Basic doc(group)

## Notes:

1. Make sure you understand the topic's depth properly *even if the linked resources don't cover it properly*, you should be covering them properly. For example, the linked directory structure video skips some directories, you are not supposed to skip. Linked resources are only for help, not the golden rule of thumb for depth of topics.
2. On day 1, you will build a simple rails blog app, you can take this app and continue adding all kinds of add-ons(devise, bootstrap etc etc) for all the topics you cover in the next few days. You can even decide to build some other app for your learning along the way. But everything should be properly practiced.
3. It will be assumed that all topics in the guide are done with hands-on practice. So make sure this is done properly.
4. In standups, mention topics, not only the day numbers.
5. Keep sending the PR links in standups.

### Before you begin - Basics of web:

1. If you are not comfortable with basic web concepts, or you didn't had chance to take a basic web course in your university education, you can use the following resources to learn the basics of web development concepts:
2. Web Application Development
3. How the Web Works: A Primer for Newcomers to Web Development (or anyone, really) (Read both parts of the above article, link to second part is at the end of the provided link)
4. **Must know:** Http vs https, **UDP** vs TCP, DNS, Request life cycle, Different request types(get, post, put etc) and their differences, application server vs web server, how encryption works vs how hashing works etc. What are HTTP headers, what are comment headers?
5. **Good to know**(above average): Details of how SSL/**HTTPS** encryption actually works, how handshake works in http, and what nameservers are.
6. **You should be comfortable with at-least these basics of web development before you begin proper training outline.**

# Day 1 (Intros, know-hows)

## Learning Goals:

- Get familiar with the framework and its norms, its **Components**, **Architecture** and build your first rails app.
- Also set the tone for the depth and velocity needed in the rest of the timeline.

## Content:

### Basics:

1. Rails' basic history and philosophy know-how.
2. Go through the rails github page, build basic know-how around it and read through any interesting pages of rails wiki you want to(**especially skim through the "Frameworks and libraries" section, explore components**).
3. Rails directory structure (in depth, **all folders covered or not in the linked video**, find and use other resources as needed for this understanding) (In depth directory Structure)
4. Learn About MVC – How ROR implements MVC?
5. Intro to Rails: What is Ruby on Rails?
6. Dive into the framework, use the rails guide(5.2) and get your first rails app up and running quickly. (go through the main page, don't go through all the linked pages)
   **Don't try to remember everything, just go along. By the end you will create your first ROR application.**
   Practice the following tasks:
   1. Generating Models, Controllers, Migrations
   2. Rails Scaffolding, types, how, why (just a know how, its use is not advised in training period)
   3. Data **Seeding**, why, what data etc
   4. Rails/**Rake** Tasks (e.g rails db:create, rails db:migrate, rails db:**seed** db:**rollback**)
      i. All Rails db Rake Tasks and What They Do
   5. Gems, what is gemfile, gemfile.lock etc (Try to install devise gem)
      i. #201 Bundler (revised)
7. How to choose open source libraries (Generic)

### More:

1. Rails versioning / Maintenance Policy for Ruby on Rails
2. #66 Custom Rake Tasks (why, how)

## Day 2 (Intros, know hows, level 2)

### Learning Goals:

- Get familiar with some advanced topics, and how to use css/scss and different flavors and libraries of JS. (Assumption: you know basic JS, advanced concepts not needed)
- Get full hands on practice of UI manipulation in Rails way.

### Content:

NOTE: If you are finding it difficult to grasp basic flows of rails, or understanding how it works, you can watch the following tutorial, it is actually linked in routing, skipping it for now is fine:
Rails Routes - Detailed walkthrough

#### Basics:

1. Revise all previous days topics(go through)
2. Ruby on Rails' basic Architecture
3. Add bootstrap gem and do some work on CSS and JS. Explore SCSS, play around with it.
    a. #268 Sass Basics
    b. How to configure sass/scss in rails
4. Dive in JS and explore different use cases of JS(within ruby on rails context). Try to handle some DOM events in your rails app. help(official)
5. **Asset Pipeline**
    a. Understanding Asset Pipeline and The Asset Pipeline - Ruby on Rails Guides
6. Add Bootstrap and **Javascript/Coffeescript**, read basic of following
    a. *If you don't know JS, this is a good time to learn its basics as well*
    b. http://www.w3schools.com/bootstrap/
    c. #390 **Turbolinks**
7. HashWithIndifferentAccess

#### More:

1. Fingerprinting, load paths, pre-compilation, minification etc(Explore the purpose of these features and benefits).
2. What is *transpilation*
3. jQuery Learning Center (only basics, what its purpose is, what it's made for)

Good to follow if you completed above before expected time:

BUILD A REAL ESTATE / PROPERTY APP - RUBY ON RAILS TUTORIAL

## Day 3 (AR and AM Basics)

### Learning Goals:

- Complete practical and conceptual walk through of “models” and migrations in RoR. When to use what approach and what to avoid, how to achieve all kinds of validations and callbacks.
- Also revise and polish basic concepts of DB like indexes etc.

### Content:

#### Basics:

1. Revise all previous days topics(go through)
2. Active Model basics and some advanced concepts
3. What are ORMs (object relational mapping), and rails’s ORM
4. **Migrations:** http://api.rubyonrails.org/classes/ActiveRecord/Migration.html
    1. http://guides.rubyonrails.org/migrations.html
    2. Generating migrations from console
    3. Creating **indexes(which columns to index, and why, pros cons)**, define data types, create tables and associations etc
5. Activerecord validations [1]
    1. Validate vs validates, custom validators(**how to pass/fail custom validation**), numeric, regex match, custom error messages
6. Model callbacks
    1. Sequence of callbacks(hands-on), before/after_commit vs before/after_create/save, before/after_validation etc

**Help:**

http://api.rubyonrails.org/classes/ActiveRecord/Callbacks.html
Ruby on Rails Tutorial Part 9 - Validation & Error Messages

#### More:

1. Up vs down vs change
2. around_save, validates_with
3. **ActiveRecord Dirty module: Get old values of attributes in callbacks**, check if attribute changed(in before save vs after save), append errors (invalidate record)

[1] Just read basic validates/validate and practice them, for all other helpers, get basic familiarity of them

# Day 4 (Controllers)

## Learning Goals:

- Complete practical and conceptual coverage of controllers concepts, you would have already seen them but today we will cover them in and out, with complete practice.
- Also revise and polish basic concepts of network/http etc like HTTP body vs headers and their usage etc.

## Content:

### Basics:

1. Revise all previous days topics(go through)
2. Learn about general Rails naming conventions (controller names, Model names, variable names).
3. Explore and understand controllers, ins and outs, with practice
    1. Action Controller Overview — Ruby on Rails Guides (Spend most of the day here, with practice)
    2. **Controller callbacks** (already covered in the Guides link)
    3. **Base controller concept, fat vs slim controllers, serializer,** strong params**(require vs permit vs permit!)**, etc. (already covered in the Guides link)
    4. Fun With The Controller - Ruby On Rails Friend List App #11
    5. **Request types**(html/js/json etc) and how to handle in rails
    6. Practice EVERYTHING in this section, play around
4. **Rails Architecture Understanding**
    1. ERB (Embedded Ruby) basics
    2. Explore Gemfile and Gemfile.lock and its purpose, different use cases of JS(within ruby on rails context)
5. **AR Associations (Basics):**
    1. Active Record Associations — Ruby on Rails Guides (Just get a skim through, and only cover following topics in detail for now, rest will be covered later, get a general idea, take notes of what you could not practice today)
    2. **Has_many, belongs_to, has_one associations**. (must practice)

### More:

1. CSRF tokens(what why how), what are HTTP headers(like request_type)
2. **How to define versions in Gemfile**, what are groups in Gemfile and other options available

# Day 5 (Routing ins and outs + Basic associations)

## Learning Goals:

- Complete practical and conceptual walk through of Routing RoR. When to use what approach and what to avoid, how to achieve all kinds of routing needs.
- Also Some basic AR associations and their different options.

## Content:

### Basics:

1. Revise all previous days topics(go through)
2. Rails Routing from the Outside In — Ruby on Rails Guides [2](Spend most of the time here, with practice)
3. RESTful Routing in Rails
4. Rails Routes - Detailed walkthrough (follow along practically)
5. Explore basic routes - Routes mapping to controller actions
    1. Intro to Rails: Adding Functionality (walks through adding a complete function)
6. Explore routes.rb file and grasp related basic concepts e.g nested routes, Restful rails, Route types.
7. **AR (Continued):**
    1. Practice is a must, with an eye on output being produced(query being run internally, which is logged on the console, and understanding why)
    2. How to run 'WHERE IN [1 ,2]' type queries the rails way
    3. **scopes**, **enums** (with practical use cases + understanding and practice)

### More:

1. Routes  namespaces vs scopes, routing concerns, **mount**
2. Explicit/custom **foriegn keys**, **"through"** option in associations(how it works). **"dependent"** option modes(in has_many)
3. exists? Vs present?

[2] In start, skip the topics from **More** section in this link

## Day 6 (ActionView)

### Learning Goals:

- Complete practical and conceptual understanding of Views building in RoR. When to use what approach and what to avoid, how to achieve all kinds of view needs.
- Optimization techniques and best practices around views.

### Content:

**IMPORTANT: Avoid scaffolding in this section's practice**

#### Basics:

1. Revise all previous days topics(go through)
2. Explore views[3]
3. Learn about view partials
4. What are view helpers(basic helpers) and how to use them in views
5. Layouts and Rendering in Rails
   Layouts and Rendering in Rails — Ruby on Rails Guides
6. Action View Form **Helpers**
   Action View Form Helpers — Ruby on Rails Guides (get an idea, don't remember all helpers)
    1. Partial vs view vs layout etc
    2. **Passing data** from controller to view, to partials, where to create partials.
    3. **Remote form submission**(practice is must), form_with vs form_for vs form_tag, custom form routes.
    4. Loops in views, conditional rendering, AJAX rendered views etc
7. **Pagination:** Use kaminari / pagy or will_paginate for pagination. Practice 1, no need to learn all. (though see what are basic differences)
8. Best practices in rails views, avoiding queries in views, N+1s

#### More:

1. Render a collection through partial(like render all comments using a partial, without custom loop)
2. Different options for pagination, **frontend pagination vs backend pagination, when to use which**
3. **What are view caches**, **how to pass data from view file to layout** etc

[3] Basics of views with practice and how things work, no need to remember everything, get conceptual understanding of how things work together in a way to be able to find and use these things when needed.

# Day 7 (AR, Level 2)

## Learning Goals:

- Activerecord COMPLETE: Associations, advanced topics. Focus on practical learning, think critically on where to use what and focus on implementation details to learn properly.
- Advanced AR concepts and their practical use, learn advanced Querying, and many helpful built-in query methods provided by rails(and where to use them).

## Content:

### Basics:

1. Revise all previous topics
2. Revise EVERYTHING relating to AR properly from previous days(content is spread out on different day, revise all of that)
3. Active Record Associations — Ruby on Rails Guides[4]
4. **Polymorphic**, **many to many**, associated record creation(user.comments.new)
    1. Query method documentation (skim through, get familiar with things)
    2. Many to Many in rails way (When we CANNOT use it)
    3. Through Association
    4. Polymorphic Associations

1. Find by SQL, Joins, Groups, Order
    1. Active Record Query Interface — Ruby on Rails Guides
    2. More help: AR Queries basics AR Queries advanced
2. **Select vs Pluck**, **includes**, joins, left_joins, aggregate functions, having etc. N+1 query problem, Find vs **find_by**, where.
3. **Custom transaction**, what, why, **when**, how, syntax and practical examples
4. ActiveRecord Queries
    1. **Practice different kinds of queries** with different kinds of conditions in AR context, especially "GROUP" and limit etc(you can practice on paper, with help of your buddy).
       Make sure to practice all AR topics/methods etc covered in previous days as well

### More:

1. STI associations, Self Associations
2. eager_loading/pre_loading, *lazy loading*
3. Rails: ActiveRecord Relational Objects | by Chris Kakos | The Startup

[4] Skim through and get a feel of all topics, must practice ones mentioned in this doc

# Day 8 (Mailers and sessions)

## Learning Goals:

- Complete practical and conceptual walk through of “**mailing”** in RoR. How everything works around mailing. Get practical hands on practice on different aspects of emailing in RoR.
- Basics of sessions cookies etc + background jobs in RoR.
- Some more AR concepts and their practical use, learn Querying in “rails way”, in depth.

## Content:

### Basics:

1. Revise all previous days topics(go through)
2. Action Mailer Basics — Ruby on Rails Guides
3. Ruby on Rails #28 Action Mailer: Gmail SMTP - send emails in production for free (Just try prod config on local for practice, don’t use letter_opener)
4. Mailer views basics(**what’s different then normal views, does asset pipeline work in mailer views?**), mailer config(SMTP), **setting subject/to/from** etc.
5. Practice everything(use **html** views), make sure you practice sending some mails with proper SMTP configuration using your company gmail account.
6. **Sessions/cookies and Devise**
    1. #209 Introducing Devise
    2. #210 Customizing Devise, customizing devise views (Just a know how)
    3. Session vs cookies, how rails manage sessions and cookies. Session stores configs in rails.
7. **ActiveRecord (Continued)**

**Queries** (with each method you should know use cases of each method, and practice and see what kind of queries each method produces, you can see produced SQL in the console when you run any of these methods)

1. includes, **left_outer_join**, where.not, group on DB level, aggregate functions(sum, avg, min, max), find_or_create, order/limit etc
2. **Practice different kinds of advanced queries** with all kinds of joins and conditions on joined models in AR. Play around on the console and get your intuition built in AR querying.
3. Make sure to practice all AR topics/methods etc covered in previous days as well

### More:

1. Devise: Remember me, login cookie time (how it works).
2. **devise extensions**

## Day 9 (Misc.)

### Learning Goals:

- Ways rails handle css/js/image assets/files. Rails asset pipeline, its usage, its difference with webpacker.
- Room to revise and practice remaining things from previous days, use this time well.

### Content:

1. Revise all previous days topics(go through thoroughly, you have enough time in this day, practice any missing things from previous days as well)
2. **Background jobs** and background working, what, why, how (**Must practice with async jobs**).
    1. Should know how the actual background process works, how the queue is filled and consumed etc.
3. **Rails 6 intro**, learn what is different(so you know what to expect if you are assigned a rails 6 project) -- Practice not necessary for this, but basic understanding is required.
    1. Understanding Webpacker in Rails 6 | Road to Rails 6 (Basic understanding/know how of webpacker and rails 6 usage of it).
    2. https://github.com/rails/rails/tree/6-0-stable/actionmailbox (Intro)
4. **Services and concerns** (in models/controllers)

## Day 10 (Final topics, intro to testing, revision)

### Learning Goals:

- Some final necessary topics to close Rails training, and prepare for test project.
- Enough room to revise, use it well, and get your external evaluation planned with training department.

### Content:

1. Revise all previous days topics(go through)
2. What are rails configurations, which files are used and how they work. Rails 5+ configuration for different environments(what are the environment files).
3. Ways to **Manage Secrets/keys** etc in rails?

4. Minitest vs Rspec(Testing suit in rails on basic level).
5. **Types of tests** and testing(theoretical understanding).
6. Revise all rails training outline and make sure you are ready to start your test project, also get sign off from your buddy for evaluation.

## Advanced, bonus section:

This section is good to know, not required. Just for the curious minds. Which means we will not give extra time for this section. If you finish rails early or you have extra free time from your schedule, you should cover these topics.

1. There are some topics that we just haven't had a chance to get into yet but will prove useful for you to know. In this section we'll cover advanced routing, layouts, and a brief introduction to metaprogramming.
2. What is rack(not rake), how it works, what are rack middlewares. What is lifecycle of http request in rails
3. Rails 6.0+ and webpacker and practice JS flows in it, using babel/es6 etc in it