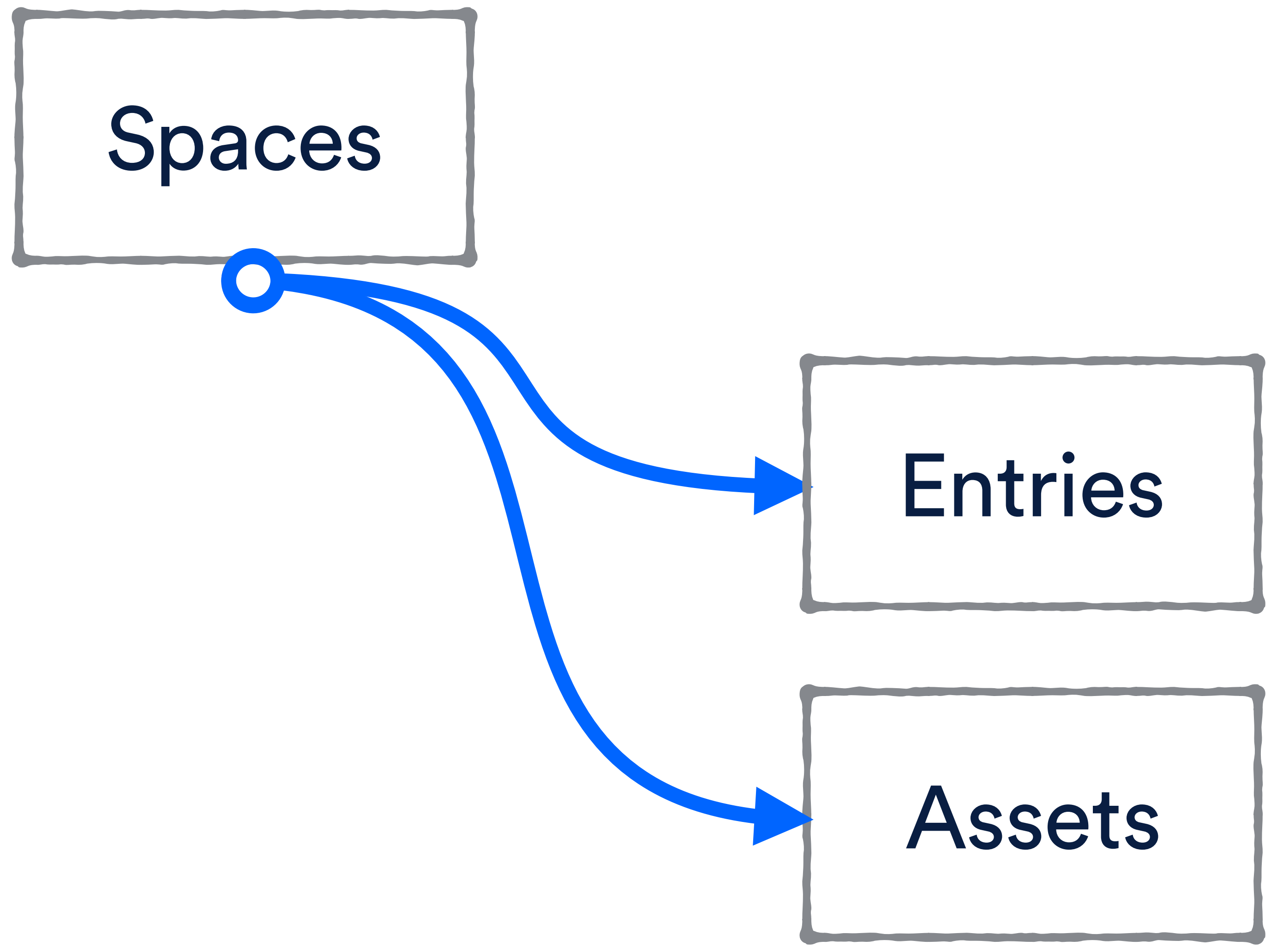
UI Coding Exercise

The objective of this test is to implement a simple layout using the UI framework of your choice which leverages a mock API to asynchronously load data for the following items:

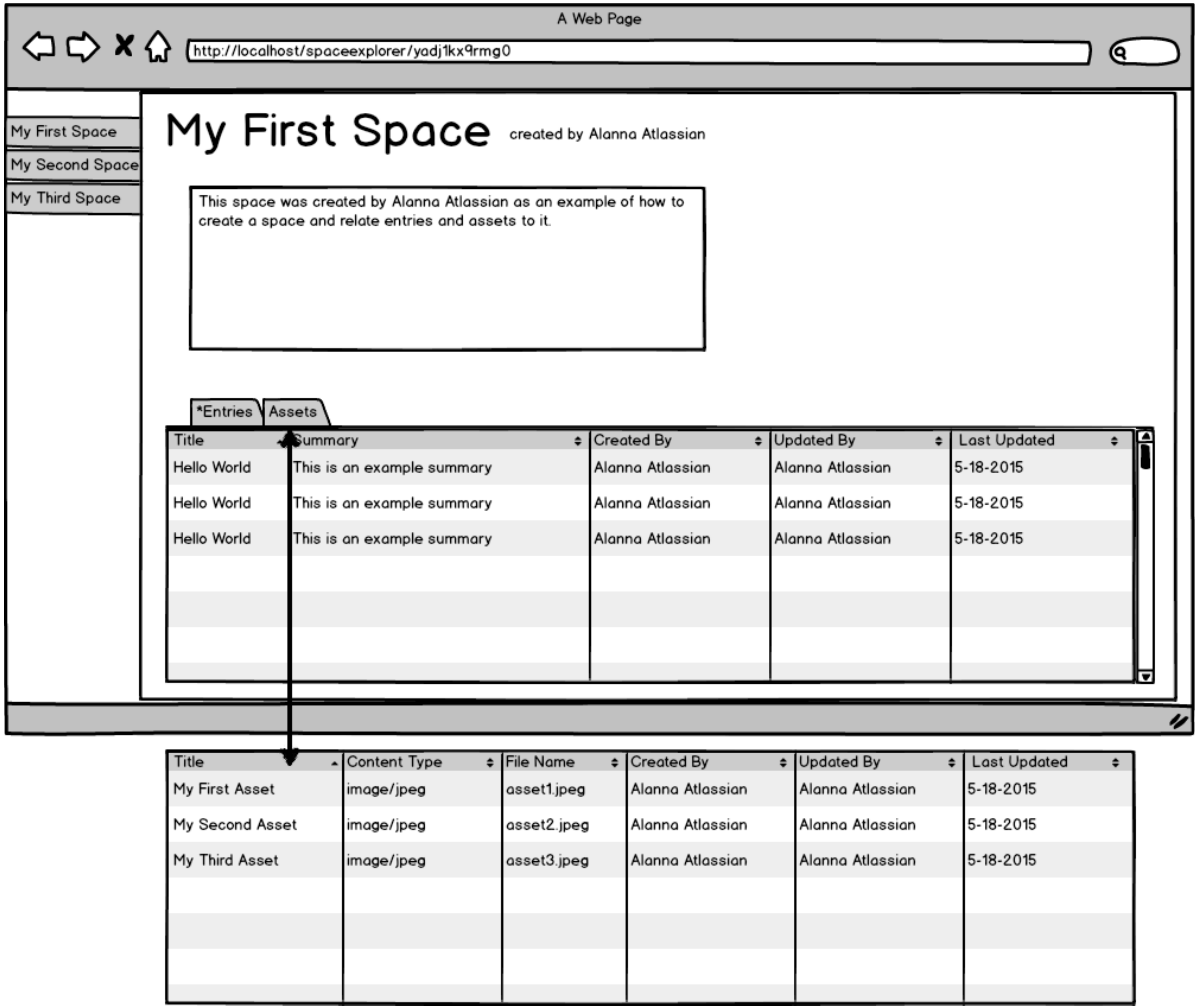
- Spaces - A general container object for Entries and Assets
- Entries - Represents a chunk of test with a Title and Summary
- Assets - A link to a file and related meta data
- Users - Records with a guid, name, and role



Two files have been included which describe the related API:

- space-api.html - an HTML rendering of the very basic RAML specification to reference while creating your application and writing unit tests.
- space-api.raml - a very basic RAML specification of an API which can be used to provide a mock API for your application

Example payloads in the provided specification center around the "My First Space" space for id: yadj1kx9rmg0.
This is the only space you are required to render fully. The Assets and Entries table should render empty or display "Not Available" for "Second" and "Third" spaces.



A Web Page

http://localhost/spaceexplorer/yadj1kx9rmg0

- My First Space
- My Second Space
- My Third Space

My First Space

created by Alanna Atlassian

This space was created by Alanna Atlassian as an example of how to create a space and relate entries and assets to it.

*Entries | Assets

Title	Summary	Created By	Updated By	Last Updated
Hello World	This is an example summary	Alanna Atlassian	Alanna Atlassian	5-18-2015
Hello World	This is an example summary	Alanna Atlassian	Alanna Atlassian	5-18-2015
Hello World	This is an example summary	Alanna Atlassian	Alanna Atlassian	5-18-2015

Title	Content Type	File Name	Created By	Updated By	Last Updated
My First Asset	image/jpeg	asset1.jpeg	Alanna Atlassian	Alanna Atlassian	5-18-2015
My Second Asset	image/jpeg	asset2.jpeg	Alanna Atlassian	Alanna Atlassian	5-18-2015
My Third Asset	image/jpeg	asset3.jpeg	Alanna Atlassian	Alanna Atlassian	5-18-2015

Key points of interaction:

Users can navigate spaces by clicking on a vertical tab on the left.

Users can navigate between the entries and assets tab for a given space.

Users can sort (on the client side) *by clicking column headers.* (For the purposes of this test, assume the API does not support server-side sort)

URL Route should contain a Space GUID.

Visiting the base URL should redirect to the first space returned by the /space end-point

Submission Requirements

- You may use any components available online.
- Scripts should be included which will build the application, host the application locally, and execute unit tests.
- A ReadMe must be provided which explains how to build and run the application.

Grading Criteria

- Coding Style is important. Your submission should be easy to grok.
- The application should elegantly handle network or API failure.
  - "What happens if a specific API errors out or is unavailable?"

Level of Commitment

Atlassian recognizes you have other commitments. Our target for this exercise is roughly a five hour investment.

How to Submit:

- Create a repository at bitbucket.org
- Ensure the Repo is Clean and has a ReadMe!
- Send an email to nreed@atlassian.com and cc: kgillenwater@atlassian.com indicating you’ve completed the assignment and include a link to your repo.
- Pat yourself on the back.

Have questions or you’ve noticed errors in the assignment:

Email: kgillenwater@atlassian.com

Thank you

We know it’s a commitment and we appreciate your efforts!