

Dashboard Controller

Dashboard Controller

```
'use strict';

const logger = require('../utils/logger');

const dashboard = {
  index(request, response) {
    logger.info('dashboard rendering');
    const viewData = {
      title: 'Template 1 Dashboard',
    };
    response.render('dashboard', viewData);
  },
};

module.exports = dashboard;
```

Template 1

Dashboard

About

Title for Dashboard Panel

To be replaced with content...

strict mode javascript for
safety

```
'use strict';
```

Dashboard Controller

strict mode javascript for
safety

import the **logger**

```
'use strict';  
  
const logger = require('../utils/logger');
```

Dashboard Controller

strict mode javascript for
safety

import the **logger**

Index method, called when
'/dashboard' request received

```
'use strict';  
  
const logger = require('../utils/logger');  
  
const dashboard = {  
  index(request, response) {
```

Dashboard Controller

strict mode javascript for
safety

import the **logger**

Index method, called when
‘/dashboard’ request received

Log message to console so we
can track app behaviour

```
'use strict';

const logger = require('../utils/logger');

const dashboard = {

  index(request, response) {

    logger.info('dashboard rendering');
```

Dashboard Controller

strict mode javascript for
safety

import the **logger**

Index method, called when
‘/dashboard’ request received

Log message to console so we
can track app behaviour

Create **viewData** object to be
sent to view

```
'use strict';

const logger = require('../utils/logger');

const dashboard = {

  index(request, response) {

    logger.info('dashboard rendering');

    const viewData = {
      title: 'Template 1 Dashboard',
    };
  }
};
```

Dashboard Controller

strict mode javascript for
safety

import the **logger**

Index method, called when
‘/dashboard’ request received

Log message to console so we
can track app behaviour

Create **viewData** object to be
sent to view

Calls **render** method on
response with 2 parameters:

(1) name of view to render
(**dashboard**)

(2) object to inject into the
view prior to rendering it

```
'use strict';

const logger = require('../utils/logger');

const dashboard = {

  index(request, response) {

    logger.info('dashboard rendering');

    const viewData = {
      title: 'Template 1 Dashboard',
    };

    response.render('dashboard', viewData);

  },

};
```

Dashboard Controller

strict mode javascript for
safety

import the **logger**

Index method, called when
'/dashboard' request received

Log message to console so we
can track app behaviour

Create **viewData** object to be
sent to view

Calls **render** method on
response with 2 parameters:

(1) name of view to render
(**dashboard**)

(2) object to inject into the
view prior to rendering it

Export the **dashboard** object
so it can be used by the router

```
'use strict';

const logger = require('../utils/logger');

const dashboard = {

  index(request, response) {

    logger.info('dashboard rendering');

    const viewData = {
      title: 'Template 1 Dashboard',
    };

    response.render('dashboard', viewData);

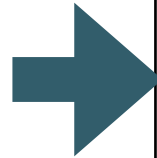
  },
};

module.exports = dashboard;
```

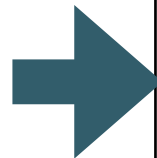
Dashboard Controller

Dashboard - Playlist 1

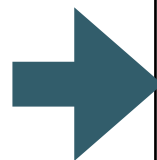
Load the
model (the
playlists)



Place model in
viewData
object



Send
viewData to
view



```
'use strict';

const logger = require('../utils/logger');
const playlistCollection = require('../models/playlist-store.js');

const dashboard = {

  index(request, response) {

    logger.info('dashboard rendering');

    const viewData = {
      title: 'Playlist Dashboard',
      playlists: playlistCollection,
    };

    logger.info('about to render', playlistCollection);

    response.render('dashboard', viewData);
  },
};

module.exports = dashboard;
```

Model/View

```
[
  {
    "title": "Beethoven Sonatas",
    "songs": [
      {
        "title": "Piano Sonata No. 3",
        "artist": "Beethoven"
      },
      {
        "title": "Piano Sonata No. 7",
        "artist": "Beethoven"
      },
      {
        "title": "Piano Sonata No. 10",
        "artist": "Beethoven"
      }
    ]
  },
  {
    "title": "Beethoven Concertos",
    "songs": [
      {
        "title": "Piano Concerto No. 0",
        "artist": "Beethoven"
      },
      {
        "title": "Piano Concerto No. 4",
        "artist": "Beethoven"
      },
      {
        "title": "Piano Concerto No. 6",
        "artist": "Beethoven"
      }
    ]
  }
]
```



```
index(request, response) {

  const viewData = {
    title: 'Playlist Dashboard',
    playlists: playlistCollection,
  };

  response.render('dashboard', viewData);
}
```

```
{{> menu id='dashboard'}}
{{#each playlists}}
  <section class="ui segment">
    <h2 class="ui header">
      {{title}}
    </h2>
    <table class="ui table">
      <thead>
        <tr>
          <th>Song</th>
          <th>Artist</th>
        </tr>
      </thead>
      <tbody>
        {{#each songs}}
          <tr>
            <td>
              {{title}}
            </td>
            <td>
              {{artist}}
            </td>
          </tr>
        {{/each}}
      </tbody>
    </table>
  </section>
{{/each}}
```

Loop through
each playlist
object

Output the title

Loop through each
song in each
playlist

Output title + artist

Beethoven Sonatas

Song	Artist
Piano Sonata No. 3	Beethoven
Piano Sonata No. 7	Beethoven
Piano Sonata No. 10	Beethoven

Beethoven Concertos

Song	Artist
Piano Concerto No. 0	Beethoven
Piano Concerto No. 4	Beethoven
Piano Concerto No. 6	Beethoven