# Programming Exercise UI Devs

Please develop a web-based application that displays a monthly calendar based on the requirements below. It is NOT important that you actually complete this exercise in the allotted time (*2 hours and a half*). It IS important that you demonstrate a fundamentally sound approach for the development of this application. **Please use client-side technologies only. If a link to a published working solution is not provided the exercise will not be evaluated.**

## Detailed Requirements:

## (You must complete the requirements in order)

1. Allow the user to specify three inputs: A **Start Date**, the **Number of days** to display and a **Country Code** literal (*the country code literal value must be defined by the user with the keyboard*).

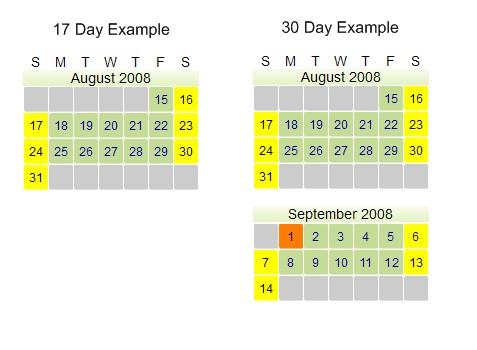
Example:

Start Date: 8/15/2008

Number of days: 17

Country Code: US

1. Render a calendar that spans as many weeks as necessary in order to cover the defined number of days. If the date range spans months, a new header needs to be created for that month. The Calendar should display its days starting on Sunday and ending on Saturday, as shown below. This should work with any number of days regardless of the years meaning that if the user specifies more than 365 days, calendars will be generated accordingly for the next year and so on.



1. The days should be color coded as follows **(they must match the picture above):**
2. Weekends are **yellow**.
3. Weekdays are **green**.
4. Holidays are **orange**.
5. Invalid days are hidden and the background should be **gray**.

Invalid days are defined as:

1. Any days in the week previous to the Start date.
2. Any days in the week after the last rendered day.
3. Any days before the first day of the month (*if the calendar spills into another month*).
4. The Holidays must be dynamically extracted from the API defined at [http://holidayapi.com **holidays are only required for the year 2008.**](http://holidayapi.com/)
5. This is a RESTful API, for which we expect web requests to be created and handled appropriately.
6. We also expect different result codes to be handled.
7. **Extra credit** will be given for adding a rollover label on Holidays.

## 

## Ground Rules:

1. The solution must be uploaded and shared to your gitHub account since this is what we will use in order to review the code. Please adhere to the following guidelines:
2. It is **MANDATORY** to create an initial commit when you sit-down to start the exercise, with whatever content you wish.
3. It is recommended that you commit and push frequently in order to protect and keep track of the progress; the frequency and contents of the commits are left to your discretion.
4. Please create a public repo so we can have access.
5. **Please provide a link with public access to the solution**

1. Please stop at 2 hours and a half (*you are not expected to finish*).
2. Use whatever resources you need, except another person’s help or old code from one of your projects.
3. Use the technology you are most comfortable with.
4. When you’re done with the exercise please provide the following information:
   1. A list of any requirements you could not implement.
   2. Bugs you’ve identified but didn’t have time to fix.
   3. Things you would do if you had more time to complete the task.