

```
ember new <name_of_the_app>
```

```
ember new loopylog
```

To start server:

```
=====
```

```
ember s
```

```
ember generate
```

```
ember g template application
```

### application.hbs

```
<h1>My Application</h1>
{{#link-to "production"}}production{{/link-to}}
{{outlet}}
```

```
ember g route production
```

### production.hbs

```
<h2>Production</h2>
```

To install addon

```
=====
```

```
ember install ember-pikaday
```

-->once install the add ons server has to refresh. so once aging enter <<ember s>> to start server

### production.hbs

```
<h2>Production</h2>
{{pikaday-input format="MM/DD/YYYY"}}
```

```
ember build --environment production
```

Check with python simple HTTP server:

```
=====
cd dist
```

```
python -m http.server 8000
```

```
python -m SimpleHTTPServer
```

Add more plugins

=====

```
ember install ember-cli-sass
```

```
ember install ember-cli-bourbon
```

```
ember install ember-font-awesome
```

restart server:

=====

```
ember s
```

update your **application.hbs** file which includes

```
<nav class="title">
```

```
<h1> Heading</h1>
```

```
</nav>
```

```
<main>
```

```
{{outlet}}
```

```
</main>
```

updated app.scss file with the spec styles.

check if the styles changed

### optional

if styles not changed.

```
rmdir /s node_modules bower_components dist tmp
```

```
npm install
```

```
bower install
```

generate route for dimension page:

```
=====
```

```
ember g route production/dimension
```

add model in production.js

```
=====
```

```
model() {  
  return {  
    "name" : "Tom",  
    "age":23,  
    "gender" : "Female"  
  }  
}
```

Add the following in the production.hbs file:

```
<h2>Production</h2>
```

```
<p>Name : {{model.name}}</p>
```

```
<p>Age: {{model.age}}</p>
```

```
<p>Gender : {{model.gender}}</p>
```

```
mkdir public\images
```

```
mkdir public\data
```

```
copy image into images
```

```
copy employee.json into data
```

### production.js

```
import Ember from 'ember';
```

```
export default Ember.Route.extend({  
  model() {  
    /*return {  
      "name" : "Tom",  
      "age":23,  
      "gender" : "Female"  
    }*/  
    return Ember.$.getJSON("/data/employee.json");  
  }  
});
```

### production.hbs

```
<h2>Production</h2>
```

```
<p>Name : {{model.[10].name}}</p>
```

```
<p>Age: {{model.[10].age}}</p>
```

```
<p>Gender : {{model.[10].gender}}</p>
```

nested routes:

ember g route production/dimension

### production.hbs

```
<h2>Production</h2>

<p>Product : {{model.[10].ProductName}}</p>
<p>Boards: {{model.[10].BoardsSum}}</p>
<p>BoardFeet : {{model.[10].BoardFeetSum}}</p>
<p>{{#link-to 'production.dimension' }}Dimension :
{{model.[10].DimensionName}} {{/link-to}}</p>

{{outlet}}
```

### dimension.hbs

```
<h3>Dimension Details</h3>
{{outlet}}
```

### Dynamic Segments:

### production.js

```
import Ember from 'ember';

export default Ember.Route.extend({
  model(params) {
    /*return {
      "name" : "Tom",
      "age":23,
      "gender" : "Female"
    }*/
    return
    Ember.$.getJSON(`/data/production.json?start=${params.start}&end=${params.end}`);
  }
});
```

### router.js

```
import Ember from 'ember';
import config from './config/environment';
```

```

const Router = Ember.Router.extend({
  location: config.locationType,
  rootURL: config.rootURL
});

Router.map(function() {
  this.route('production', {path: "/*:start/to/*:end"}, function()
  {
    this.route('dimension', {path: "/*dimension/*:dimension_id"});
  });
});

export default Router;

```

### production.hbs

```

<h2>Production</h2>

<p>Product : {{model.[10].ProductName}}</p>
<p>Boards: {{model.[10].BoardsSum}}</p>
<p>BoardFeet : {{model.[10].BoardFeetSum}}</p>
<p>{{#link-to 'production.dimension'
model.[10].DimensionID}}Dimension : {{model.[10].DimensionName}}
{{/link-to}}</p>

{{outlet}}

```

### dimension.hbs

```

<h3>Dimension : {{model.DimensionName}}</h3>
{{outlet}}

```

Use the below link in browser:

<http://localhost:4200/yesterday/to/today>

**Index Routes:**

ember g template production/index

### index.hbs

```
<p>Product : {{model.[10].ProductName}}</p>
<p>Boards: {{model.[10].BoardsSum}}</p>
<p>BoardFeet : {{model.[10].BoardFeetSum}}</p>
<p>{{#link-to 'production.dimension'
model.[10].DimensionID}}Dimension : {{model.[10].DimensionName}}
{{/link-to}}</p>
```

### production.hbs

```
<h2>Production</h2>
```

```
{{outlet}}
```

Check the below URL again in browser:

<http://localhost:4200/yesterday/to/today>

### Redirecting and Loading:

ember g route index

### index.js

```
import Ember from 'ember';

export default Ember.Route.extend({
  beforeModel() {
    this.transitionTo('production', "yesterday", "today");
  }
});
```



ember g template production-loading

production-loading.hbs

LOADING!!!

copy loading.gif into public/images folder

production-loading.hbs

```
<div class="centroid">
  <br>
  Loading Production data...
</div>
```

Update production.js file as below:

production.js

```
import Ember from 'ember';
const Promise = Ember.RSVP.Promise;
export default Ember.Route.extend({
  model(params) {
    /*return {
      "name" : "Tom",
      "age":23,
      "gender" : "Female"
    }*/
    //return
    Ember.$.getJSON(`/data/production.json?start=${params.start}&end=${params.end}`);
    return new Promise(function(resolve) {
      setTimeout(function() {
        let data =
Ember.$.getJSON(`/data/production.json?starts_at=${params.start}&ends_at=${params.end}`);
        resolve(data);
      }, 400);
    });
  }
});
```

```
}  
});
```

## Using Templates Properly

### Loops and Conditions

Create index.hbs file under template/production

Change index.hbs file under templates/production/index.hbs

#### index.hbs

```
<table class="table">  
  <thead>  
    <tr>  
      <th>Dimension</th>  
      <th>Boards</th>  
      <th>BoardFeet</th>  
    </tr>  
  </thead>  
  <tbody>  
    {{#each model as |row|}}  
      <tr>  
        <td>  
          {{#link-to "production.dimension" row.DimensionID}}  
            {{row.DimensionName}}  
          {{/link-to}}  
        </td>  
        <td class="quantity">{{row.BoardsSum}}</td>  
        <td class="quantity">{{row.BoardFeetSum}}</td>  
      </tr>  
    {{/each}}  
  </tbody>  
</table>
```

#### Production.hbs

```
<h2>Production</h2>
```

```
{{outlet}}
```

## Apply if condition

### index.hbs

```
<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>

    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}
      <tr class="{{if row.BoardsSum "" "Zero"}}">
        <td>
          {{#link-to "production.dimension" row.DimensionID}}
            {{row.DimensionName}}
          {{/link-to}}
        </td>
        <td class="quantity">{{row.BoardsSum}}</td>
        <td class="quantity">{{row.BoardFeetSum}}</td>

      </tr>
    {{/each}}
  </tbody>
</table>
```

ember install ember-truth-helpers

and restart server

ember s

index.hbs

```
<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>

    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}
      <tr class="{{if (lt row.BoardsSum 10) (if (gt row.BoardsSum
5) "Zero" "") "" }}">
        <td>
          {{#link-to "production.dimension" row.DimensionID}}
            {{row.DimensionName}}
          {{/link-to}}
        </td>
        <td class="quantity">{{row.BoardsSum}}</td>
        <td class="quantity">{{row.BoardFeetSum}}</td>

      </tr>
    {{/each}}
  </tbody>
</table>
```

Negation can be mentioned with *unless* keyword

```
{{unless (lt row.BoardsSum 10) "Zero" ""}}
```

Apply Boolean value:

index.hbs

```
<table class="table">
  <thead>
    <tr>
```

```

    <th>Dimension</th>
    <th>Boards</th>
    <th>BoardFeet</th>
    <th>{{fa-icon "cogs"}}</th>
  </tr>
</thead>
<tbody>
  {{#each model as |row|}}
    <tr class="{{if (lt row.BoardsSum 10) (if (gt row.BoardsSum
5) "Zero" "") ""}}">
      <td>
        {{#link-to "production.dimension" row.DimensionID}}
          {{row.DimensionName}}
        {{/link-to}}
      </td>
      <td class="quantity">{{row.BoardsSum}}</td>
      <td class="quantity">{{row.BoardFeetSum}}</td>
      <td><input type="checkbox" disabled={{eq row.BoardsSum
0}}></td>
    </tr>
  {{/each}}
</tbody>
</table>

```

## Named Arguments:

For number formatting use below add-ons

ember install ember-cli-accounting

ember s → to restart

## index.hbs

```

<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>
      <th>{{fa-icon "cogs"}}</th>
    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}

```

```

5) <tr class="{{if (lt row.BoardsSum 10) (if (gt row.BoardsSum
    <td>
      {{#link-to "production.dimension" row.DimensionID}}
      {{row.DimensionName}}
      {{/link-to}}
    </td>
    <td class="quantity">{{row.BoardsSum}}</td>
    <td class="quantity">{{format-number row.BoardFeetSum
precision=2}}</td>
    <td><input type="checkbox" disabled={{eq row.BoardsSum
0}}></td>
  </tr>
  {{/each}}
</tbody>
</table>

```

## Create a Helper:

ember g helper row-class

ember s

you will be getting row-class.js under helpers/row-class.js

do the below changes

### row-class.js

```

import Ember from 'ember';

export function rowClass(params/*, hash*/) {
  if (params[0]) {
    var row = params[0],
        class_names = [];
    if (row.Boards === 0 || row.BoardsSum === 0) {
      class_names.push("zero");
    }
    //other conditions

    return class_names.join(" ");
  }
}

```

```

    return '';
}

export default Ember.Helper.helper(rowClass);

```

## index.hbs

```

<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>
      <th>{{fa-icon "cogs"}}</th>
    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}
      <tr class="{{row-class row}}">
        <td>
          {{#link-to "production.dimension" row.DimensionID}}
            {{row.DimensionName}}
          {{/link-to}}
        </td>
        <td class="quantity">{{row.BoardsSum}}</td>
        <td class="quantity">{{format-number row.BoardFeetSum
precision=2}}</td>
        <td><input type="checkbox" disabled={{eq row.BoardsSum
0}}></td>
      </tr>
    {{/each}}
  </tbody>
</table>

```

## Debugging

log and debugger

## index.hbs

```
<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>
      <th>{{ fa-icon "cogs" }}</th>
    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}
      <tr class="{{row-class row}}">
        <td>
          {{#link-to "production.dimension" row.DimensionID}}
            {{row.DimensionName.length}}
          {{/link-to}}
        </td>
        <td class="quantity">{{row.BoardsSum}}</td>
        <td class="quantity">{{format-number row.BoardFeetSum
precision=2}}</td>
        <td><input type="checkbox" disabled={{eq row.BoardsSum
0}}></td>
      </tr>
    {{/each}}
  </tbody>
</table>
```

## Add log

```
<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>
      <th>{{ fa-icon "cogs" }}</th>
    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}
```



```

<tr class="{{row-class row}}">
  <td>
    {{#link-to "production.dimension" row.DimensionID}}
      {{log row.DimensionName}}
      {{log row.DimensionName.length}}
    {{/link-to}}
  </td>
  <td class="quantity">{{row.BoardsSum}}</td>
  <td class="quantity">{{format-number row.BoardFeetSum
precision=2}}</td>
  <td><input type="checkbox" disabled={{eq row.BoardsSum
0}}></td>
</tr>
{{/each}}
</tbody>
</table>

```

### Add debugger

It will add break point

```

<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>
      <th>{{fa-icon "cogs"}}</th>
    </tr>
  </thead>
  <tbody>
    {{#each model as |row|}}
      <tr class="{{row-class row}}">
        <td>
          {{#link-to "production.dimension" row.DimensionID}}
            {{log row.DimensionName.length}}
          {{/link-to}}
          {{debugger}}
        </td>
        <td class="quantity">{{row.BoardsSum}}</td>
        <td class="quantity">{{format-number row.BoardFeetSum
precision=2}}</td>
        <td><input type="checkbox" disabled={{eq row.BoardsSum
0}}></td>

```

```
</tr>
{{/each}}
</tbody>
</table>
```

*Use below in console:*

1. *row*
2. *get("row")*
3. *get("row").DimensionName*

When you click dimension it will move to details.

Update routes → production → dimension.js

### dimension.js

```
import Ember from 'ember';

export default Ember.Route.extend({
  model(params) {
    //return
    Ember.$.getJSON('/data/production.json'...).then(...);
    return this.modelFor('production')
      .findBy('DimensionID', parseInt(params.dimension_id));
  }
});
```

### Ember.object

Add computed property as well

ember g model production

update model/production.js

### production.js

```
import Ember from 'ember';

export default Ember.Object.extend({
  boards: Ember.computed('BoardsSum', function() {
    return this.get('BoardsSum');
  }),
  boardfeet: Ember.computed.alias('BoardFeetSum'),
  volume: Ember.computed('NominalLength', 'NominalWidth',
    'NominalThickness', function() {
    return this.get('NominalLength') * 12 *
    this.get('NominalWidth') * this.get('NominalThickness');
  })
});
```

update routes/production.js

### production.js

```

import Ember from 'ember';
const Promise = Ember.RSVP.Promise;
import Production from 'loopyapp/models/production';

export default Ember.Route.extend({
  model(params) {
    // return {
    //   "ProductName": "Ponderosa",
    //   "DimensionName": "1\" X 4\" X 14'",
    //   "BoardsSum": 20,
    //   "BoardFeetSum": 10000
    // }
    // return
    Ember.$.getJSON(`/data/production.json?start=${params.start}&end=${params.end}`);
    return new Promise(function(resolve) {
      setTimeout(function() {

Ember.$.getJSON(`/data/production.json?starts_at=${params.start}
&ends_at=${params.end}`, function(data) {
        let records = [];
        data.forEach(function(item) {
          records.push( Production.create(item) );
        });
        resolve(records);
      });
    }, 400);
  });
});

```

## Create controller

ember g controller production/index

update controller/production/index.js

### index.js

```

import Ember from 'ember';

export default Ember.Controller.extend({

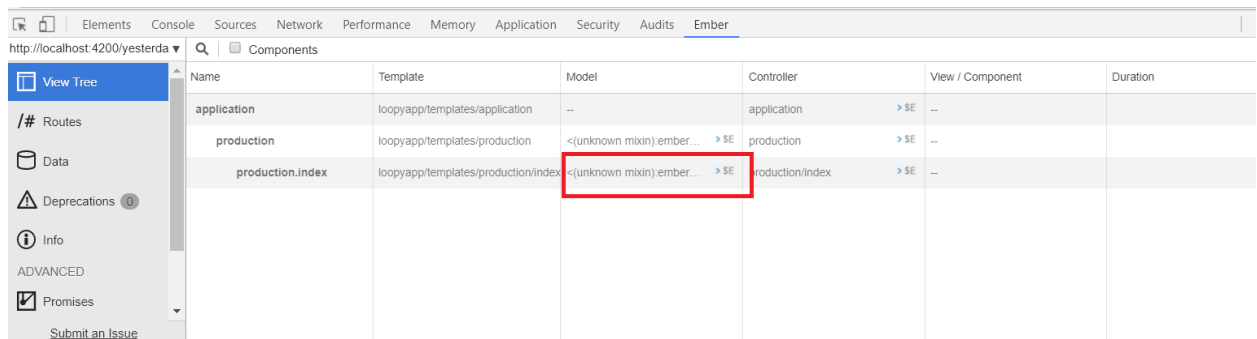
```

```

// total_boards: Ember.computed('model.@each.boards',
function() {
//   return this.get('model').reduce(function(prev, row) {
//     return prev + parseFloat(row.get('boards'));
//   }, 0);
// }));
boards: Ember.computed.mapBy('model', 'boards'),
total_boards: Ember.computed.sum('boards'),
max_boards: Ember.computed.max('boards'),
boardfeets: Ember.computed.mapBy('model', 'boardfeet'),
total_boardfeet: Ember.computed.sum('boardfeets'),
max_boardfeet: Ember.computed.max('boardfeets')
});

```

1. First work with commented statements
2. Go to chrome and open inspector
  - a. Under Ember → View Tree→Model



3. Select \$E under model → production.index as highlighted above.
4. In the console enter below commands
 

```

$E[0]
$E[0].get('boards')
$E[0].set('boards',5)

```
5. This will not update the total, now enable the highlighted code. Now you can see total gets changed.

## Reusable components

ember g component table-production

1. Go to template→production→ index.hbs

Replace the code as

index.hbs

```
{{table-production model=model}}
```

2. update template→ component→ table-production.hbs file as follows

table-production.hbs

```
<table class="table">
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
      <th>BoardFeet</th>
      <th>{{fa-icon "cogs"}}</th>
    </tr>
  </thead>
  <tbody>
    <tr class="total">
      <td>All Dimensions</td>
      <td>{{total_boards}}</td>
      <td>{{format-number total_boardfeet precision=2}}</td>
      <td></td>
    </tr>
    {{#each model as |row|}}
      <tr class="{{row-class row}}">
        <td>
          {{#link-to "production.dimension" row.DimensionID}}
            {{row.DimensionName}}
          {{/link-to}}
        </td>
        <td class="quantity">{{row.boards}}</td>
        <td class="quantity">{{format-number row.boardfeet
precision=2}}</td>
        <td><input type="checkbox" disabled={{eq row.boards
0}}></td>
      </tr>
    {{/each}}
```

```

    {{/each}}
  </tbody>
</table>

```

Note: Now if you check the browser Total will not appear

### Production

Dimension	Boards	BoardFeet
All Dimensions		0.00
1" X 4" X 6'	13	26.00
1" X 4" X 8'	14	37.33
1" X 4" X 10'	10	33.33
1" X 4" X 12'	12	48.00
1" X 4" X 14'	8	37.33
1" X 4" X 16'	17	90.67
1" X 6" X 6'	6	18.00
1" X 6" X 8'	25	100.00

3. Comment the computed property from controller

Controller → production → index.js

#### index.js

```

import Ember from 'ember';

export default Ember.Controller.extend({
  // total_boards: Ember.computed('model.@each.boards',
function() {
  //   return this.get('model').reduce(function(prev, row) {
  //     return prev + parseFloat(row.get('boards'));
  //   }, 0);
  // }),
/* boards: Ember.computed.mapBy('model', 'boards'),
total_boards: Ember.computed.sum('boards'),
max_boards: Ember.computed.max('boards'),
boardfeets: Ember.computed.mapBy('model', 'boardfeet'),
total_boardfeet: Ember.computed.sum('boardfeets'),
max_boardfeet: Ember.computed.max('boardfeets')*/
});

```

4. Move that code to component → table-production.js

## table-production.js

```
import Ember from 'ember';

export default Ember.Component.extend({
  boards: Ember.computed.mapBy('model', 'boards'),
  total_boards: Ember.computed.sum('boards'),
  max_boards: Ember.computed.max('boards'),
  boardfeets: Ember.computed.mapBy('model', 'boardfeet'),
  total_boardfeet: Ember.computed.sum('boardfeets'),
  max_boardfeet: Ember.computed.max('boardfeets')
});
```

## Changing the tag Name

Component → table-production.js

## table-production.js

```
import Ember from 'ember';

export default Ember.Component.extend({
  tagName: 'table',
  classNames: ['table'],
  boards: Ember.computed.mapBy('model', 'boards'),
  total_boards: Ember.computed.sum('boards'),
  max_boards: Ember.computed.max('boards'),
  boardfeets: Ember.computed.mapBy('model', 'boardfeet'),
  total_boardfeet: Ember.computed.sum('boardfeets'),
  max_boardfeet: Ember.computed.max('boardfeets')
});
```

templates → components → table-production.hbs

## table-production.hbs

```
<!--<table class="table">-->
  <thead>
    <tr>
      <th>Dimension</th>
      <th>Boards</th>
```



```

    <th>BoardFeet</th>
    <th>{{fa-icon "cogs"}}</th>
  </tr>
</thead>
<tbody>
<tr class="total">
  <td>All Dimensions</td>
  <td>{{total_boards}}</td>
  <td>{{format-number total_boardfeet precision=2}}</td>
  <td></td>
</tr>
{{#each model as |row|}}
  <tr class="{{row-class row}}">
    <td>
      {{#link-to "production.dimension" row.DimensionID}}
        {{row.DimensionName}}
      {{/link-to}}
    </td>
    <td class="quantity">{{row.boards}}</td>
    <td class="quantity">{{format-number row.boardfeet
precision=2}}</td>
    <td><input type="checkbox" disabled={{eq row.boards
0}}></td>
  </tr>
{{/each}}
</tbody>
<!--
</table>-->

```

## Handling a block

Create another component  
ember g component table-th

1. Components → table-th.js

### table-th.js

```

import Ember from 'ember';

export default Ember.Component.extend({
  tagName: 'th'
});

```

2. templates→components→table-th.hbs

### table-th.hbs

```
{{#if hasBlock}}
  {{yield}}
{{else}}
  {{name}}
{{/if}}
```

3. templates→components→table-production.hbs

### table-production.hbs

```
<!--<table class="table">-->
<thead>
  <tr>
    {{table-th name="Dimension"}}
    {{table-th name="Boards"}}
    {{table-th name="BoardFeet"}}
    {{#table-th}}{{fa-icon "cogs"}}{{/table-th}}
  </tr>
</thead>
<tbody>
  <tr class="total">
    <td>All Dimensions</td>
    <td>{{total_boards}}</td>
    <td>{{format-number total_boardfeet precision=2}}</td>
    <td></td>
  </tr>
  {{#each model as |row|}}
    <tr class="{{row-class row}}">
      <td>
        {{#link-to "production.dimension" row.DimensionID}}
          {{row.DimensionName}}
        {{/link-to}}
      </td>
      <td class="quantity">{{row.boards}}</td>
      <td class="quantity">{{format-number row.boardfeet
precision=2}}</td>
      <td><input type="checkbox" disabled={{eq row.boards
0}}></td>
    </tr>
  {{/each}}
</tbody>
<!--
</table>-->
```