

INF4375 - Paradigmes des échanges Internet

Préprocesseur Pug

Jacques Berger

# Objectifs

Introduire le préprocesseur Pug

# Prérequis

Javascript

HTML

# Templating

Générer du code HTML en faisant systématiquement des `res.write(...)`, ce n'est pas l'idéal

Cette pratique fonctionne mais rend le code particulièrement difficile à lire et à maintenir

# Templating

Pour favoriser la maintenance des applications, on tente de n'avoir qu'un seul langage de programmation dans un fichier source

Si on génère du HTML avec Node.js, nous avons deux langages dans le même fichier : HTML et Javascript

# Templating

On peut donc extraire la partie HTML dans un fichier distinct et l'invoquer à partir du Javascript

Le HTML devient un template (gabarit)

# Templating

Il existe beaucoup de langages de templating qui permettent de générer du HTML

# Pug

Langage de templating permettant de générer du HTML

Assez répandu dans la communauté Javascript

Se nommait Jade auparavant

Express.js utilise encore Jade par défaut



# Pug

Pug est disponible au travers de npm :

```
npm install pug
```

# Pug

Extension habituelle : .pug

Pas de MIME type car on ne transfère pas le document Pug mais bien le HTML généré

# Syntaxe

C'est du HTML de haut niveau, sans les balises fermantes et les irritants syntaxiques du HTML

# Syntaxe

On écrit seulement le nom de la balise à générer

head

devient

`<head></head>`

# Syntaxe

L'utilisation des attributs id est facilité

```
span#result-container
```

devient

```
<span id='result-container'></span>
```

# Syntaxe

## Même chose avec les classes

`button.btn-small`

devient

`<button class='btn-small'></button>`

# Syntaxe

Lorsque le nom d'élément est omis, un div est assumé par défaut

```
.text-error
```

devient

```
<div class='text-error'></div>
```

# Syntaxe

Les id et les classes peuvent être combinés sur un même élément

```
#result-container.text-success.pull-left
```

devient

```
<div id='result-container' class='text-success pull-left'></div>
```



# Syntaxe

## L'indentation remplace l'imbrication des éléments

```
html
  head
    title Un joli titre
```

## devient

```
<html>
  <head>
    <title>Un joli titre</title>
  </head>
</html>
```

# Syntaxe

Pour placer du texte dans un élément

`span INF4375`

devient

`<span>INF4375</span>`

# Syntaxe

Le texte sur la ligne suivante ou plusieurs lignes

```
p
| Une première ligne
| Une deuxième ligne
```

devient

```
<p>Une première ligne Une deuxième ligne</p>
```

# Syntaxe

## Utilisation de variables Javascript

Si le template est invoqué avec l'objet suivant en paramètre :  
`{ 'user': 'Jacques' }`

`span #{user}`

ou

`span= user`

## devient

`<span>Jacques</span>`

# Syntaxe

## Commentaires HTML

```
// Ceci est un commentaire
```

## devient

```
<!-- Ceci est un commentaire -->
```

# Syntaxe

## Commentaires Pug

```
//- Ceci est un commentaire
```

Ceci n'apparaît pas dans le HTML généré

# Syntaxe

## Les attributs des éléments

`a(href='http://www.jberger.org/') Cliquez ici`

devient

`<a href='http://www.jberger.org/'>Cliquez ici</a>`

# Syntaxe

Les attributs doivent être séparés par une virgule

```
td(colspan='4', style='color:red;') Données
```

devient

```
<td colspan='4' style='color:red;'>Données</td>
```



# Syntaxe

## Structures conditionnelles

```
if (username === 'Jacques774')  
  p This user is not allowed here!  
else  
  p Welcome #{username}!
```

# Itérations

## Boucler dans les éléments d'une liste

```
if courses.length
  ul
    each course in courses
      li #{course.title}
```

# Express.js

Pug est intégré à Express.js

On place les templates Pug dans le répertoire views d'un projet Express.js

# Express.js

On utilise la méthode `render()` sur une réponse pour invoquer l'interprétation d'un template et l'envoyer au client

```
var uneRoute = function(req, res) {  
  res.render('accueil');  
};
```

# Express.js

La méthode `render()` est configurée pour chercher les templates dans le répertoire `views` directement

Donc, le chemin à lui passer en paramètre doit être relatif au répertoire `views`

On peut également omettre l'extension du fichier

# Express.js

On peut également passer un objet Javascript au template, en lui donnant comme deuxième paramètre

```
var objet = {nom: 'Berger', prenom: 'Jacques'};

var uneRoute = function(req, res) {
  res.render('accueil', objet);
};
```

# XML

Pug peut également servir à générer du XML

Dans la template, il suffit d'utiliser un doctype différent :

```
doctype xml
```

# XML

Express.js s'attend à ce que Pug produise du HTML, il faut donc manuellement changer le MIME type de la réponse pour que ça fonctionne correctement

```
res.header("Content-Type", "application/xml");
```



# Fonctionnalités

Pug offre plusieurs plus de fonctionnalités très intéressantes et utiles

Allez lire la documentation du langage

# Conclusion

Pug permet de rédiger du HTML qui respecte les standards du web en simplifiant la syntaxe

En retirant la lourdeur syntaxique du HTML, on obtient une meilleure lisibilité du code, ce qui favorise la maintenance à long terme

# Plus loin...

Jade

<https://pugjs.org>