Implement a rails application that satisfies the requirement in the story below. The rails application must use the following web service to retrieve the data for the application.

http://www.webservicex.net/country.asmx?WSDL

The currency data does not change frequently. Care should be given to minimize web service requests. It is not necessary to create a ruby soap client from scratch. Existing ruby soap clients are available, and their use is strongly encouraged. Three such ruby soap clients are:

- Soap4R (http://dev.ctor.org/soap4r)
- Savon (http://github.com/rubiii/savon)
- HandSoap (http://github.com/unwire/handsoap)

Additionally, feel free to use any other gems that will facilitate building the application faster. Test evaluation will be focused upon functionality and less on user interface design (i.e. css).

Please provide a set of instructions for both verifying the functionality of the application and running the application. This shouldn’t take any longer than 6 consecutive hours to complete the entire test. Please indicate in your response the length of time required to complete the test.

Background

Mr. Smart is a private money collector. He wakes up one morning with a brilliant idea: create the only complete private currency collection in the world. To do this, he must first be able to identify all of the currencies that are available in the world and then to identify where in the world the currencies are located. His plan is to pick a single currency and then systematically proceed to each country where the denomination is available to acquire bills and coins in various denominations. His problem: he doesn't know what all of the available currencies are and he doesn't know where to get them.

Smart breathes a sigh of relief. *sigh*

He is not worried. He knows of an engineer (you) that has built a web application that will facilitate his conquest to complete the collection. The following user stories detail his use of the application.

Identifying available currencies

Mr. Smart is not aware of the complete set of available currencies in the world. To identify the list of currencies, he navigates his browser to a page in the application that presents him with a list of the available currencies.

Identifying the countries that offer a particular currency

Mr. Smart decides to collect a particular currency. He indicates to the application the currency he is interested in and is presented with a list of countries offering that currency.

Tracking visited countries

Mr. Smart selects a currency from the application. He is presented with a list of countries. He then selects the countries that he has visited for that currency.

Identifying the remaining countries from the currency list

Mr. Smart navigates his browser to a page in the application that presents him with a list of available currencies. For each currency, the page also displays the number of countries that are available for the currency, and the number of remaining countries to be visited.

Next Steps:

Your application has been released. Mr. Smart is using it. He is almost done acquiring the currencies necessary to complete his collection. Being a personal friend of yours, he makes the following requests to enhance your application. You are going to make the following modifications and then re-release your application.

Filtering currencies from the list

Mr. Smart navigates his browser to a page in the application that presents him with a list of all available currencies in the world. He is able to toggle the view, limiting the display of currencies to those that still have countries to visit.

Tracking trips to specific countries
Mr. Smart wants to enter his trip itinerary instead of selecting the currencies and then the countries he visited. Whenever he visits a country, he collects all of the currencies associated with that country.

In the application, he navigates to a travel itinerary page in the application and enters a brief trip description (i.e. description, date). He then enters in the various countries he has visited on his trip. Each currency associated with a particular country is marked as being collected when the trip itinerary is completed.

Listing trips made
Mr. Smart wants to be more efficient in his travels. He navigates to a list of trips he has made. The page displays the number of countries he visited and the dates the trips were made.

And Then:
Mr. Wicked already has a private currency collection. Upon learning of Mr. Smart's quest to build the same thing, he be-friends Smart into letting him track his progress in your application. Wicked begins unselecting the countries Smart has visited. As a result, another feature request comes your way.

Allow writable access to the country selections
Mr. Smart identifies himself with the application. Upon successfully identifying himself with the application, he is able to navigate into a particular currency's list of countries and indicate he has visited those countries.

Allow read only access to the country selections
Mr. Wicked navigates into a particular currency's list of countries. He is not required to identify himself with the application. He is able to view the countries that have been visited, but is not able to change anything.