Ruby (https://cloud.google.com/ruby/) Guides

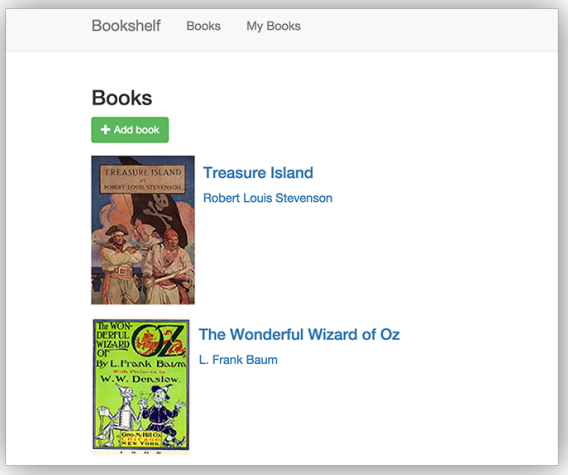
Ruby Bookshelf app

The Bookshelf app is a sample web app written in Ruby that shows how to use various Google Cloud products, including:

- App Engine flexible environment
- Cloud SQL
- Datastore
- Cloud Storage
- Pub/Sub

This tutorial explores the Bookshelf sample app and discusses how each feature of the app is implemented using technologies and services that Google Cloud provides.

The Bookshelf sample app is based on the Ruby on Rails (http://rubyonrails.org) web application framework, but the concepts and technologies explored are applicable regardless of which framework you use. You can implement this app in another Rack (http://rack.github.io) -based Ruby web framework, for example, Sinatra (http://www.sinatrarb.com).



The Bookshelf sample app stores a collection of book titles. Anyone with access to the app can modify the list. The sample app lets users:

- View the list of books.
- Add books to the list.
- Remove books from the list.
- Edit book details.
- Upload cover images for books.
- Sign in with a Google Account and view the books that they added to the list.

Objectives

- Clone or download the sample app.
- Build the app and run it on your local machine.
- Deploy the app to App Engine.
- Walk through the sample code.
- Learn how the app stores structured data.
- Learn how the app stores binary data in Cloud Storage.
- Learn how the app authenticates users.
- Learn how the app creates event logs that are visible in the Google Cloud Console.

- Learn how the app uses Pub/Sub to send tasks to a background worker.

Costs

This tutorial uses billable components of Google Cloud, including Compute Engine.

This tutorial has several steps, and each step is documented on its own page. The final page of the tutorial includes instructions for cleaning up resources so you won't continue to be billed for Google Cloud services. If you decide not to complete all the steps of the tutorial, see the cleanup instructions (https://cloud.google.com/ruby/getting-started/using-pub-sub#cleanup) on the final page.

Before you begin

1. To set up your Google Cloud project, use the Cloud Console:
    a. Create a Google Cloud project, create an App Engine app, and then enable billing in that project.

    GO TO APP ENGINE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECTOR/APPENG

    When prompted, select the region (https://cloud.google.com/appengine/docs/locations) where you want your App Engine app located, and then enable billing. After your Google Cloud project is created, the **Dashboard** opens.

    b. Enable the Cloud Datastore, Cloud Pub/Sub, Cloud Storage JSON, Stackdriver Logging, and Google+ APIs.

    ENABLE THE APIS (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=DAT
2. Download, install, and initialize the Cloud SDK.

    DOWNLOAD THE CLOUD SDK (HTTPS://CLOUD.GOOGLE.COM/SDK/DOCS/)
3. Acquire local credentials for authenticating with Google Cloud services.

```
gcloud auth application-default login
```

4. Verify that your default project is correct.

```
gcloud config list
```

If the project ID listed in the output isn't the project that you intended to use for this tutorial, set the project.

```
gcloud config set project [YOUR_PROJECT_ID]
```

where `[YOUR_PROJECT_ID]` is the ID of the project that you created or chose to use for this tutorial.

Note: You can create Cloud SDK configurations to set configuration properties, such as a Google Cloud project ID. You can switch between those configurations each time that you use the `gcloud` tool. For more information, see Managing Cloud SDK configurations (https://cloud.google.com/sdk/docs/managing-configurations).

5. Clone the sample repository.

```
git clone https://github.com/GoogleCloudPlatform/getting-started-ruby.git
```

Alternatively, you can download the sample (https://github.com/GoogleCloudPlatform/getting-started-ruby/archive/master.zip) as a zip file and extract it.

This tutorial assumes that you are familiar with Ruby on Rails (http://rubyonrails.org/) and that you have Ruby (https://www.ruby-lang.org/en/documentation/installation/) 2.0.0 or newer installed. To run the `gem` command, you need RubyGems (http://guides.rubygems.org/rubygems-basics/), which is included in Ruby. You also need to have Bundler (https://bundler.io) installed. As with most Ruby development, installing Ruby gems in this tutorial requires compiling C code and linking against system libraries. Ensure that you have a suitable development environment configured for your particular operating system. This includes a C compiler, linker, and header files for those system libraries.

Tutorial structure

The Bookshelf tutorial has several parts that demonstrate how the sample app uses various Google Cloud services.

The structured data part of the tutorial demonstrates how the sample app stores book information in a SQL or NoSQL database.

The sample app's web page displays a form where the user enters the title, author, description, and publication date of a book. The app stores this information in a database so it can be retrieved later for viewing or editing. For this step of the tutorial, you have your choice of three databases: Cloud SQL, Cloud Datastore, or PostgreSQL. After you complete this step with one of the databases, you can move on to the next step.

The Cloud Storage part of this tutorial demonstrates how the sample app stores binary data in Cloud Storage. On the app's web page, the user can specify a cover image for each book. The app stores the cover images in a Cloud Storage bucket.

The authorization part of this tutorial demonstrates how the sample app provides a sign-in flow for the user. When a user is signed in, any books that the user enters are associated with the user. Signed-in users see their own books and books that are created by anonymous users. Other users can't see books that are created by a signed-in user.

The logging part of this tutorial demonstrates how the sample app writes logs that are visible in the Google Cloud Console. Logs of this type can provide diagnostic information during app development.

The Pub/Sub part of this tutorial demonstrates how the app uses Pub/Sub to send tasks to a background worker. The worker gathers information from the Google Books API and updates the book information in the database.

< PREV

NEXT > (HTTPS://CLOUD.GOOGLE.COM/RUBY/GETTING-STARTED/USING-STRUCTURED-DATA)

Last updated December 4, 2019.