```
belongs_to :project
```

32. Im View/Tasks/_form.html.erb anpassen:

```

<%= form_for [@project, @task] do |f| %>
<% if @task.errors.any? %>
<div id="error_explanation">
<h2><%= pluralize(@task.errors.count, "error") %> prohibited this task from being saved:</h2>

<ul>
<% @task.errors.full_messages.each do |message| %>
<li><%= message %></li>
<% end %>
</ul>
</div>
<% end %>

<div class="field">
<%= f.label :title %><br>
<%= f.text_field :title %>
</div>
<div class="field">
<%= f.label :description %><br>
<%= f.text_area :description %>
</div>
<div class="field">
<%= f.label :start_date %><br>
<%= f.datetime_select :start_date %>
</div>
<div class="field">
<%= f.label :end_date %><br>
<%= f.datetime_select :end_date %>
</div>
<div class="actions">
<%= f.submit %>
</div>
<% end %>

```

33. Im View/Tasks/new.html.erb link_to ergänzen:

```
<%= link_to 'Back', project_tasks_path %>
```

34. Im Controller/tasks_controller.rb create action anpassen

```

def create
  @task = @project.tasks.new(task_params)

  respond_to do |format|
    if @task.save
      format.html { redirect_to project_tasks_path(@project.id), notice: 'Task was successfully created.' }
      format.json { render :show, status: :created, location: @task }
    end
  end
end

```

```
      else
        format.html { render :new }
        format.json { render json: @task.errors, status: :unprocessable_entity }
      end
    end
  end
end
```

35. Im View/Tasks/show.html.erb link_to ergänzen:

```
<%= link_to 'Edit', edit_project_task_path(@project.id, @task.id) %> |
<%= link_to 'Back', project_tasks_path %>
```

36. In View/Tasks/edit.html.erb link_to ergänzen:

```
<%= link_to 'Show', project_task_path(@project.id, @task.id) %> |
<%= link_to 'Back', project_tasks_path(@project.id) %>
```

37. Im controller/tasks_controller.rb update action ergänzen:

```
def update
  respond_to do |format|
    if @task.update(task_params)
      format.html { redirect_to project_task_path, notice: 'Task was successfully updated.' }
      format.json { render :show, status: :ok, location: @task }
    else
      format.html { render :edit }
      format.json { render json: @task.errors, status: :unprocessable_entity }
    end
  end
end
```

38. Im controller/tasks_controller.rb destroy und def new ergänzen:

```
def new
  @task = @project.tasks.new
End

def destroy
  @task.destroy
  respond_to do |format|
    format.html { redirect_to project_tasks_url, notice: 'Task was successfully destroyed.' }
    format.json { head :no_content }
  end
end
```

39. Und zum Ende wollen wir nun den Task zum jeweiligem Project zuweisen!
Im controller/tasks_controller.rb index action ergänzen:

```
def index
  @tasks = @project.tasks
end
```