ARIA Tutorial :: Week 6 -- MVC Applications
Monday March 5th, 2012

Exercise 1 - Describe an MVC App

For this exercise, please get in groups of two.

You're going to take an existing application and break down the architecture into MVC components.

1) TAKE A LOOK AT AN MVC APPLICATION

Together with your team member, take a look at one of the following sites:

ToDo App: http://documentcloud.github.com/backbone/examples/todos/index.html
Notes App: http://mc-mobile-app.s3.amazonaws.com/index.html
Bit.ly App: http://addyosmani.com/resources/spinebitly/

Try using it.

What does it do?

If it's not obvious, keep clicking around until you figure out what it is doing.

2) BREAK DOWN THE DATA MODELS

You're going to create what you think the data model is.

As an example, imagine we had a site that listed albums and songs. The data might be an album, which can have a collection of songs plus one artist, plus one year, plus a length. And each song can have a writer, a composer, a length and a title.

Figure out which data components are used in your app and break them down into smaller pieces.

I would list my album example like this:

```
SONG:
        title -- string
        length -- int
        writer -- string
        composer -- string
        artist -- string
        trackNumber -- int

ALBUM:
```

multiple songs (array of songs)
artist -- string
albumTitle -- string
year -- int
length -- int

Create your list of data objects used in the application in a similar format.

3) BREAK DOWN THE VIEWS

The next step is to figure out what the views are.

In my albums example, I might have one view where the user could look at the different albums and one where they could look at the songs on an individual album.

My views might look like this:

AlbumCollectionView:
List of Albums including title, artist, year

AlbumIndividualView:
Songs on a given album, including title, track number, length

Or it could be more complex if I also have categories of music (reggae, classical, etc), guest performers on some tracks, etc.

Think about the application you have chosen. What are the views it uses? What are the different ways you can look at the data?

Create your list of views used in the application in a similar format.

4) BREAK DOWN THE CONTROLLERS

The final step is to figure out what the controllers are. This means, how can users interact with this data?
What user interface controls are there for them to click on.
Are there any?
Are there many?
Does it change depending on the data that's been added to the screen?

Going back to my albums example, I might have an application that just allows users to view the albums. Or I might have an application that allows them to add new albums and delete others.

What does your application let you do? How can you interact with what's on the screen for each different view?

Adding a track to an album is different from adding a new album to the list of albums.

My list of controllers might look something like this:

PAGE	EVENT	ACTION
main album view	Click on album	go to individual album view
main album view	Click on type	sort albums by type
individual album view	Click "+" icon	Add new track to album
individual album view	Click "-" icon	Delete track from album

5) COMPARE WITH OTHER GROUPS

Now that you've created lists of models, views and controllers, find another group who chose the same app & compare your information.

Do you have the same data model or is one more complex?

Did you both find all the possible views?

Are your lists of user actions for the controllers different or the same?