


Cognos 11 reporting tutorial

I'm not robot  reCAPTCHA

[Continue](#)



project is based on the use of CDN files and the use of npm packages to build the project. JSX is a continuation of JavaScript. This is a template script where you will have the right to use Html and javascript together. The components are similar to pure javascript features that help make the code easy by dividing logic into reusable independent code. The state is a javascript object similar to props that has data to be used with reactjs visualization. These states are a private object and are used in components within the class. Props are properties that will be used inside the component. The component's lifecycle is divided into initialization, installation, upgrade, and unMounting stages. In reactjs HTML inputs such as the input has its own state and needs to be updated when the user interacts using the setState method. Dealing with events in reactjs is the same as you would do in javascript. You can use all the event handlers that are used in javascript. Teh Teh The method is used to update a state where a user interacts with any html element. ReactJS allows you to work with external css as well as inline CSS using javascript. Page 3Ruby is a pure object-oriented programming language. It was created by Yukihiro Matsumoto in Japan in 1995. It's a dynamic open source language that has a large community behind it. It was designed for simplicity and performance. Ruby encourages writing code for people first, and computers second. What are rails? Rails is the basis of Ruby's development to create web applications. It was created as the basis of the Basecamp application and then released as an open source software for everyone. It was created by David Heinemeier Hasson in a popularly known as DHH. It is one of the most influential and popular tools for building web applications. It offers a variety of standard features and built-in features, making it a suitable option for your MVP prototyping and development. It is used by some of your favorite sites such as Airbnb, Github, Shopify, etc. In this tutorial for beginners, you will learn: Why rails? Here, the pros/benefits of using Rails: Rails is packaged as Ruby's Pearl, and you can use it to create a variety of apps. This allows you to create regular web applications, e-commerce applications, content management, and more. Rails is a complete structure that incorporates everything you need to create a database-driven web application using the Model-View-Controller template. This means that all layers are built to run smoothly along with less code. This requires fewer lines of code than other frameworks. Installing Ruby on the Windows installation process will depend on your operating system. You'll go through the Ruby installation on Windows, Mac and Linux. The easiest way to install Ruby on a Windows computer is through Ruby Installer, which can be downloaded on . All you have to do is run the loaded installer. Step 1) Double tap the downloaded installer 'rubyinstaller-2.4.1-2-x64.exe'. Step 2) Select the radio button I accept the license and click next. This should lead you to the following picture below. Step 3) Check out the first two checkboxes to make running Ruby scripts easier. Click Install to start installing. You should see the picture below when the installation is complete. Step 4) Don't control the checkbox option that MSYS2 installs. Click Finish to complete the installation, and you'll see the command fast window shown in the picture below. Step 5) This step installs MSYS2, a construction platform that has a package manager for easy package installation on Windows. Click Enter to install all the components They all need to have a smooth Ruby on the rails development environment to work with windows. The RailsYou installation should have Ruby installed from the previous section and now you are going to install the rails. You Are You install Rails with a package from Railsinstaller, but the problem is that you don't get the latest version of the packages you need. If you already have the newest Ruby and a basic set of necessary RubyGems and installed extensions. All you have to do now is run the next command on command tip to get the rails on your system: gem set the rails. You'll also need to install Node.js if you don't already have one, because some of the libraries that Rails depend on require that Javascript run time work properly. You can get a node on the . This is a more common and preferred approach to development on Windows. The Rails community uses the Windows Linux subsystem, which provides a GNU/Linux environment with command-and-line tools, utilities, and shared apps directly on Windows. Ruby's MacYour Mac installation has already been pre-installed on Ruby. However, the pre-installed version may be old, so you'll need to install a new/last version. The easiest way to do this is with the help of a package manager such as Homebrew. You may need to install Homebrew first by leaping the command below in the Terminal. /usr/bin/ruby-es(curl-fsSL. This will display a warning and ask you to enter a password. Enter the Mac password you won't see the characters when you type in). Just click 'Enter' when you've finished entering your password. Then run this simple Homebrew command to install Ruby on your Mac. brew install rubyAnd also run this command: echo export PATH/usr/local/bin:/usr/local/sbin:\$PATH s/ bash\_profileTo install this Ruby installation as the default ruby to run in your system, not for pre-installed Ruby. To confirm a successful installation, you can run the next one in the ruby terminal --versionthis will print the Ruby version number you installed. The output will look like a ruby 2.6.0p0 (2018-12-25 revision 66547) (x86\_64-darwin18) Installing Ruby on Ubuntu (Linux) The easiest way to get Ruby installed on your Ubuntu system is through apt package manager. To install the last Ruby from the Ubuntu repositories, you need to stick out the following commands in the Terminal. sudo apt update - This will be the default update Ubuntu repositoriessudo apt install ruby-full - It will download and install the latest RubyTo confirm the installation was successful, you can run the next ruby-version, it will print the ruby version you installed. Installing rails on Ubuntu (Linux) you have to follow the steps below to successfully install the rails on your Linux computer. Step 1) Update your computer's gem manager by running a 'gem update-system' in the Terminal or command request. Step 2) Run 'gem set rails' to install the latest version of the rails on your Step 3) You have to install a gem bundler for easy management of dependence on Rails applications. You run a 'gem set bundler' to get it. Two RailsRails principles follow basic software principles and encourage you to use these principles too. The two most common are: Don't repeat yourself (DRY) - it makes you write concise, consistent, and support code. Configuration Convention - Rails is pre-configured to use reasonable defaults that correspond to the most common use. This makes the development of the application quick, and you also have less code to service. Rails - Project file structures with rails now installed on your system lets you create a Rails app! You'll learn how to create the Todo list app in this tutorial. Start the next rails todo\_app command in the Terminal to create the app. This team creates a catalog called todo\_app in the current catalog with the basic folder structure of the Rails web application, as shown in the picture below: You will go through the main directories in this section. App - This catalog of groups using different sub-direction for the user interface/layout (views and assistants), controller (controller files) and models (business/logic application). App/Controllers - This directory stores controller files used by Rails to process requests from the customer. app/assets - It contains static files, which is a necessity for the front end of the application grouped into folders depending on their type - JavaScript files, images and style tables. app/helpers - This subdirectional contains assistant features that make your application model, controller presentation and logic focused, small and uncluttered. App/model - it contains files that simulates the app's database. Model classes make working with the database very simple. app/views - This is the retention of the template/model of files that your app user interacts with. Patterns are a combination of HTML and data from the database. Ben - It contains The Rails scripts that launches your app. It can also include other scripts that you use to customize and update your app. Config - This contains configuration files - database.yml, environment.rb, routes.rb, etc. that your app should work. DB - This catalog contains files/scripts that are used to manage the application database. lib - This catalog contains an advanced module for your app. Magazine - This contains log files - server.log, development.log, test.log, and production.log, etc., which are used to debug or monitor your application. Public - This catalog contains static files and collected resources such as HTML files, Javascript files, images, and style tables. Test - This catalog contains test files that you write to check the functionality of the app. tmp - This directory contains temporary files such as cache and pid files. Supplier - This catalog contains third-party libraries. Gemfile - This file determines what the basic requirements of gemstones to run a web application. You can group gems into design, testing or production, and Rails will know when to turn on each gem. Gemfile.lock - Unlike Gemfile, which clearly lists the gems you want in your app, the app, Additionally contains other gems that you list in Gemfile depends on what is then automatically installed to meet dependencies. Readme.md - You use this file to share important details about your app, such as what the app does, how to go about installing and launching the app. Rakefile - This file contains a variety of rake task definitions that help automate the day-to-day administration tasks of your application. config.ru - This is a rack configuration file that provides an interface for the webserver to start your app. Change the catalog to the todo\_app Rails catalog, created and run a rail server to run the app. Think localhost:3000 in the address page of your web browser, you should see the picture below if all went well. This is the default app homepage, and you'll change that in a later section of the tutorial. You can stop the server by clicking 'Ctrl-C'. Rails - Create teams that generate Rails teams, use templates to create many more useful things in your app. You can use these generators to save a lot of time. This helps by writing boiler code, the code you need to run a web application. You can run the rails generate by itself by command hint or terminal to see the list of available generators, as shown below: You can also run the rails to generate commands to see a description of what the team is doing. It offers convenient options that can work with the team and an example of use. The image below shows the exit of the rails running to generate the controller: You'll use the rails to generate scaffold commands to automatically create a model, view and controller for the to-do list app that you build. Run' rails generate scaffold todo\_list title: description of the line: text in your terminal (check you are still in the todo\_app catalog). This will create a complete CRUD (Create, read, update and delete) web interface for the TodoLists table. Another useful team to know is the rails to destroy, it changes all the rails generate ... Makes. Rails - routingThe Rails routing system, rails router, handles all incoming requests in the web application. It does this by studying the URL of incoming queries, and then displays each request for the action of the controller responsible for processing it using the special syntax listed in the route file (config/routes.rb). The route file helps you control every aspect of your web application's URL. Rails by default use a RESTful design based on the architectural style of REST, which provides a mapping between the verbs http and queries (URL addresses) for controller actions. The route file was created when you launched the new rails in the previous section of this tutorial. Continuing the Todo app you're building In the command line make sure you're still at the root of the app (catalogue todo\_app). Then start server again with a rail server. Enter in your browser and click Enter. You should get back the web page as shown in the picture below: It's Todo lists the view that the scaffold team generated and it's controlled by the TodoListsController Action Index. Go ahead and add a to-do list by clicking on the new Todo list on the page you should get the page as shown below: Notice URL changed to . This page is to create a new to-do list, and it's controlled by a new TodoListsController method/action. Enter the to-do list name and description in the form and click the Create Todo list button, the URL should change to shown in the picture below: This is the todo list show page, and it is controlled by the TodoListsController/action show. If you go back to you should now see the figure below with a new todo list added: Rails has been able to match different queries (URLs) to relevant TodoListsController actions using route definition in config/routes.rb. If you look at this file, you'll see one line of resources: todo\_lists, is Rails' default way of writing the rest of the routes. This single line creates seven routes, all displaying the TodoLists controller. By convention, each controller also displays a specific CRUD operation (Creating, Reading, Update, and Delete) in the database. You can run rake routes in the command line to see the different routes available in the app. The image below shows the launch of the rail routes in the command line/terminal. Rails - The ViewsThe View layer is one of the components of the MVC paradigm and is responsible for generating an HTML response for each request into your app. Rails by default use ERB (Embedded Ruby), which is a powerful patterning system for Ruby. ERB simplifies and supports pattern writing by combining simple text with Ruby code for variable replacement and thread management. The ERB pattern has an extension of .html, .erb or .erb. You'll basically use a combination of two tag markers only, each of which triggers a built-in code to process and process in a certain way. A tag with an equal sign, Lt;% of the gt;, indicates that the embedded code is an expression and that the result of the code should be replaced by a renderer when it displays the template. Another tag without the zlt;% tag indicates to the render that the result of the code should not be replaced/printed when it displays the template. Each controller in the Rails app has an appropriate sub-direct in the app/views, and each action/method in the controller has an appropriate .html and .erb file in that catalog. Take a look at the todo app you're building You'll find a sub-directed todo\_lists inside this sub-directed .html.erb files with names, Actions/methods in Controller. Rails - ActiveRecord, active recording template, and ormActiveRecord is the implementation of the Ruby Active Record template, which is a simple template where a class represents a table and a class instance is a line in that class. ActiveRecord is popularly referred to as ORM (Object Relational Mapping), a method that lets you manage your database using the language you're most comfortable with. So it is an agnostic database, so you can easily switch between databases (e.g. SQLite, MySQL, PostgreSQL, SL Server, Oracle, etc.). This set is larger for your application requirement with the same code/logic. So if you want to get an array containing a list of all the to-do lists in the app, so instead of writing code to initiate a connection to the database, then make a kind of SQL SELECT request, and convert those results into an array. To do this, you just need to enter 'TodoList.all' and Active Record gives you an array filled with TodoList objects with which you can play as you like. All you have to do is customize the correct configuration in the configuration/database.yml, and ActiveRecord will work all the differences between the different database systems. So when you switch from one to the other, you don't have to think about it. You focus on writing code for your app, and Active Record will be thinking about the low-level details of connecting you to your database. Active Record uses naming conventions to create a display between models and database tables. Rails pluralize model class names to find the appropriate database table. For the TodoList class, for example, ActiveRecord will create a database table called TodoLists. Rails - Migration MigrationsRails is simply a scenario that is used to edit the application database. It is used to set up or modify the database and avoids manually writing SQL code for this purpose. It uses Ruby to determine changes to the database diagram and allows you to use the version control to synchronize the database. Rail migration uses a specific Ruby Domain (DSL) language. This acts as an abstraction and allows you to use or modify the database engine depending on your requirements. They can be passed on to anyone who is working on the app, and can also be rolled back to undo any changes to the database. This is a high security mechanism because you don't have to worry about causing permanent damage to your database. Rails - ActiveRecord AssociationsA connection between the two ActiveRecord models is known as the association. The association greatly simplifies operations on different records in the code. It can be divided into four categories: - One to one: - This means that the record contains exactly one instance of the other model. A good example is the user profile. The user has one profile. He uses has\_one word. One to many: - This is the most common association, and it indicates that one model has zero or more instances of another model. Your use has\_many many to designate this association. To many: - This association is a little more complicated, and ActiveRecord provides two ways to deal with it. Using has\_and\_belongs\_to\_many has\_many, which gives you access to a relationship that is defined in a separate table. Polymorphic One for many: - This more advanced association is available to you in Rails. It can be used to identify a model that may belong to different models in the same association. Rails - ActiveRecord ValidationsValidation helps to ensure that you have the right data because working with the wrong data is a terrible thing and can cost you your money and business. The check also provides an extra layer of security for your app from malicious users from gaining access to information in your database. Rails offers you a good API of verification assistants at ActiveRecord to keep your database clean, secure and error-free. ActiveRecord checks work on model objects before being stored in the database, making them more reliable as well as best practice to create an application. ActiveRecord's following methods trigger checks when you use or call to create, create, create!, save, save!, update and update!. Those with a bang (create!, save! and upgrade!) raise an exception if the entry is invalid while thothentTTT.The most common ActiveRecord verification assistants at your disposal are: Confirmation:- This verification assistant is useful for checking two fields have the same record. For example, confirming a password and password, it is used in conjunction with a presence verification assistant. Presence:- This checks that the field is not empty. Uniqueness: Provides a unique value for the field, such as the length of the username: - To limit the length of the field symbol, you can also create a custom check using the verification method and giving it the name of the verification method. You can check the object of the model error to find out why the check. We hope you have ideas to make your application more limited and more secure to allow only secure data in your database. Rails - ActionControllerThe Rails controller is the center of your web application. It facilitates and coordinates the communication between the user, models and views. Controller classes are inherited from ApplicationController, which contains code that can be run in all other controllers and inherited from the ActionController class. The controller provides the following for your application: It routes external queries to internal activities that it manages to cache, giving performance to your application, it manages assistant methods that enhance the capabilities of view patterns. It also manages custom sessions, giving them a smooth experience using your app. Rails - ConfigurationsYou can customize various components such as start-ups, assets, generators, medium programs, etc. using Rails app starters and configuration files in the configuration catalog. Files as like config/environments/development.rb and config/environments/test.rb, etc. Rails - DebuggingWho you build your app, the time will come you need/have to debug your code. Rails make it easy with the help of pearl byebug. You can start the debugging session by putting the keyword 'byebug' anywhere in the application code. This will temporarily stop the execution at this point. The byebug gem gives you several commands to use. The most useful are: the following: a command that allows you to move to the next line of code, skipping all the methods caused by the current line step. There are other debugging gems such as pry, and they all provide similar features but slightly different syntax. Gem debugging should not be used in manufacturing as it poses risks to your application and a bad experience for app users. There are log files that you can check for errors in production and handle them. In addition, you should follow the TOD (test-driven development) approach when developing your application to make sure everything is working well before you can go into production. Summary: Ruby is a pure object-oriented programming language Ruby has an elegant syntax that is easy to read and write. Rails is the basis of Ruby's development, and the installation process will depend on your operating system to create web applications. Rails is packaged like Ruby's Pearl and you can use it to create different apps. You'll create the Todo list app in this tutorial, run followincomm'n' netoda'p'r rails in YouR Terminal to create the app. The Rails team uses templates to create a lot more useful things in the app. Rails routing system, rail router will help you process all incoming requests in your web application. The View layer is one of the components of the MVC paradigm and is responsible for generating an HTML response for each request into your app. ActiveRecord is the implementation of the Ruby Active Record template. Rail migration is simply a scenario that is used to edit the application database. The link between the two ActiveRecord models is known as the association. Checking helps ensure that you have the right data because working with the wrong data is a terrible thing and can cost you your money and business. The Rails controller facilitates and coordinates the communication between the user, the models and the views. Rail will help you customize different components such as initializers, assets, generators, medium programs, etc etc etc.

[addition\\_and\\_subtraction\\_of\\_polynomials\\_worksheet\\_1\\_answers.pdf](#)

[tone\\_of\\_poem\\_mother\\_to\\_son.pdf](#)

[mortal\\_kombat\\_project\\_2020.pdf](#)

[blank\\_map\\_of\\_sub\\_saharan\\_africa](#)

[all\\_dyes\\_in\\_minercraft](#)

[krups\\_proraroma\\_coffee\\_maker\\_manual](#)

[sections\\_of\\_a\\_newspaper.pdf](#)

[trouser\\_cutting\\_pattern.pdf](#)

[x\\_vpn\\_apk\\_for\\_android\\_free\\_download](#)

[human\\_embryology\\_pdf\\_download](#)

[the\\_break\\_even\\_point\\_calculation\\_is\\_affected\\_by](#)

[diccionario\\_latin\\_espaol\\_pdf\\_gratis](#)

[dragon\\_mania\\_mod\\_apk\\_offline\\_unlimited\\_gems](#)

[pengertian\\_filisafat\\_pdf\\_jurnal](#)

[programming\\_principles\\_and\\_practice\\_using\\_c](#)

[export\\_certain\\_pages\\_of\\_pdf](#)

[scph10000\\_bin\\_and\\_rom1\\_bin\\_ps2emu](#)

[download\\_song\\_dehizez\\_pe\\_mere\\_dil\\_ki](#)

[normal\\_5f8724de973b1.pdf](#)

[normal\\_5f87a450dce2b.pdf](#)

[normal\\_5f87ad504ded4.pdf](#)

[normal\\_5f87931727fe3.pdf](#)