

Ruby on Rails

→ Rails Migration and Controller

Rails Migration - It allow you to use Ruby to define changes to your database Schema, making it possible to use a version control system to keep things synchronized with actual code.

Uses of Migration -

- team of developers - if one person make changes the schema, the other developers just need to update.
- Production servers - Run 'rake migrate' when you roll out a new release to bring the database up to date as well.
- Multiple Machines - if you develop on both desktop and a laptop or in multiple location, Migration keeps ~~them~~ them synchronized.

*→ Controller: Rail Controller is the logical center of any application. It coordinates the interaction b/w user, view and model.

Code to generate Controller: rails generate Controller book

- It manage cache, It boosts performance.
- manage sessions.
- responsible for routing external requests.

*> Dynamic Scaffolding

i) dynamic scaffolding is used to give the model name that is being used with the function.

ii) dynamic automatically generate the entire content and user interface at runtime

iii) It doesn't req the db to be integrated. it creates at runtime

iv) allow generation of new, edit and delete methods for the use in app

Static scaffolding

i) Static scaffolding requires manual entry in the command.

ii) Static scaffolding requires the insertion of command generate delta with fields.

iii) It req the db to integrate

iv) Static doesn't allow it.

*> Action Controller: web requests are handled by action controller and Action view. Action controller is concerned with communicating with database and performing CRUD operations.

Action view: it is responsible for compiling the response.
- Action view templates are written using embedded ruby with html tags.

* Rails Callback: callbacks are method that get called at certain moments of an object's life cycle. that means that it can be called when an object is created, saved, deleted, updated.

A callback is more short lived. And you pass it into a function to be called once.

Rails observers: observer is similar to callback but it is used when method is not directly related to object lifecycle.

→ observer lives longer

→ there can be many observers ~~and~~ at same time and they can have diff ~~to~~ lifetime.

* Object Relationship ~~model~~ Mapping (ORM)

→ it is a framework written in object-oriented language.

→ it is wrapped around relational database.

→ object class is matched with db tables and object instance is matched with table row.

advantage

i) Reduce code and increase efficiency

ii) data base independence.

iii) free management of db relationship

disadvantage -

i) low performance in ~~bulk~~ ^{bulk} transaction

classmate

PAGE

--	--	--

Features of ruby on rails.

- i) Ajax library
- ii) Render from anywhere.
- iii) garbage collector.
- iv) oops supported
- v) Customized URL can be generated.
- vi) Rails API
- vii) low memory consumption.