

Dust.js templates

Prof. Cesare Pautasso

<http://www.pautasso.info>

cesare.pautasso@usi.ch

[@pautasso](#)

Dust.js

- Good compromise between pure, logic-less templates and embedded Javascript templates
- Uses full HTML markup (only required to learn dust.js notation)
- Templates can be precompiled
- Runs both on client and server-side

{value}

```
{ name: "Peggy"                                     json
, address: { street: "via Buffi"
              , number: 13 }
}
```

```
{name}<br/>
{address.street} {address.number}                  dust
```

```
Peggy<br/>
via Buffi 13                                       HTML
```

Replace the JSON element with its value

{#array}

```
{ names: [ "a", "b", "c" ]                          json
}
```

```
{#names}
{.}
{/names}                                           dust
```

```
a b c                                             HTML
```

Iterate over all elements of the names array

Use once if names is not an array

Skip if names does not exist in the model

{#array}

```
{ users: [ {name: "a"}
            , {name: "b"}
            , {name: "c"} ]
}
```

json

```
{#users}
  {$idx}. {name}
{:else}
  no users found
{/users}
```

dust

```
1. a
2. b
3. c
```

HTML

Control what happens if the array is empty with `{:else}`

Predefined counters: `$idx`, `$len`

Conditional Section

```
{ name: "Peggy"
  , address: { street: "via Buffi"
              , number: 13 }
}
```

json

```
{^telephone}
  {! telephone missing !}
{/telephone}
{?address}
  {! address present !}
{/address}
```

dust

Decide whether to include parts of the template depending on whether elements are found (?) or missing (^) in the model

{>partial /}

Reuse templates by including them within other templates

```
{>"template_name" /} dust
```

```
{>"template_{name}" /}
```

The choice of which template to include can be dynamic, based on a value found in the model

Parameters

Pass parameters to sections and partials

```
{#section param="value"} dust
```

String constant parameter

```
{>"template_name" param=value /}
```

Parameter value taken from model named value

Where to render?

- on the Client:
 - Offload work from the server
 - Separate API from UI
 - Performance Optimization (render only the part of the page that is visible to the user, cache templates on clients)
- on the Server:
 - Some clients cannot render by themselves (old browsers, search engine crawlers)

Rendering on the client

```
dust.loadSource(  
  dust.compile('template code', "template_name")  
);
```

Compile on the client and load the template once

```
<script src="/views/template_name.js"></script>
```

Precompile on the server and load the template on the client

```
dust.render("template_name",  
  { /* model json to be rendered */ },  
function(err, out) {  
  if (!err) {  
    //show the result of the model rendering  
    view.innerHTML = out;  
  }  
});
```

Asynchronously render the model using the template whenever the view needs refreshing

References

- [Dust.JS Documentation \(http://akdubya.github.io/dustjs/\)](http://akdubya.github.io/dustjs/)