Dust

In this lab, you will work with a Dust template in an Kraken application using Makara and Adaro.

# Objectives

In this lab, you will

* extract a model from a static HTML page,
* create a Kraken app and add the two pages for our application,
* use Dust logic helpers and section tags to navigate your model,
* transform a static HTML page into a Dust template.

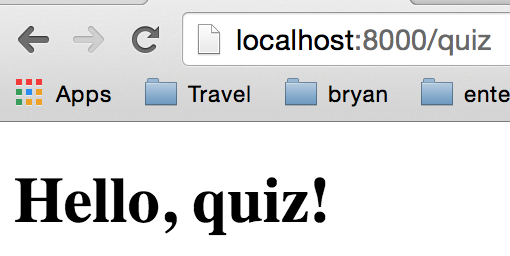
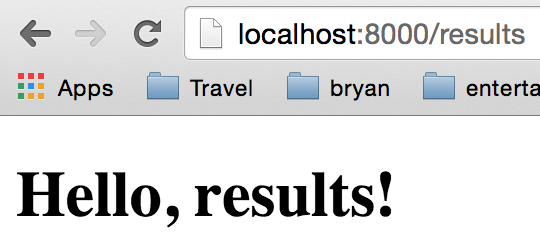
# Enable Dust templates in a Kraken application

In your lab directory you will see a folder named files. This contains the bulk of the code to create your application.

1. Open a shell, cd to the lab directory, and run yo kraken. Create an app named quiz. This will install all of the necessary modules in the lab directory in the folder, quiz.
2. By default, Kraken does not enable the dust helpers. Enable them by adding the dusjs-helpers configuration to the dust helpers section of config/config.json.

# Use yeoman to create the templates for the two pages.

1. Change to the folder, quiz. Use yeoman to create the two new pages along with their models and controllers. Use ‘yo kraken:controller quiz’ and ‘yo kraken:controller results’ to create the two pages.
2. Change to the folder lab/quiz and start the server with npm start. You should now have two pages as shown localhost:8000/quiz and localhost:8000/results as shown below. This indicates that Kraken is ready for our app.

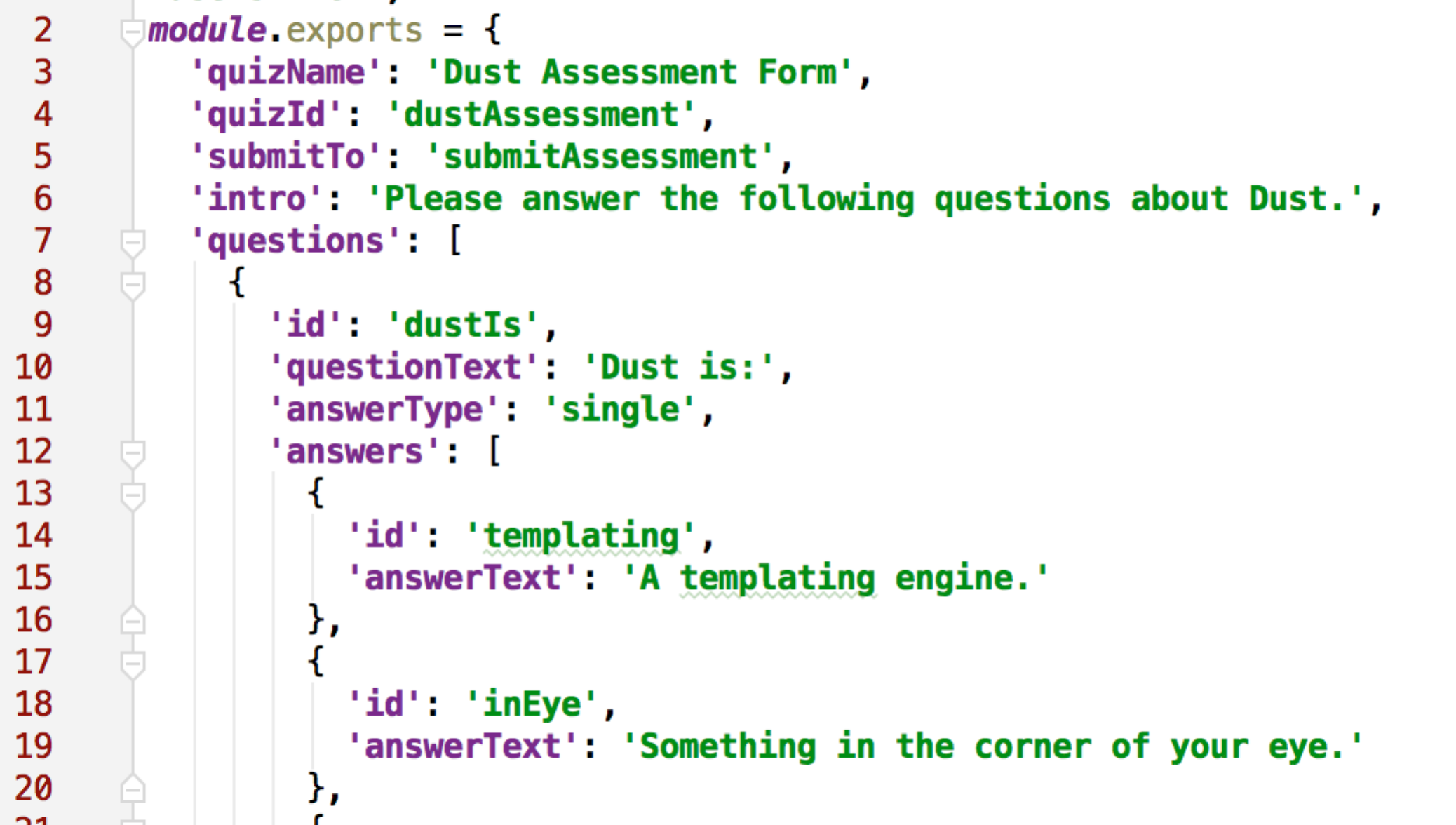
 

1. Copy the contents of the labs/files directory into the labs/quiz folder. This copies the model data shown below as well as the results controller and dust page.

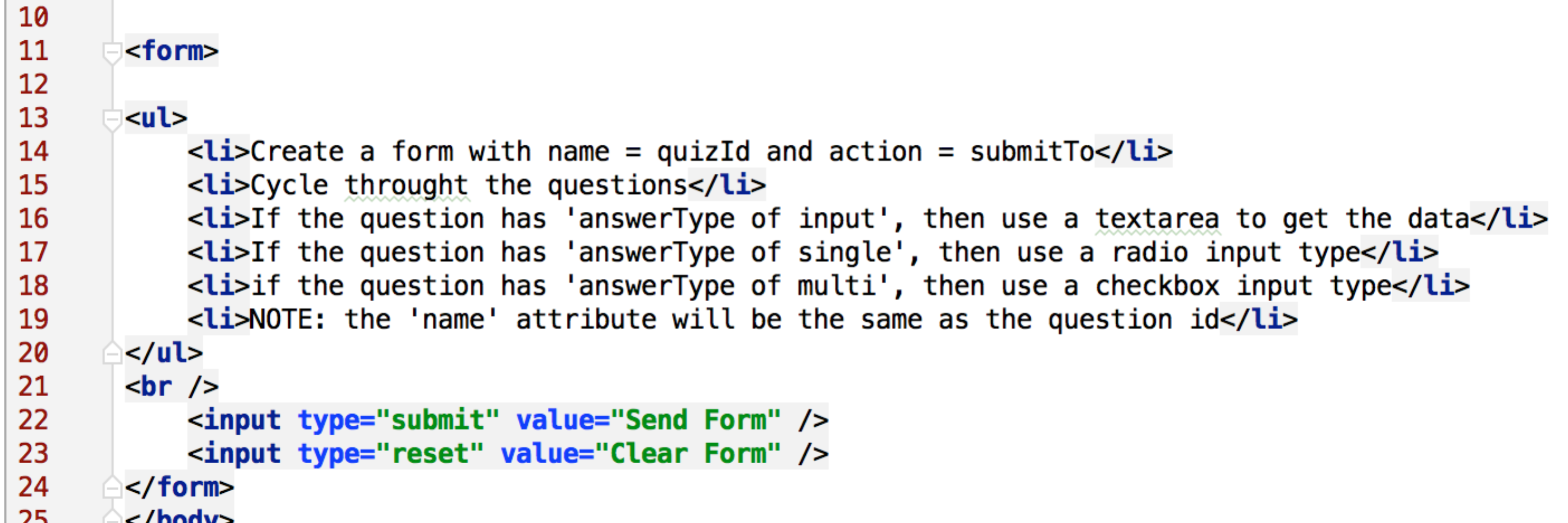
# Examine the files in the controllers folder

1. In the controllers folder, open the file results.js in an editor. Notice the code that creates a view model from the request body to make it easier for the results.dust page to render. This data will come directly from the quiz.dust form.
2. Open quiz.js.  This renders the quiz.dust page using the model data provided. res.render('quiz', model);
3. Examine your data model, models/quiz.js.

This defines a quiz with title, id, etc. Each quiz has multiple questions. Each question has text, type of question (single answer, multiple answers, input text) and answers. Each answer has an id and the answer text as shown below:



1. The initial page, public/templates/quiz.dust, is a set of instructions for how to build the dynamic quiz.dust page.



1. Open the original page, localhost:8000/original-page.html, our task is to create a dynamic page, quiz.dust, that mimics this original static page using the data model provided.

Make sure you have a hidden input field with the name, \_csrf*, with the value* {\_csrf}. Otherwise, you will have an error when you send the page to Kraken.

# Create the template logic

1. There are a number of places where you will want to use the easy iteration of arrays built into Dust sections. There's also a lot of similarity between a set of radio button inputs and a checkbox input. Consider using a logic helper.

There is no "right" solution to this lab. There is a tremendous amount of flexibility in Dust. Have fun coming up with surprises!

When you're finished page matches the original, this lab is complete!