

# Callback Newsfeed

Assignment 3 Overview

# Overview

A dynamically updating newsfeed that interfaces with **Flickr**, **SoundCloud**, and **YouTube** to display content.

**[demo]**

# Client-side (“Frontend”)

views/index.html

js/search-model.js

js/search-view.js

js/post-model.js

js/newsfeed-view.js

js/main.js

js/main-view.js

js/handlebars-helpers.js

js/lib/\*.js

routes/index.js

lib/flickr.js

lib/soundcloud.js

lib/youtube.js

models/post.js

app.js

package.json

settings.js

Flickr API

SoundCloud API

YouTube API

# Server-side (“Backend”)

views/index.html

js/search-model.js

js/search-view.js

js/post-model.js

js/newsfeed-view.js

js/main.js

js/main-view.js

js/handlebars-helpers.js

js/lib/\*.js

routes/index.js

lib/flickr.js

lib/soundcloud.js

lib/youtube.js

models/post.js

app.js

package.json

settings.js

Flickr API

SoundCloud API

YouTube API

# APIs

views/index.html

js/search-model.js

js/search-view.js

js/post-model.js

js/newsfeed-view.js

js/main.js

js/main-view.js

js/handlebars-helpers.js

js/lib/\*.js

routes/index.js

lib/flickr.js

lib/soundcloud.js

lib/youtube.js

models/post.js

app.js

package.json

settings.js

Flickr API

SoundCloud API

YouTube API

# Tasks

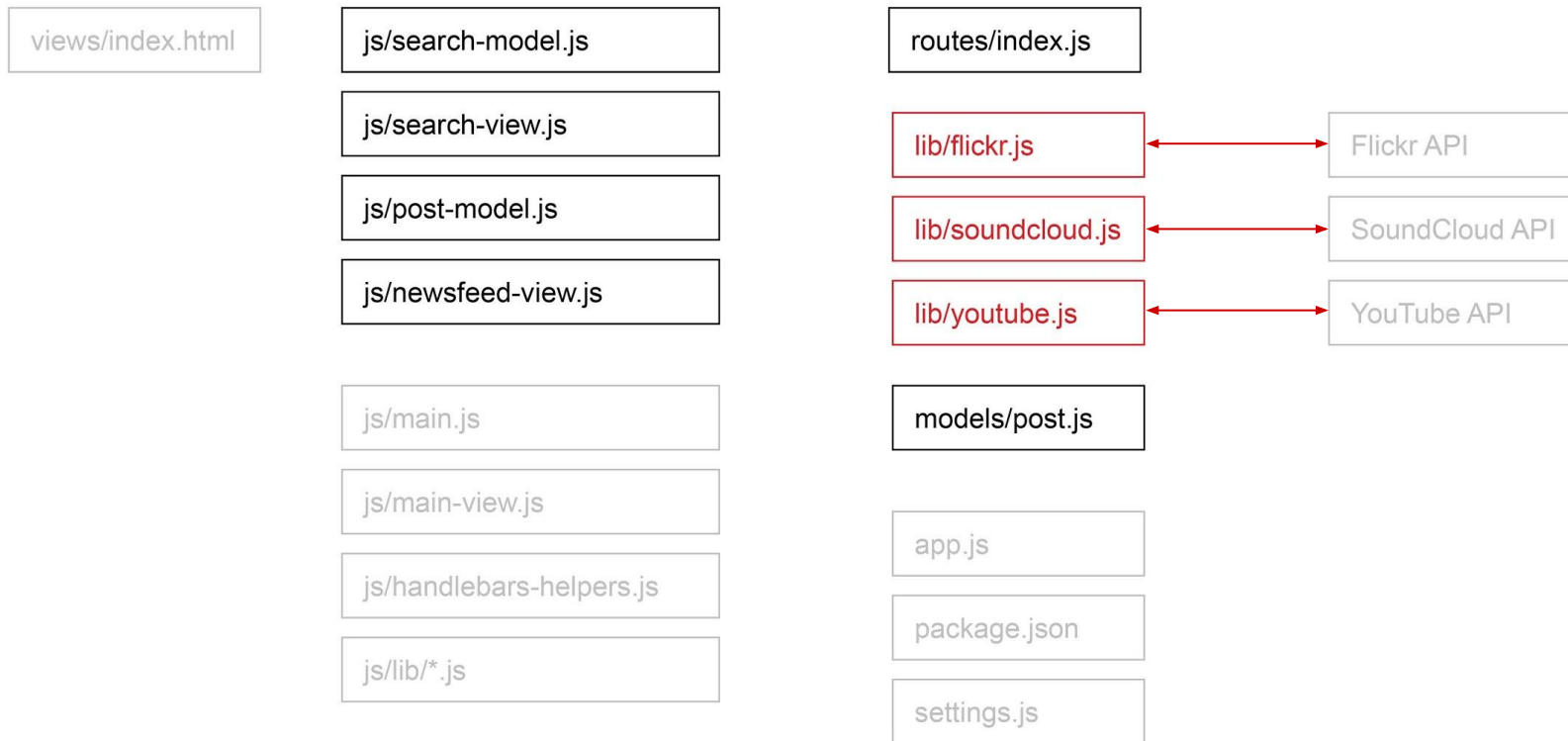
1. Interfacing with each API to retrieve results
2. Simultaneously query from all APIs
3. Displaying the search results in the view
4. Implementing a REST API to add/remove/get posts from the newsfeed.
5. Update the posts in the view

# Tasks

1. Interfacing with each API to retrieve results
2. Simultaneously query from all APIs
3. Displaying the search results in the view
4. Implementing a REST API to add/remove/get posts from the newsfeed.
5. Update the posts in the view



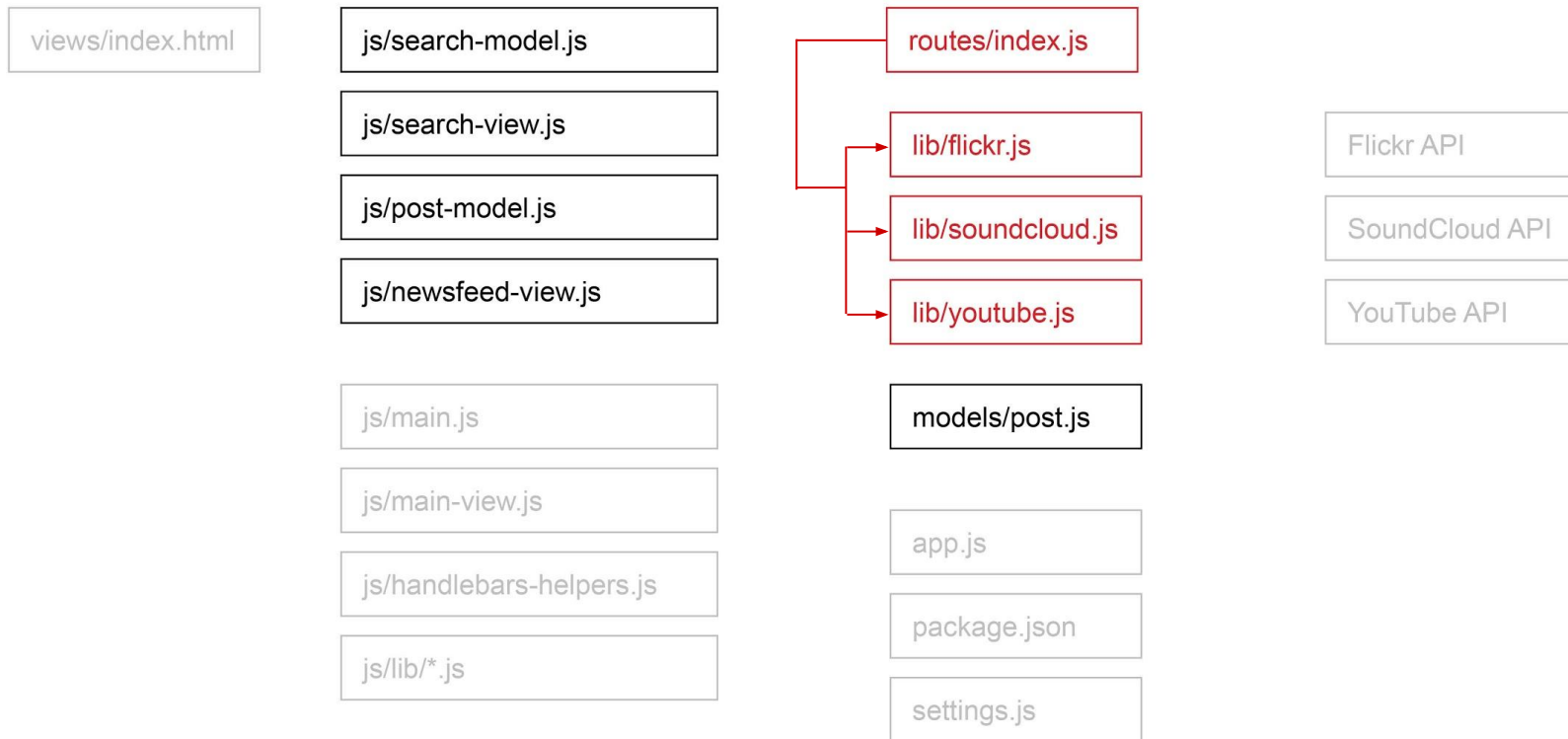
# Task: Interfacing with the APIs



# Tasks

1. Interfacing with each API to retrieve results
2. **Simultaneously query from all APIs**
3. Displaying the search results in the view
4. Implementing a REST API to add/remove/get posts from the newsfeed.
5. Update the posts in the view

# Task: Joint API



# Simultaneous Queries

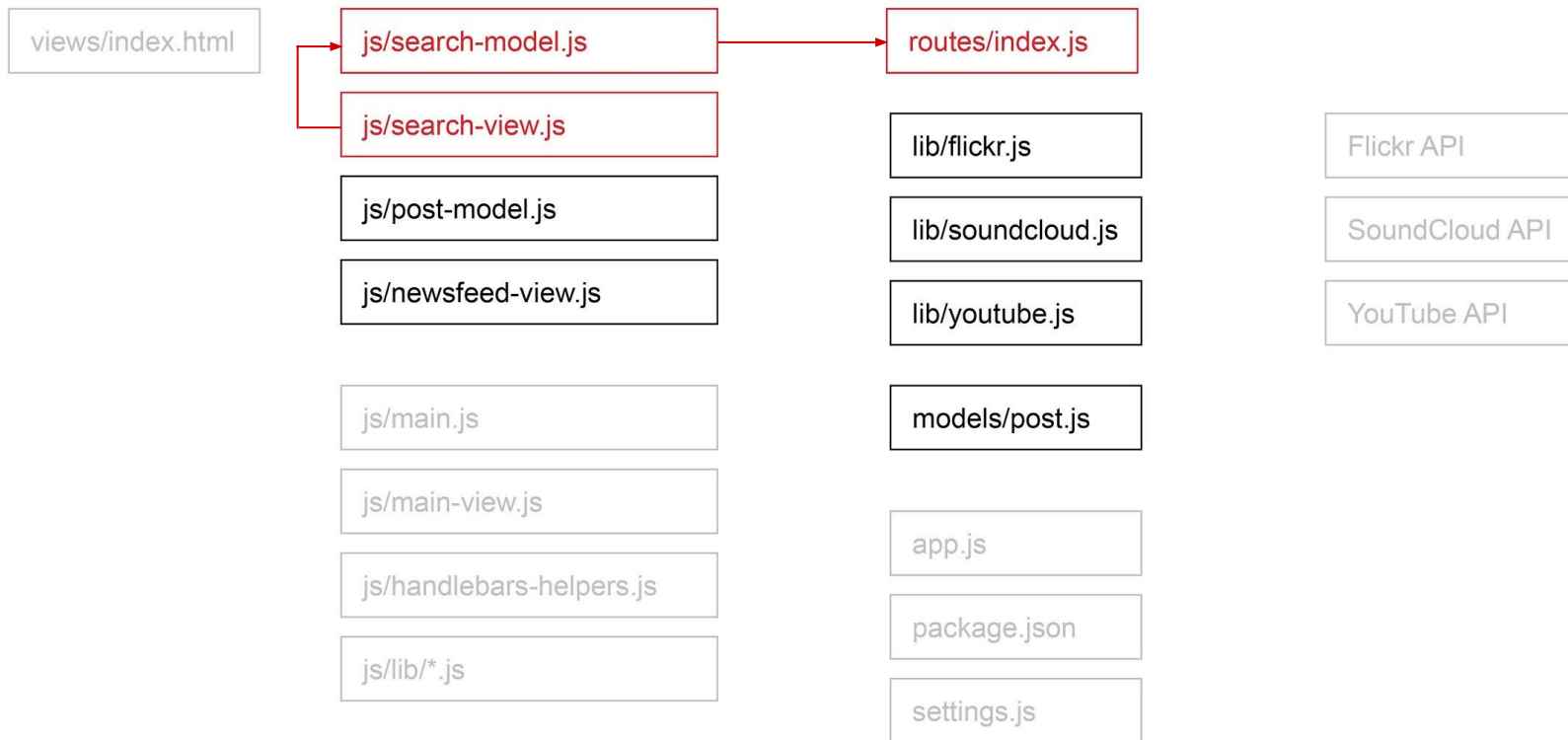
The API requests should be simultaneous; that is, your code should have the following form:

```
soundcloud.search(query, function() { ... });  
youtube.search(query, function() { ... });  
flickr.search(query, function() { ... });
```

# Tasks

1. Interfacing with each API to retrieve results
2. Simultaneously query from all APIs
3. **Displaying the search results in the view**
4. Implementing a REST API to add/remove/get posts from the newsfeed.
5. Update the posts in the view

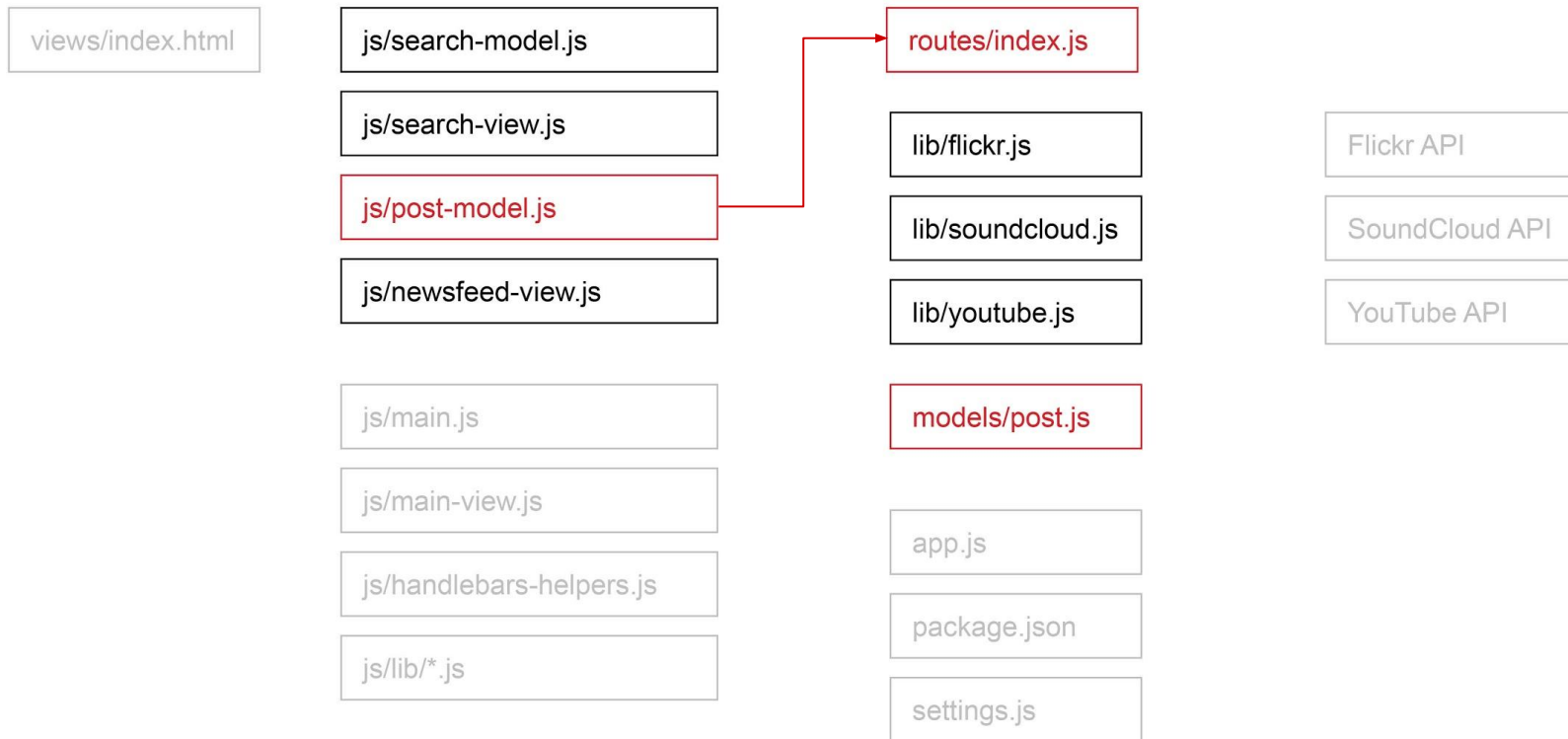
# Task: Displaying search results



# Tasks

1. Interfacing with each API to retrieve results
2. Simultaneously query from all APIs
3. Displaying the search results in the view
4. Implementing a REST API to add/remove/get posts from the newsfeed.
5. Update the posts in the view

# Task: REST API for Posts

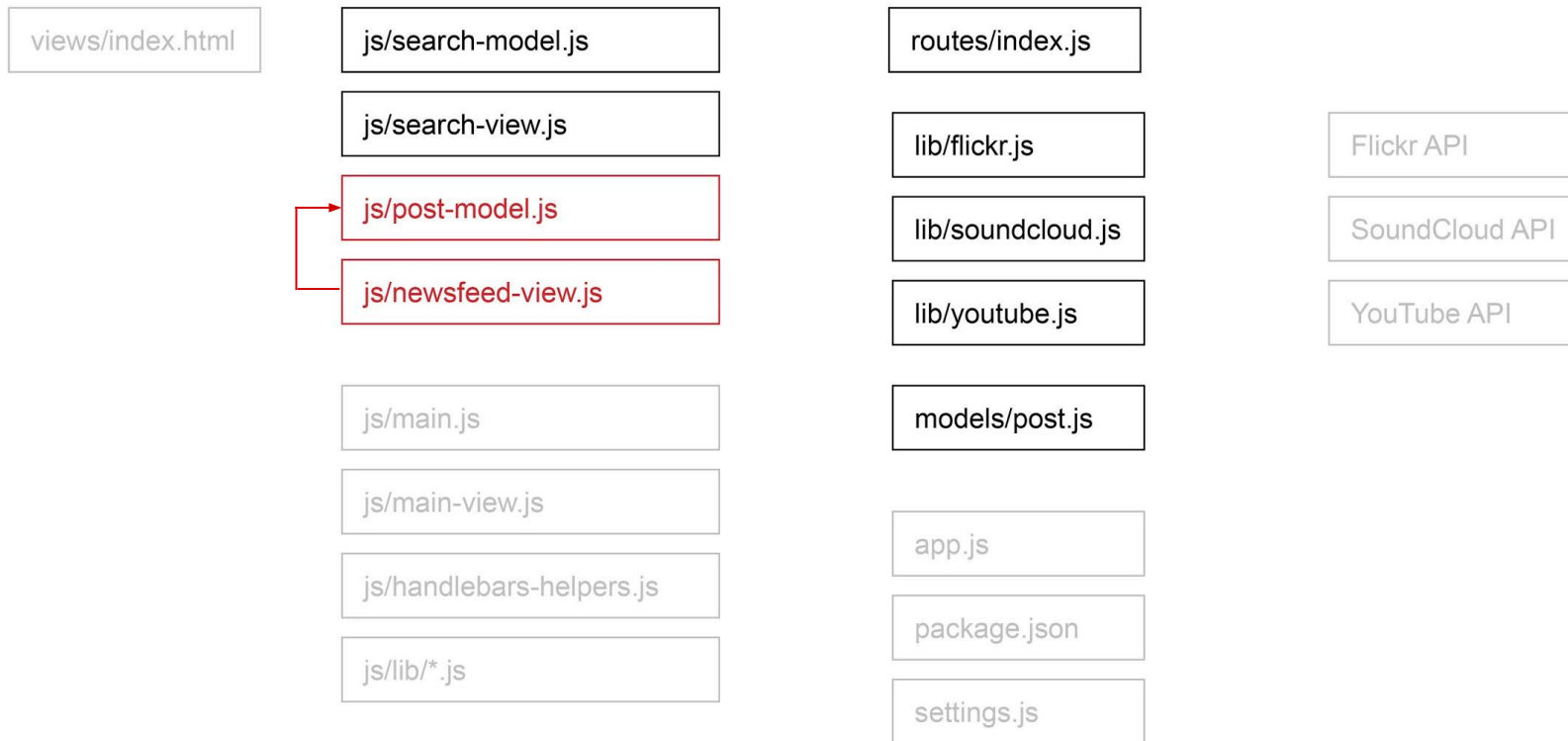




# Tasks

1. Interfacing with each API to retrieve results
2. Simultaneously query from all APIs
3. Displaying the search results in the view
4. Implementing a REST API to add/remove/get posts from the newsfeed.
5. Update the posts in the view

# Task: Updating the newsfeed view



# New Technologies

Developer APIs to retrieve media content:

YouTube: <<https://developers.google.com/youtube/v3/docs/#Search>>

Flickr: <<https://www.flickr.com/services/api/flickr.photos.search.html>>

SoundCloud: <<https://developers.soundcloud.com/docs/api/reference>>

MongoDB: <<http://docs.mongodb.org/manual/>>

Masonry: <<http://masonry.desandro.com/methods.html>>

# MongoDB

Database that allows information to persist

Organizes and structures data as **models**

**Models** are defined by **schemas**

We use **Mongoose** to interface with MongoDB in node.js applications

# Masonry

Cascading grid layout library

Places elements vertically and then horizontally