



# Université IBM i

10 et 11 mai 2016 – IBM Client Center de Bois-Colombes

## S6 – Développer en open-source avec python et node.js sous IBM i

*Mardi 10 mai – 14h00-15h30*

Nathanaël BONNET – Gaia



# Gaia

- Conseil et formation IBM i (AS/400) depuis 1995
  - Inter et intra entreprise
- Base de connaissance en ligne
  - <http://know400.gaia.fr>
- Organisateur des matinées 400 iday
  - 24 mai à Lyon
  - 31 mai à Paris
  - <http://www.gaia.fr/400iday-2>
- Contact
  - [contact@gaiainfo.com](mailto:contact@gaiainfo.com)
  - <http://www.gaia.fr>
  - <http://twitter.com/GaiaFrance>

# L'IBM i et l'Open Source

- Les technologies Open Source sont présentes depuis longtemps dans l'environnement IBM i
  - Principalement via PASE
- Certaines fonctionnalités de l'OS sont Open Source, ou s'appuient partiellement sur des produits Open Source pakagés pour l'IBM i
  - OpenSSL/SSH
  - Serveur Apache HTTP
  - IWS avec wsdl2ws, Axis2 (fondation Apache également)
  - Kerberos (MIT)
  - ...
- Également des outils clients
  - RDi, Data Studio ...

# L'IBM i et l'Open Source

- IBM participe par ailleurs activement à plusieurs projets Open
  - OpenPower
  - Connecteur IBM i avec JT Open, version Open Source de jt400
  - Linux : 6<sup>ème</sup> entreprise contributrice au noyau Linux
    - Supporté sur les Power Systems

*ZDNet.fr > News > IBM va investir un milliard de dollars dans Linux et l'open source pour Power Systems >*

**La nou  
dévelo  
sur de** IBM va investir un milliard de dollars dans Linux et l'open source pour Power Systems **ent les  
'aillent**

LÉLIA DE MATHAREL

**Open Source :** Le centre IBM de Montpellier va bénéficier des trois quarts de cet investissement, qui doit aider les clients de Big Blue "à tirer avantage du Big Data et du cloud computing".

# Le choix de l'Open Source

- Pourquoi développer en Open Source ?
  - De nombreux softs, outils et frameworks existants
    - A disposition gratuitement
  - Pérennité
    - Vous avez les sources
  - Réactivité par rapport aux disponibilités des versions de softs
    - Besoin d'intégration d'outils existants, cycles de vie plus courts
  - Permet de faire tourner en environnement IBM i vos développements Open Source existants
    - Profitez des facilités, de la stabilité et de la puissance de l'IBM i
  - Facilité de trouver des développeurs dans ces technologies
    - Des compétences chez les jeunes diplômés
  - Des communautés actives d'utilisateurs
  - Exemples
    - WordPress (25% des sites au 11/2015)
    - Thelia, Sylius : commerce électronique
    - OpenERP, Compiere, Adempiere, Neogia, OpenBravo, ERP5, OfBiz ...

# PASE

- Le support de l'Open Source est actuellement assuré par PASE
  - Portable Application Solution Environment
    - **5770SS1 option 33**
      - Environnement d'exécution AIX dans l'IBM i (exécute des binaires AIX)
    - Ne pas confondre avec Qshell
      - **5770SS1 option 30**
        - Interpréteur de commande shell en environnement IBM i
    - Différence notable
      - CCSID (géré par QSH alors que PASE considère qu'il est ANSI)
- Par exemple, dans PASE
  - La pile TCP/IP
  - La JVM (pour Power Systems)
  - Zend Server
  - ...

# Logiciels disponibles

- Distribution par IBM via produit

**5733 OPS Open Source**

- Options

\***BASE SAMBA**

**1 Node.js v1**

**2 Python 3**

**3 GCC et chroot**

**4 Python 2 (à venir)**

**5 Node.js v4 (à venir)**

**6 Git (à venir)**

**7 Tools (à venir ??)**

**8 Orion (à venir)**

# XMLService

# XMLService - Principe

- XMLService
  - Produit RPG Open Source (cf YIPs)
  - Permet l'utilisation de ressources IBM i (programme, commande, SQL ...) via un échange de messages XML
    - Stateless ou Statefull
  - Utilisable
    - Via procédures stockées DB2
    - Via HTTP / CGI
  - Utilisé dans la plupart de Toolkits des nouveaux langages Open Source disponibles (PHP, Node.js, Python, Power Ruby)
  - Peut être utilisé par n'importe quel langage sachant communiquer via HTTP ou DB2 (.Net par exemple, mais également RPG...)
- En synthèse, c'est un outil qui automatise ce que nous utilisons depuis longtemps : procédure cataloguée DB2 et CGI

# XMLService – Exemple SQL

- Via un formulaire HTTP

```
<form name="input" action="/cgi-bin/xmlcgi.pgm" method="post">
<input type="hidden" name="db2" value="*LOCAL">
<input type="hidden" name="uid" value="NB">
<input type="hidden" name="pwd" value="MonMotDePasse">
<input type="hidden" name="ipc" value="*NA">
<input type="hidden" name="ctl" value="*here">
<input type="hidden" name="xmlin" value="<?xml version='1.0'?>
<sql>
  <script>
    <options options='myoptions' naming='sql' sqllib='nb_univ_16' />
    <connect conn='maconn' options='myoptions' />
    <query conn='maconn' stmt='lst_oper'>select * from dg_oper where id_cpt = 1</query>
    <fetch stmt='lst_oper' block='all' desc='on' />
    <free conn='maconn' />
  </script>
</sql>">
<input type="hidden" name="xmLout" value="32768">
<input type="submit" value="Submit" />
</form>
```

# XMLService – Exemple SQL

- Retourne

```
<?xml version='1.0'?>
<sql>
<options options='myoptions' naming='sql' sqllib='nb_univ_16' ><success>+++ success myoptions</success>
</options>
<connect conn='maconn' options='myoptions' ><success>+++ success maconn</success>
</connect>
<query conn='maconn' stmt='lst_oper'>
<success><![CDATA[+++ success select * from dg_oper where id_cpt = 1]]></success>
</query>
<fetch stmt='lst_oper' block='all' desc='on'>
<row><data desc='ID'>1</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>2</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>3</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>4</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>5</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>6</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>7</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<row><data desc='ID'>8</data><data desc='ID_CLI'>1</data><data desc='ID_CPT'>1</data><data desc='LIBELLE'>
<success>+++ success lst_oper</success>
</fetch>
<free conn='maconn'><success>+++ success maconn</success>
</free>
</sql>
```

# XMLService – Exemple ILE

## ■ Via un formulaire HTTP

```
<input type="hidden" name="xmlin" value=<?xml version='1.0'?>
<pgm name='WS_SRV' lib='NB_UNIV_16' func='GETOPERATIONARRAY' mode='ile'>
  <parm comment='id' by='ref' io='in'>
    <data type='10i0'>1</data>
  </parm>
  <parm comment='maxResult' by='ref' io='in'>
    <data type='5u0'>20</data>
  </parm>
  <parm comment='operationArray' io='out' by='ref'>
    <ds dim='20'>
      <data comment='id' type='10i0'></data>
      <data comment='id_cli' type='10i0'></data>
      <data comment='id_cpt' type='10i0'></data>
      <data comment='libelle' type='25a'></data>
      <data comment='montant' type='10p2'></data>
      <data comment='maj' type='26a'></data>
    </ds>
  </parm>
  <parm comment='p_oprArray_Length' by='ref' io='out'>
    <data comment='p_oprArray_Length' type='10i0'></data>
  </parm>
  <parm io='out' by='ref'>
    <data comment='rc' type='3A'></data>
  </parm>
  <return><data comment='return' type='10i0'></data></return>
</pgm>">
```

# XMLService – Exemple ILE

## ■ Retourne

```
<?xml version='1.0'?>
<pgm name='WS_SRV' lib='NB_UNIV_16' func='GETOPERATIONARRAY' mode='ile'>
    <parm comment='operationArray' io='out' by='ref'>
        <ds dim='20'>
            <data comment='id' type='10i0'>1</data>
            <data comment='id_cli' type='10i0'>1</data>
            <data comment='id_cpt' type='10i0'>1</data>
            <data comment='Libelle' type='25a'>Salaire</data>
            <data comment='montant' type='10p2'>2763,25</data>
            <data comment='maj' type='26a'>2015-04-20-10.49.39.720153</data>
        </ds>
        <ds dim='20'>..</ds>
        <ds dim='20'>..</ds>
        ...
        <ds dim='20'>..</ds>
    </parm>
    <parm comment='p_oprArray_Length' by='ref' io='out'>
        <data comment='p_oprArray_Length' type='10i0'>8</data>
    </parm>
    <parm io='out' by='ref'>
        <data comment='rc' type='3A'></data>
    </parm>
    <return>
        <data comment='return' type='10i0'>1</data>
    </return>
    <success><![CDATA[+++ success NB_UNIV_16 WS_SRV GETOPERATIONARRAY ]]></success>
</pgm>
```

# XMLService – Exemple procédure cataloguée

```
call iPLUG4K( '*NA',
               '*here',
               '<?xml version=''1.0'''?><pgm name=''WS_SRV''
lib=''NB_UNIV_16'' func=''GETOPERATIONARRAY'' mode=''ile'''><parm
comment=''id'' by=''ref'' io=''in'''><data
type=''10i0'''>1</data></parm><parm comment=''maxResult'' by=''ref'''
io=''in'''><data type=''5u0'''>20</data></parm><parm
comment=''operationArray'' io=''out'' by=''ref'''><ds dim=''20'''><data
comment=''id'' type=''10i0'''></data><data comment=''id_cli'''
type=''10i0'''></data><data comment=''id_cpt'''
type=''10i0'''></data><data comment=''libelle'''
type=''25a'''></data><data comment=''montant'''
type=''10p2'''></data><data comment=''maj'''
type=''26a'''></data></ds></parm><parm comment=''p_oprArray_length'''
by=''ref'' io=''out'''><data comment=''p_oprArray_length'''
type=''10i0'''></data></parm><parm io=''out'' by=''ref'''><data
comment=''rc'' type=''3A'''></data></parm><return><data
comment=''return'' type=''10i0'''></data></return></pgm>',
      '' ) ;
```

# XMLService – Synthèse

## ■ Avantages et inconvénients

- Ne nécessite pas d'adaptation des programmes à appeler
- Ne nécessite pas de code supplémentaire pour exposer un programme
- Ne sait pas utiliser PCML
  - Possibilité d'automatiser la construction du template de flux XML à partir du PCML par programmation ?
  - Permet surtout de contourner les limites de PCML
- Capacités statefull/stateless
- Capacités HTTP/DB2
- Performances ?

# Ruby

# Ruby on Rails

- Ruby
  - Langage de programmation libre, objet et interprété
    - Standardisé ISO en 2012 (ISO 30170:2012)
  - Philosophie
    - Toute donnée est un objet, y compris les types et constantes
    - Toute fonction est une méthode
    - Toute variable est une référence à un objet
  - Syntaxe
    - Simple, inspirée par Eiffel et Ada
- Rails
  - Framework libre, écrit pour Ruby
  - Basé sur MVC pour des applications web

# Power Ruby

- Power Ruby est une implémentation de Ruby on Rails pour IBM i !
- Elle permet de tirer partie
  - De DB2
    - Base de données robuste et performante
  - Du framework Rails
  - Et de toute la communauté et des ressources disponibles sur l'IBM i
    - Avec la stabilité et la capacité de gestion de charges importantes

# Prérequis

- Vous devez disposer des produits suivants

**SC1 / \*BASE** IBM Portable Utilities for i

**SC1 / 1** OpenSSH, OpenSSL, zlib

**SS1 / 33** PASE

**DG1 / \*BASE** IBM HTTP Server for i

# Installation

- Télécharger le produit ici
  - <http://powerruby.com/download/>
    - Ou directement :  
[http://powerruby.com/releases/power\\_ruby\\_current.tar.bz2](http://powerruby.com/releases/power_ruby_current.tar.bz2)
  - Les procédures d'installation sont susceptibles de changer au fil des versions
- Installation
  - La procédure d'installation/mise à jour
    - <https://powerruby.worketc.com/kb?id=11>
  - On ne détail pas la 1<sup>ère</sup> partie classique
    - Création des SAVF
    - Restauration du programme sous licence **1PRUBY1**

# Installation

- Attention, en version IBM i 7.2
  - La version d'**OpenSSL** n'est pas compatible avec le runtime Power Ruby
  - Il faut installer le package suivant
    - [http://powerruby.com/releases/power\\_ruby\\_v1.0.8\\_optimized\\_NO\\_power5.tar.bz2](http://powerruby.com/releases/power_ruby_v1.0.8_optimized_NO_power5.tar.bz2)
    - Qui contient une version intégrée d'**OpenSSL** et devrait devenir la release

# Installation

- Les options suivantes sont installées

**1PRUBY1** \*BASE Core functionnality

**1PRUBY1** 2 Ruby 2.0.x + Rails 4.0.x

- D'autres options existent

**1PRUBY1** 1 Ruby 1.9.3 + Rails 3.2.14

— Non distribué

**1PRUBY1** 3 Ruby 2.1.x + Rails 4.0.x

— En béta (2015-06-05)

# Installation

## ■ Vérifier l'installation

### **ADDLIB LIB(POWER\_RUBY)**

- Contient les exécutables natifs IBM i

### **SETPOWRBY VRM(\*V20)**

- Créer les liens symboliques dans l'IFS pour la version 2.0.x de Ruby

**ruby -v et rails -v** dans l'environnement PASE

**CALL QP2TERM**

**ruby -v**

```
> ruby -v
  ruby 2.0.0p643 (2015-02-25 revision 49749) [powerpc-aix6.1.0.0]
#
> rails -v
  Rails 4.0.13
```

# Éléments installés

## ■ Bibliothèque **POWER\_RUBY**

- Liste des commandes

**RAILSNEW** (Créer l'application)

**RAILSSVR** (Arrêter ou démarrer le serveur Rails)

**RMVPOWRBY** (Détruire les liens symboliques)

**SETPOWRBY** (Créer les liens symboliques pour une version de Ruby)

- Création des liens de /QOpenSys/usr/bin vers /PowerRuby/prV2R0/bin

## ■ Répertoire **/PowerRuby**

- Contient

**/PowerRuby/oss**

- Logiciels open source

**/PowerRuby/prV2R0**

- Ruby 2.0.x et Rails 4.0.x

# Créer une nouvelle application

- Suivant les informations indiquées
  - <https://powerruby.worketc.com/kb?id=22>

- Création d'un serveur

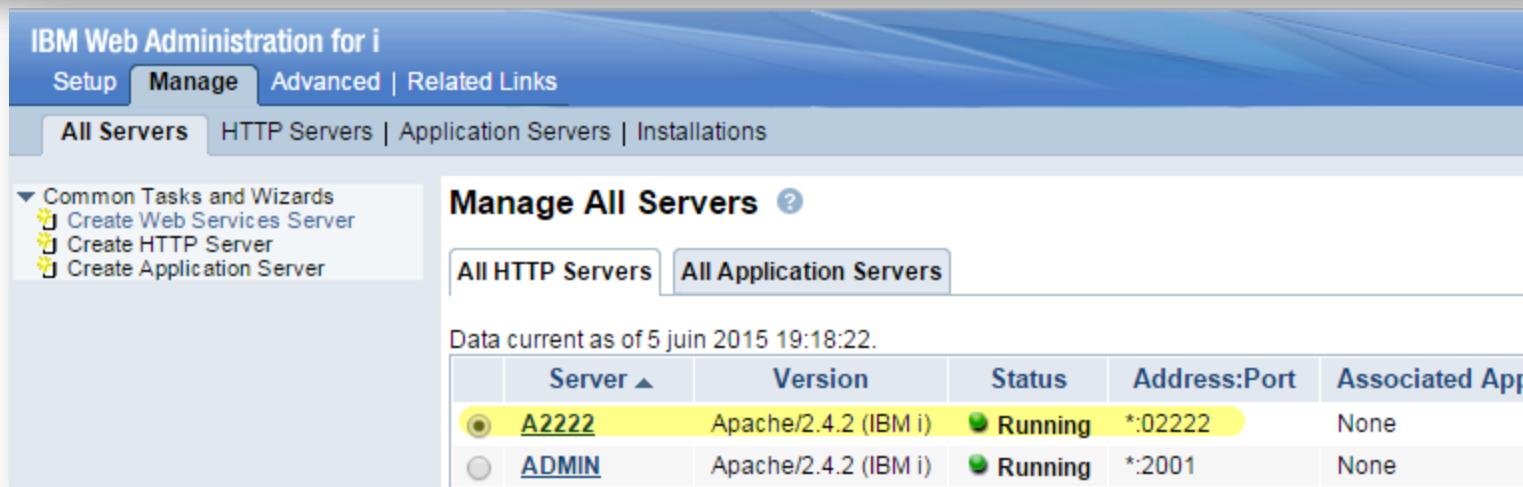
**RAILSNEW VRM(\*V20) PLACE(\*WWW) RAILSAPP(A2222)  
DBUSR(A2222) DBPASS(A2222) DBROOT(A2222) HTTPSrv(A2222)  
EXTPATH('') EXTPORT(2222) INTPATH('') INTPORT(2202)**

- Crée
  - Une instance de serveur Apache A2222 écoutant sur le port 2222
  - Une application RubyOnRails (répertoire `htdocs/A2222`)
  - Les collections A2222\_D (Dév), A2222\_T (Test) et A2222\_P (Prod)
  - Un profil utilisateur A2222 (mot de passe A2222)
  - Démarre l'instance Apache
  - Démarre le serveur Rails sur le port interne 2202

# Créer une nouvelle application

- Exécution de la commande

```
> RAILSNEW VRM (*V20) RAILSAPP (A2222) DBUSR (A2222) DBPASS (A2222) DBROOT (A2222)
  2) HTTPSRV (A2222) EXTPATH('') EXPORT (2222) INTPATH('') INTPORT (2202) PLA
CE (*WWW)
Rails new job completed
Bundle install job completed
```



The screenshot shows the 'Manage All Servers' page in the IBM Web Administration for i interface. The page displays a table of servers with the following data:

	Server	Version	Status	Address:Port	Associated App
●	A2222	Apache/2.4.2 (IBM i)	Running	*:02222	None
●	ADMIN	Apache/2.4.2 (IBM i)	Running	*:2001	None

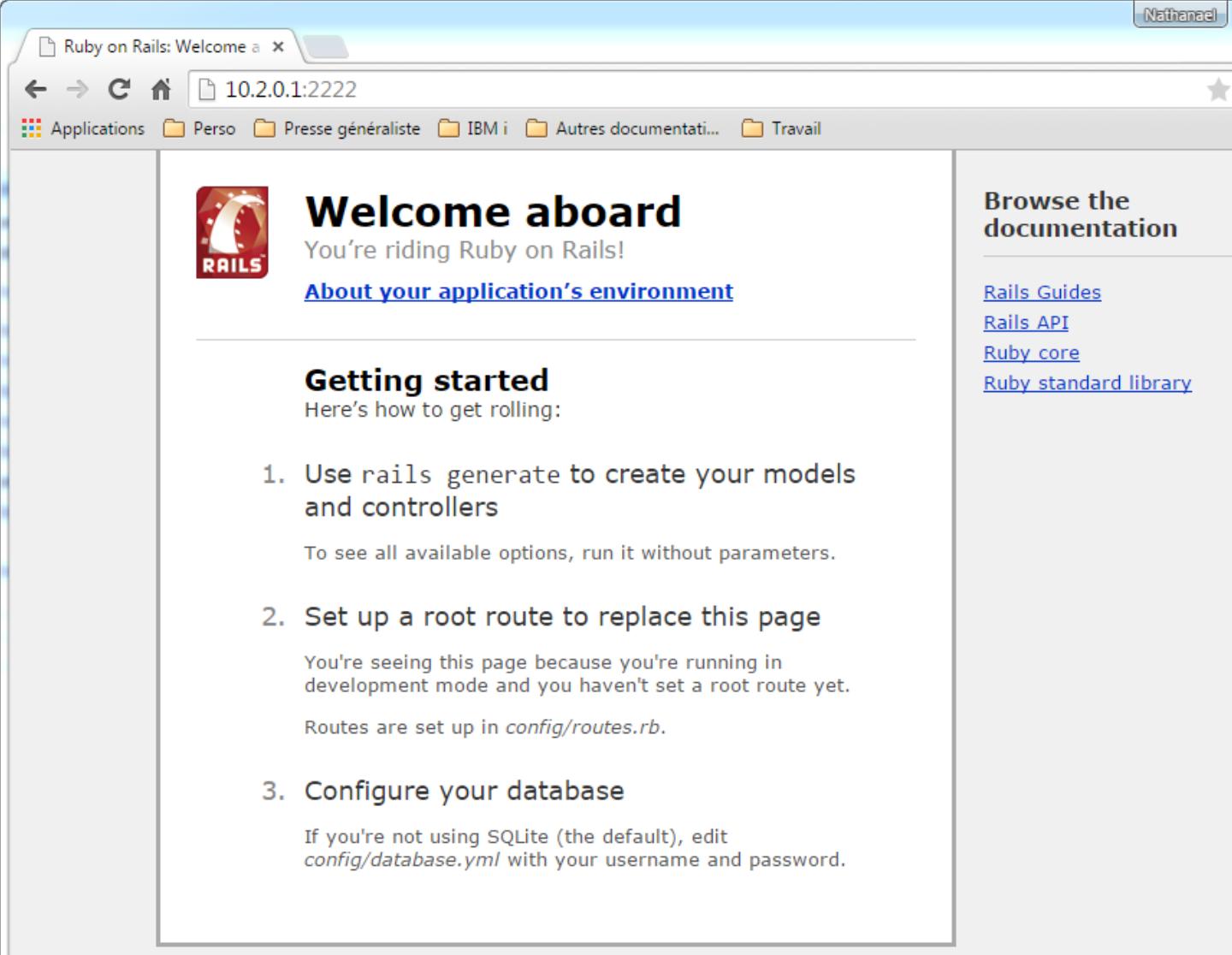
- Le serveur créé a pour rôle de réécrire les URL et de donner la main au serveur Rails interne sur le port 2202

# Créer une nouvelle application

## ■ Configuration du serveur Apache

```
##  
## PowerRuby  
##  
LoadModule proxy_module /QSYS.LIB/QHTTPSRV.LIB/QZSRCORE.SRVM  
LoadModule proxy_ftp_module /QSYS.LIB/QHTTPSRV.LIB/QZSRCORE.SRVM  
LoadModule proxy_http_module /QSYS.LIB/QHTTPSRV.LIB/QZSRCORE.SRVM  
LoadModule proxy_connect_module /QSYS.LIB/QHTTPSRV.LIB/QZSRCORE.SRVM  
Listen *:02222  
DocumentRoot /www/A2222/htdocs/A2222/public  
RewriteEngine On  
RewriteCond %{REQUEST_URI} !^/assets/.* [OR]  
RewriteCond %{DOCUMENT_ROOT}/ %{REQUEST_FILENAME} !-f  
RewriteRule ^(.*) http://127.0.0.1:02202/$1 [P]  
ProxyPassReverse / http://127.0.0.1:02202/
```

# Créer une nouvelle application



The screenshot shows a web browser window titled "Ruby on Rails: Welcome aboard". The address bar displays "10.2.0.1:2222". The navigation bar includes links for Applications, Perso, Presse généraliste, IBM i, Autres documentati..., and Travail. A user profile "Nathanael" is visible in the top right corner.

**Welcome aboard**  
You're riding Ruby on Rails!  
[About your application's environment](#)

---

**Getting started**  
Here's how to get rolling:

1. Use rails generate to create your models and controllers  
To see all available options, run it without parameters.
2. Set up a root route to replace this page  
You're seeing this page because you're running in development mode and you haven't set a root route yet.  
Routes are set up in `config/routes.rb`.
3. Configure your database  
If you're not using SQLite (the default), edit `config/database.yml` with your username and password.

**Browse the documentation**

---

[Rails Guides](#)  
[Rails API](#)  
[Ruby core](#)  
[Ruby standard library](#)

# Générer un modèle/contrôleur/vue

- Ruby dispose de commandes permettant de générer automatiquement des éléments
  - BD, modèle, contrôleur et vue
  - Ces éléments sont bien sur personnalisables par la suite
- Dans PASE
  - Générer le modèle
    - rails generate model Employe num:integer nom:string prenom:string sexe:string tel:string direction:string salaire:integer bonus:integer
  - Générer la BD
    - rake db:migrate
  - Générer le contrôleur
    - rails g scaffold\_controller employe num:integer, nom:string, prenom:string, sexe:string, tel:string, direction:string, salaire:integer, bonus:integer

# Générer un modèle/contrôleur/vue

- Après alimentation

## Listing employes

Num	Nom	Prenom	Sexe	Tel	Direction	Salaire	Bonus	
10	HAAS	CHRISTINE	F	3978 A00	52750	1000	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
20	THOMPSON	MICHAEL	M	3476 B01	41250	800	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
30	KWAN	SALLY	F	4738 C01	38250	800	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
50	GEYER	JOHN	M	6789 E01	40175	800	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
60	STERN	IRVING	M	6423 D11	32250	500	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
70	PULASKI	EVA	F	7831 D21	36170	700	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
90	HENDERSON	EILEEN	F	5498 E11	29750	600	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
100	SPENSER	THEODORE	M	0972 E21	26150	500	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	
110	LUCCHESI	VINCENZO	M	3490 A00	46500	900	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>	

## Editing employe

Num

Nom

Prenom

Sexe

Tel

Direction

Salaire

Bonus

[Update Employe](#)

[Show](#) | [Back](#)

# Arrêt et démarrage

## ■ Démarrer

**STRTCPSVR SERVER(\*HTTP) HTTPSVR(A2222)**

**RAILSSVR ACTION(\*START) APP('/www/A2222/htdocs/A2222')**  
**VRM(\*V20) PORT(2202)**

## ■ Arrêter

**RAILSSVR ACTION(\*END) APP('/www/A2222/htdocs/A2222')**

**ENDTCPSVR SERVER(\*HTTP) HTTPSVR(A2222)**

# QCCSID

- Comme toujours, cela fonctionne mal lorsque votre valeur système **QCCSID** est à **65535**

**DSPSYSVAL SYSVAL(QCCSID)**

- Dans ce cas

**CHGJOB CCSID(37)**

# Utilisation du toolkit pour XMLService

```
# File: getOperationArray.rb
require "./auth"

getOpArr = XMLService::I_SRVPGM.new("WS_SRV", "GETOPERATIONARRAY",
                                     $toolkit_test_lib, {'error'=>'on'})

getOpArr << XMLService::I_Int32.new('id', 1)
getOpArr << XMLService::I_Uint16.new('maxResult', 20)
getOpArr << XMLService::I_DS.new("operationArray", 20, 'p_oprArray_length',
                                 [
                                   XMLService::I_Int32.new('ID', 0),
                                   XMLService::I_Int32.new('ID_CLI', 0),
                                   XMLService::I_Int32.new('ID_CPT', 0),
                                   XMLService::I_Char.new('LIBELLE', 25, ''),
                                   XMLService::I_PackedDecimal.new('MONTANT', 10, 2, 0.0),
                                   XMLService::I_Char.new('MAJ', 26, '0001-01-01-00.00.00.000000')
                                 ])
getOpArr << XMLService::I_Int32.new('p_oprArray_length', 0, 'p_oprArray_length')
getOpArr << XMLService::I_Char.new('rc', 3, ' ')
getOpArr.setReturn("integer", XMLService::I_Int32.new('id', 0))

# Appel
getOpArr.call
```

# Utilisation du toolkit pour XMLService

```
# Erreur ?
rc = getOpArr.xmlservice_error
if rc
    puts getOpArr.dump_error
end

# output
puts " p_oprArray_length....#{getOpArr.response.p_oprArray_length}\n"
i = 0
getOpArr.response.operationArray.each do |operationArray|
    i += 1
    puts "operationArray[#{i}]"
    puts " ID...#{operationArray.ID}"
    puts " ID_CLI...#{operationArray.ID_CLI}"
    puts " ID_CPT...#{operationArray.ID_CPT}"
    puts " LIBELLE...#{operationArray.LIBELLE}"
    puts " MONTANT...#{operationArray.MONTANT}"
    puts " MAJ...#{operationArray.MAJ}"
end
```

# Python

# Python

- Python
  - Langage de programmation libre et Open Source
    - Orienté Objet et dispose d'un typage fort
    - Permet également une programmation de type fonctionnelle
    - Apparu en 1990
    - <http://www.python.org>
  - Syntaxe
    - Simple, inspirée par C, PHP
  - Extensions
    - Il est possible d'installer des « add-ons » c'est-à-dire des extensions

# Prérequis

- Vous devez disposer des produits suivants
  - SC1 / \*BASE** IBM Portable Utilities for i
  - SC1 / 1** OpenSSH, OpenSSL, zlib
  - SS1 / 33** PASE
- Comme d'habitude, les derniers Group PTF sont recommandés

# Installation

- Python est packagé dans  
**5733-OPS (Open Source for IBM i) Option 2**  
**PTF SI57008**
- Cela vous permet d'installer le runtime Python  
**/QOpenSys/QIBM/ProdData/OPS/Python3.4**
- Package à télécharger via ESS  
**F\_MULTI\_NLV\_110\_IBM\_i\_Open\_Source\_Solutions**
  - Gratuit si vous possédez IBM i 7.1 ou 7.2

# Installation

- Les extensions sont installables sous forme de PTF
  - DB2 for i connector (basé sur le projet **ibm\_db**)  
PTF SI57253
  - Toolkit for IBM i  
PTF SI57254
  - fastCGI gateway (permet l'usage du framework flipflop)  
PTF SI57255
  - lightweight web framework (permet l'utilisation de bottle.py)  
PTF SI57256
- Les extensions sont installées dans  
**/QOpenSys/QIBM/ProdData/OPS/Python-pkgs**
- Ces PTFs sont à télécharger via FixCentral
  - Facultatives à l'exécution de Python

# Installation

- Pour vérifier l'installation
  - Sous QSH ou QP2TERM

```
python3 -V
Python 3.4.2
$
```

```
> python3 -V
Python 3.4.2
$
```

# Principales commandes

- **python3 / python3.4**
  - Programme principal Python
- **pip3 / pip3.4**
  - Installation de packages
- **2to3 / 2to3-3.4**
  - Transformer du code Python 2.x en code Python 3.x
- **pydoc3 / pydoc3.4**
  - Générateur de documentation
- **pyvenv / pyvenv-3.4**
  - Fourni les environnements virtuels
- **easy\_install3 / easy\_install-3.4**
  - Compiler, installer et gérer les packages Python, y compris les extensions livrées en standard

# Installer les extensions fournies

- Il faut avoir téléchargé et appliquer les PTFs correspondantes
- Ensuite

- Installation du connecteur DB2 natif

```
easy_install3 /QOpenSys/QIBM/ProdData/OPS/Python-
pkgs/ibm_db/ibm_db-*.egg
```

- Installation du Toolkit pour IBM i

```
easy_install3 /QOpenSys/QIBM/ProdData/OPS/Python-
pkgs/itoolkit/itoolkit-*.egg
```

- Installation du support FastCGI gateway

```
easy_install3 /QOpenSys/QIBM/ProdData/OPS/Python-
pkgs/flipflop/flipflop-*.egg
```

- Installation du lightweight web framework

```
easy_install3 /QOpenSys/QIBM/ProdData/OPS/Python-
pkgs/bottle/bottle-*.egg
```

# Installer les extensions fournies

- Exemple d'installation d'une extension

```
$  
> easy_install3 /QOpenSys/QIBM/ProdData/OPS/Python-pkgs/ibm_db/ibm_db-*.egg  
Processing ibm_db-2.0.5.1-py3.4-os400-powerpc.egg  
creating /QOpenSys/QIBM/ProdData/OPS/Python3.4/lib/python3.4/site-packages/  
Extracting ibm_db-2.0.5.1-py3.4-os400-powerpc.egg to /QOpenSys/QIBM/ProdData/  
Adding ibm-db 2.0.5.1 to easy-install.pth file  
  
Installed /QOpenSys/QIBM/ProdData/OPS/Python3.4/lib/python3.4/site-packages/  
Processing dependencies for ibm-db==2.0.5.1  
Finished processing dependencies for ibm-db==2.0.5.1  
$
```

# Installer d'autres extensions

- Des packages sont disponibles dans **PyPI (Python Package Index)**
  - Modules Python Open Source
- Il est possible d'installer ces packages
  - Via la commande **pip3**
  - Nécessite une connexion internet
- Fonctionne avec
  - QSH
  - PASE (QP2TERM)
- Documentation
  - <https://docs.python.org/3/installing/>

# Installer d'autres extensions

- Exemple

```
pip3 install xlsxwriter
```

```
pip3 uninstall xlsxwriter
```

```
pip3 install xlsxwriter
Downloading/unpacking xlsxwriter
  Downloading XlsxWriter-0.7.3-py2.py3-none-any.whl (132kB): 132kB downloaded
Installing collected packages: xlsxwriter
Successfully installed xlsxwriter
Cleaning up...
#
```

# Compilation native

- Certains packages Python nécessitent de compiler du code C/C++
- Il faut donc un compilateur que **pip3** puisse utiliser **GCC**
  - Maintenant distribué via **57330PS**
  - Auparavant
    - GCC (GNU Compiler Collection)
      - gcc est un compilateur libre, très utilisé dans le monde Linux
      - Vous pouvez l'installer en suivant les indications données ici  
<http://www.youngprofessionals.com/wiki/index.php/PASE/GCC>
    - IBM XL C/C++ for AIX
      - Vous pouvez installer le compilateur IBM pour AIX
      - L'ensemble des packages n'est pas compatible

# Hello World

- Pour créer un programme Python
  - Créer le fichier sample.py sur l'IFS avec le contenu suivant

```
from bottle import route, run

port_number=9090
host_location='10.2.0.1'

@route('/')
def hello():
    return "OK ! Ca marche !"

run(host=host_location, port=port_number, debug=True)
```

# Hello World

- Tapez la commande

```
python3 sample.py
```

```
> python3 /home/NB/python/sample.py
Bottle v0.12.8 server starting up (using WSGIRefServer())...
Listening on http://10.2.0.1:9090/
Hit Ctrl-C to quit.
```

- Dans un navigateur web

- L'IP et le port dépendent de ce que vous avez indiqué dans le fichier **sample.py**



# Hello World

- Cela génère les jobs
  - QZSHSH
    - Interpréteur QSH
  - QP0ZSPWR
    - Process Python : serveur web

QINTER	QSYS	SBS	0,0	DEQW
QPADEV0004	QSECOFR	INT	0,0	CMD-WRKACTJOB RUN
QPOZSPWT	QSECOFR	BCI	0,2	PGM-python3 SELW
QZSHSH	QSECOFR	BCI	0,0	PGM-QZSHSH EVTW

# Page statique

- Servir une page web statique au travers d'un programme Python
  - Créer une page web statique de votre choix  
`sample.html`
  - Créer un fichier sample2.py contenant le code Python

```
from bottle import static_file, route, run
port_number=9090
host_location='10.2.0.1'
file_name='sample.html'

@route('/sample')
def sample():
    return static_file(file_name, root='/home/NB/python')

run(host=host_location, port=port_number, debug=True)
```

# Page statique

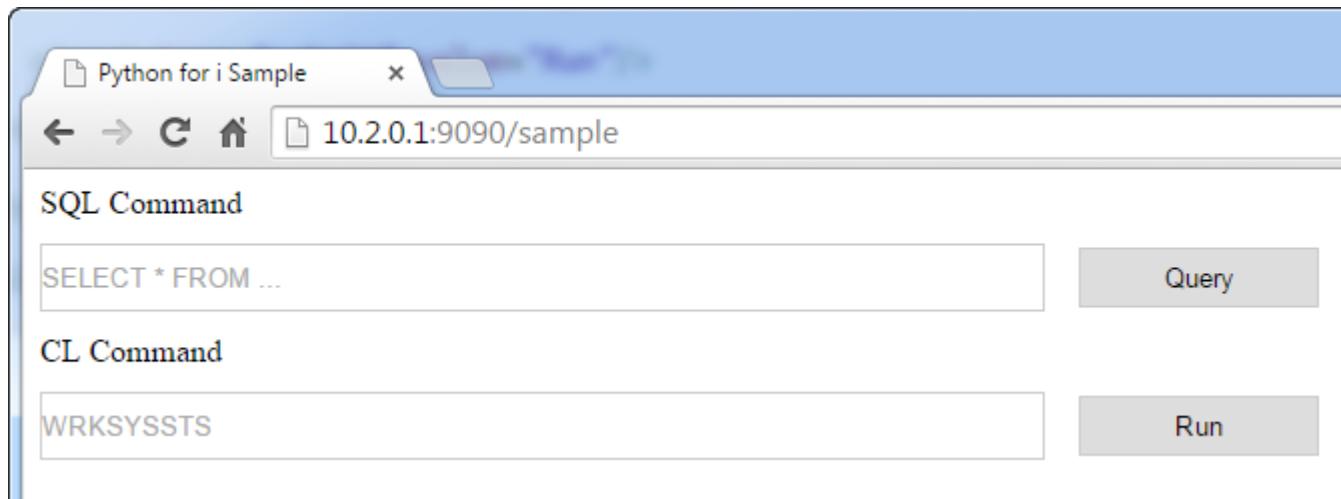
- Tapez la commande

```
python3 sample2.py
```

```
> python3 sample2.py
  Bottle v0.12.8 server starting up (using WSGIRefServer())...
  Listening on http://10.2.0.1:9090/
  Hit Ctrl-C to quit.
```

- Dans un navigateur web

- Le résultat dépend du contenu de votre fichier HTML



The screenshot shows a web browser window titled "Python for i Sample". The address bar contains the URL "10.2.0.1:9090/sample". The page content is a static HTML page with two main sections: "SQL Command" and "CL Command". The "SQL Command" section contains a text input field with the placeholder "SELECT \* FROM ...". To its right is a "Query" button. The "CL Command" section contains a text input field with the placeholder "WRKSYSSTS". To its right is a "Run" button. The overall layout is clean and functional, typical of a web-based management interface.

# db2

```
@get('/db2')
def query_ibm_db():

    xml_result = "<?xml version=\"1.0\" encoding=\"UTF-8\"?><operations>"

    conn = ibm_db.connect(db_name, username, password)

    if conn:
        result_set = ibm_db.exec_immediate(conn, "select * from nb_univ_16.dg_oper")
        num_columns = ibm_db.num_fields(result_set)

        # insert results into XML value
        while (ibm_db.fetch_row(result_set)):
            xml_result += "<operation>"
            for num in range(0, num_columns):
                result = ibm_db.result(result_set, num)
                col = ibm_db.field_name(result_set, num)
                xml_result += "<" + col + ">" + str(result) + "</" + col + ">"
            xml_result += "</operation>"

        xml_result += "</operations>"
        ibm_db.close(conn)
        response.content_type = 'text/xml; charset=utf-8'
        return xml_result

    else:
        return "<p>Connection Failed.</p>"
```

# ILE

```
@get('/ile')
def cmd_toolkit():

    itransport = iLibCall()
    itool = iToolKit()
    itool.add(iCmd('chcurlib', 'CHGURLIB CURLIB(NB_UNIV_16)'))
    itool.add(
        iSrvPgm('getOperationArray', 'WS_SRV', 'GETOPERATIONARRAY')
            .addParm(iData('id', '10i0', '1'))
            .addParm(iData('maxResult', '5u0', '20'))
            .addParm(
                iDS('operationArray', {'dim': '20'})
                    .addData(iData('ID', '10i0', '0'))
                    .addData(iData('ID_CLI', '10i0', '0'))
                    .addData(iData('ID_CPT', '10i0', '0'))
                    .addData(iData('LIBELLE', '25a', ''))
                    .addData(iData('MONTANT', '10p2', '0.0'))
                    .addData(iData('MAJ', '26a', '0001-01-01-00.00.000000'))
            )
            .addParm(iData('p_oprArray_Length', '10i0', '0'))
            .addParm(iData('rc', '3a', ''))
    )
```

# ILE

```
# xmbservice
itool.call(itransport)

# output
chgcurlib = itool.dict_out('chgcurlib')
if 'success' in chgcurlib:
    print (chgcurlib['success'])
else:
    print (chgcurlib['error'])
exit()
```

# Node.js

# Prérequis

- Vous devez également disposer des produits suivants
  - **5330SS1 option 33**  
**Portable Application Solutions Environment (PASE)**
  - **5770SC1 option 1**  
**OpenSSH, OpenSSL, zlib functions, IBM Portable Utilities for i**
  - **5770DG1\*BASE**  
**IBM HTTP Server for i**
- De plus, si vous désirez installer des extensions de Node.js, vous devrez utiliser les produits Open Source suivants, non officiellement supportés par IBM
  - **Python** - requis par npm pour l'installation d'extensions tierces
  - **GCC** – compilateur C/C++ pour les extensions
  - **Git** - Pour utilisation de Github

# Obtenir Node.js

- Téléchargeable depuis ESS (Entitled Systems Support) sous le nom  
**F\_MULTI\_NLV\_110\_IBM\_i\_Open\_Source\_Solutions**
- Vous devez alors restaurer le produit **57330PS**  
**\*BASE**  
**Option 1 – Node.js**
- Vous devez également installer le dernier niveau pour le **Group PTF HTTP**
  - 7.1 : SF99368 – niveau >= 31
  - 7.2 : SF99713 – niveau >= 5
    - + PTF 57330PS: SI55499
- Le répertoire suivant est créé  
**/QOpenSys/QIBM/ProdData/Node/**

# Valider l'installation

- Via QSH ou PASE

- node -v**

- Version de **Node.js**

- npm -v**

- Version de **npm**

```
$  
> node -v  
v0.10.29  
$  
> npm -v  
1.4.14  
$
```

# Environnement

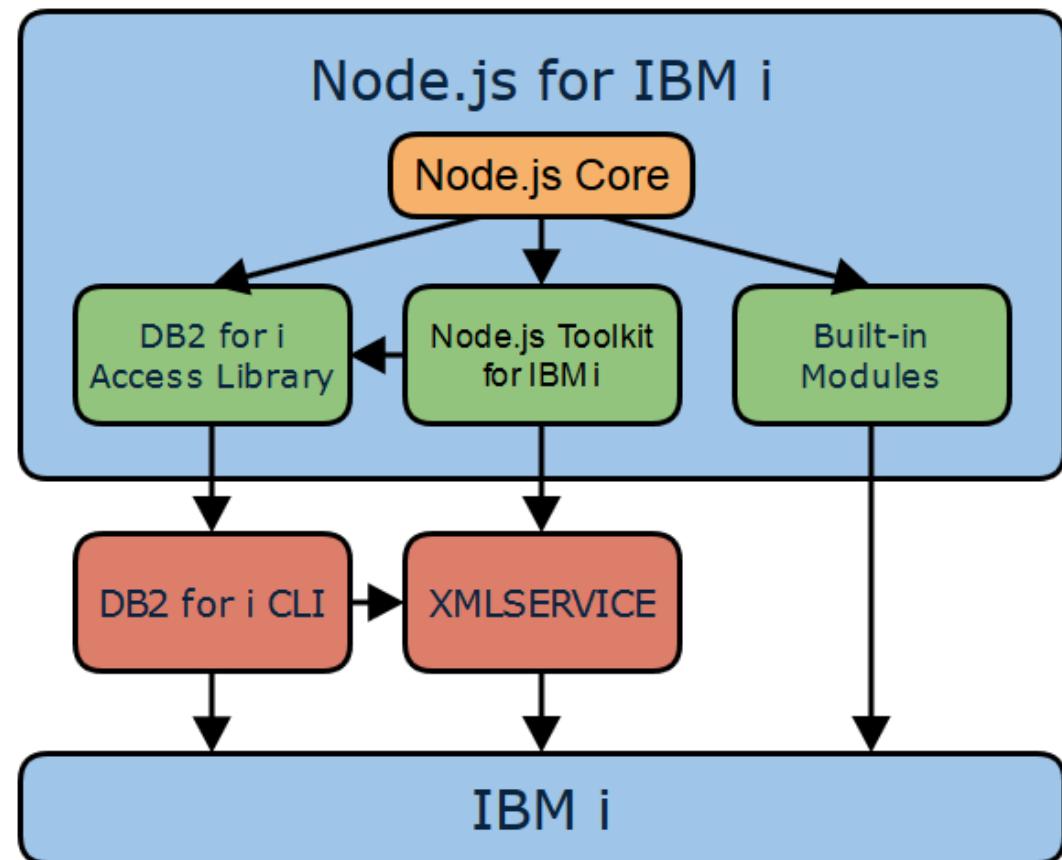
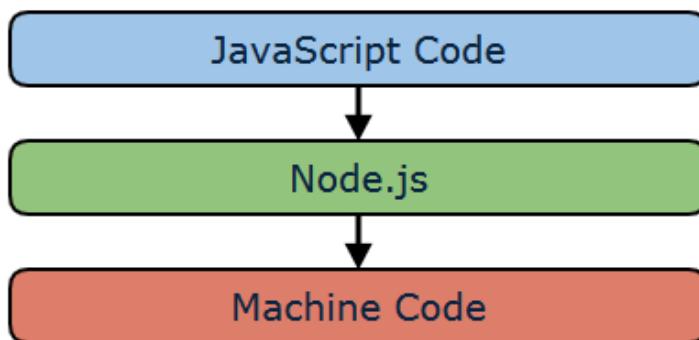
- Si vous avez des erreurs **QSH**  
`errno 5 : unexpected exception in a child shell`
- Cela provient d'une limitation de **QSH** en nombre de threads
  - Vous devez créer ou modifier la variable d'environnement  
**QIBM\_MULTI\_THREADED**  
`ADDENVVAR ENVVAR(QIBM_MULTI_THREADED) VALUE(Y)`
  - Pour l'indiquer au niveau système  
`ADDENVAR ... LEVEL(*SYS)`

# Usage

- Node.js est un projet Open Source basé sur le moteur Javascript V8 de Chrome (Google)
  - C'est un moteur de javascript côté serveur !
    - Le Javascript a été créé en 1995, c'est maintenant un langage mature
    - Permet de générer du HTML
      - Comme PHP, Java, Ruby ...
- Liens
  - <http://fr.wikipedia.org/wiki/Node.js>
  - <https://nodejs.org/>
  - <http://openclassrooms.com/>

# Principe de fonctionnement

- Node.js
  - Transforme du JavaScript en code machine exécutable



# Packaging

- 3 éléments sont fournis
  - **Node.js**
    - Le moteur d'exécution
    - Il fonctionne exclusivement dans PASE
  - **Extension : DB2 for i Node.js**
    - Permet à des scripts Node.js d'accéder à DB2
    - L'extension utilise le CLI au final
  - **Extension : IBM i ILE object toolkit**
    - Ce toolkit s'appuie sur XMLService afin de permettre l'accès à de nombreux objets du système : programmes et programmes de service, data areas, data queues ...

# Hello World

- Dans l'IFS, créer un fichier « helloworld.js » avec le contenu suivant :

```
var http = require('http');
var ip = "10.2.0.1";
var port = 8085;
var webserver = http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World\n');
});
webserver.listen(port, ip);
console.log('Server running at http://' + ip + ':' + port);
```

- Vous pouvez utiliser EDTF ou des outils Windows d'édition

# Hello World

- Ce script effectue les actions suivantes
  - Création d'une instance de serveur web
    - Sur l'IP 10.2.0.1 et le port 8085
  - Définition d'une fonction de traitement d'une requête HTTP
    - Ecrire « 200 » dans l'entête HTTP, ce qui est le code OK pour HTTP
    - Ecrire « Hello World » sur le flux de données HTTP
  - Démarrer l'écoute sur l'IP/port
  - Afficher un message dans la console QSH

# Hello World

- Dans QSH, exécuter votre script

```
qsh
```

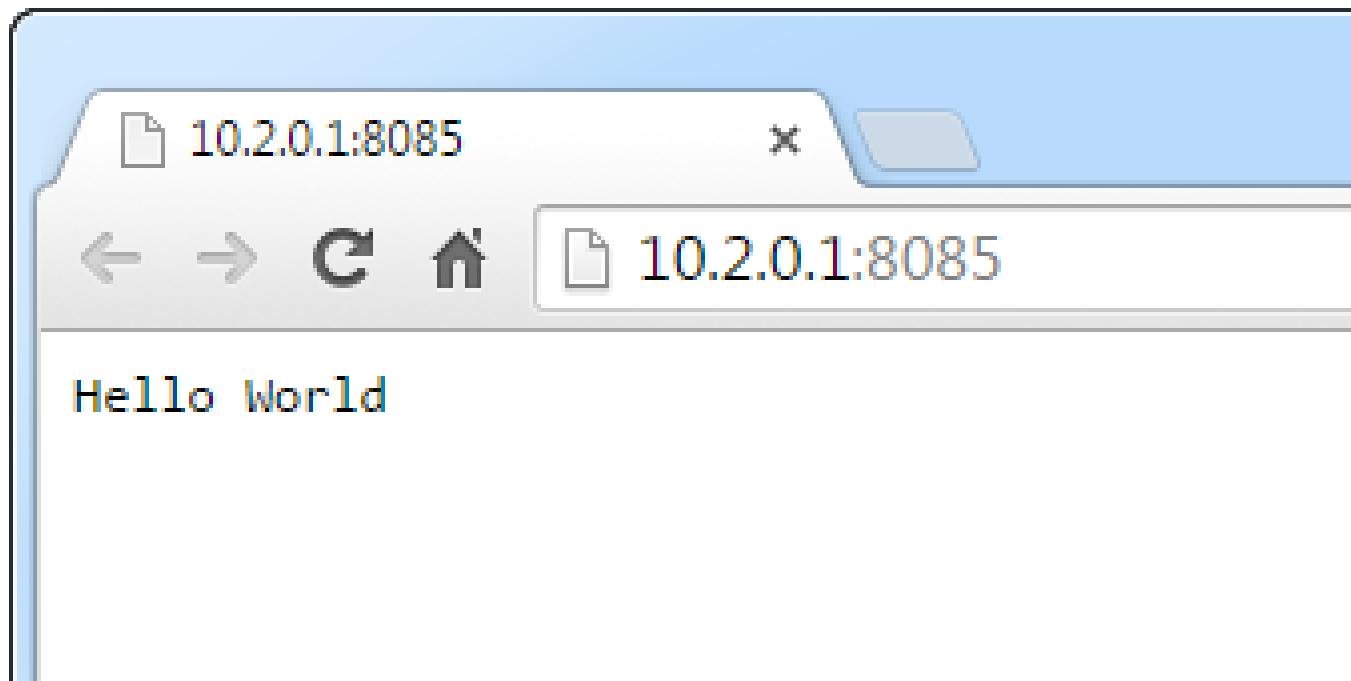
```
node /home/nb/nodejs/helloworld.js
```

- L'exécution du script lance le serveur, vous devez obtenir

```
> node /home/nb/nodejs/helloworld.js
Server running at http://10.2.0.1:8085
```

# Hello World

- Dans un navigateur, ouvrez l'URL
  - <http://10.2.0.1:8085>



# Démarrage / arrêt

- Lancement par QSH

SBMJOB

CMD(node /home/nb/nodejs/helloworld.js)

- Arrêt manuel (faire un CL...)

— WRKACTJOB

— Le travail est celui exécutant PGM-node

—	QINTER	QSYS	SBS	0,0	DEQW
—	QPADEV0002	NB	INT	0,0	CMD-WRKACTJOB RUN
4	QPOZSPWP	NB	BCI	0,0	PGM-node SELW
—	QZSHSH	NB	BCI	0,0	PGM-QZSHSH EVTW

# Accéder à la BD

- Pour accéder à DB2, on utilise l'extension  
`DB2 for i Node.js`
- Disponible dans l'IFS  
`/QOpenSys/QIBM/ProdData/Node/os400/db2i/lib/db2`

# Accéder à la BD

- Créer un autre script dans l'IFS
  - item.js

```
var http = require('http');
var db = require('/QOpenSys/QIBM/ProdData/Node/os400/db2i/lib/db2');
http.createServer(function (req, res) {
    db.init();
    db.conn("*LOCAL");
    db.exec("SELECT * FROM ERPGFREE.ITEM", function(rs) {
        res.writeHead(200, {'Content-Type': 'text/plain'});
        res.end(JSON.stringify(rs));
    });
    db.close();
}).listen(8085, '10.2.0.1');
console.log('Server running at http://10.2.0.1:8085/');
```

# Accéder à la BD

- L'utilisation de l'extension db2 nécessite

```
var db =  
require('/QOpenSys/QIBM/ProdData/Node/os400/db2i/lib/db2'  
);
```

- Connexion à la Base de Donnée locale

```
db.conn("*LOCAL");
```

- Ou via le nom indiqué dans WRKRDBDIR

```
db.conn("NEPTUNE");
```

- Exécution d'une instruction SQL

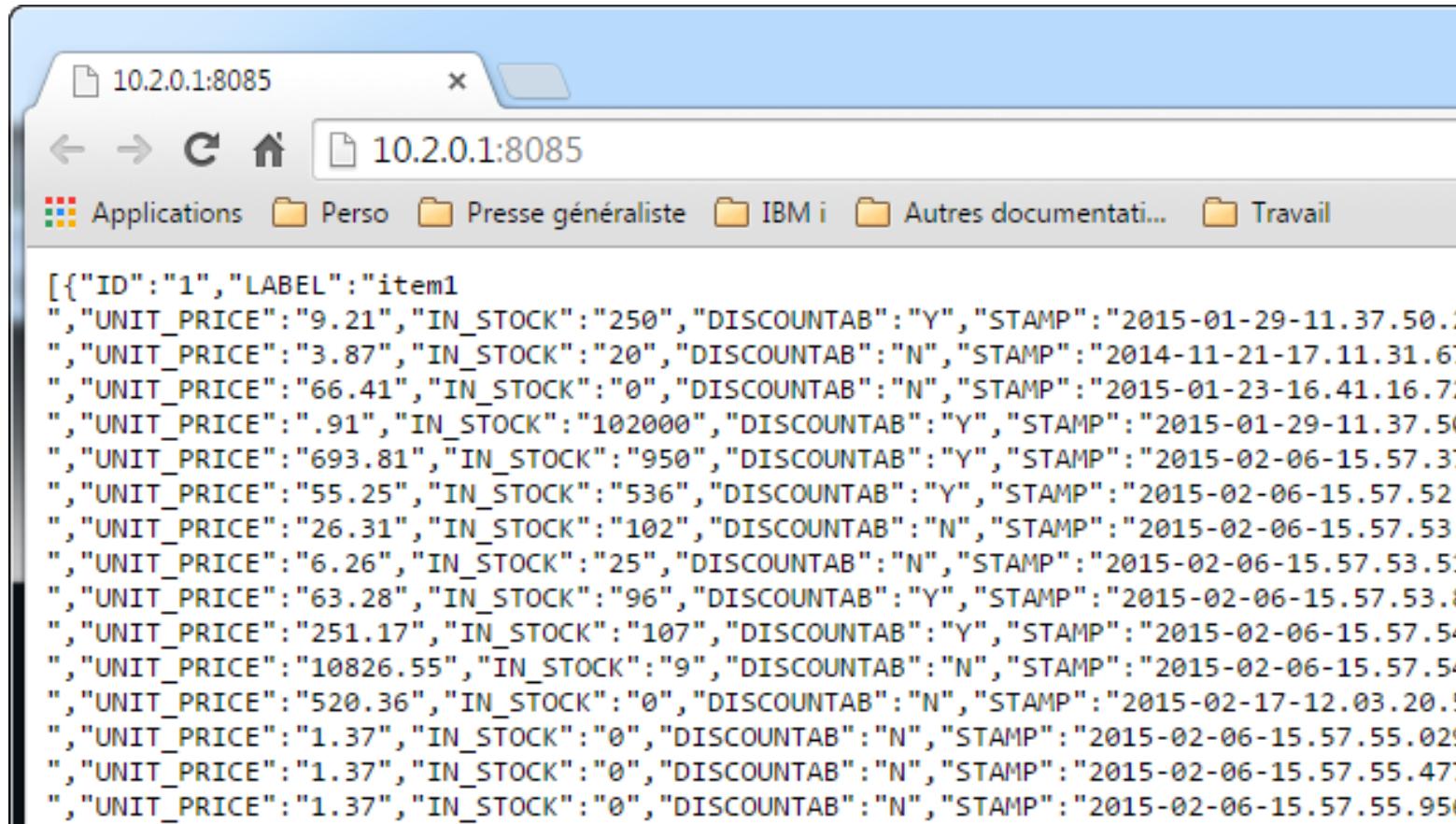
```
db.exec
```

- Encodage en JSON du résultat

```
JSON.stringify(rs)
```

# Accéder à la BD

## ■ Résultat



The screenshot shows a web browser window with the URL `10.2.0.1:8085` in the address bar. The page content displays a JSON array of items:

```
[{"ID": "1", "LABEL": "item1", "UNIT_PRICE": "9.21", "IN_STOCK": "250", "DISCOUNTAB": "Y", "STAMP": "2015-01-29-11.37.50.2"}, {"ID": "2", "LABEL": "item2", "UNIT_PRICE": "3.87", "IN_STOCK": "20", "DISCOUNTAB": "N", "STAMP": "2014-11-21-17.11.31.67"}, {"ID": "3", "LABEL": "item3", "UNIT_PRICE": "66.41", "IN_STOCK": "0", "DISCOUNTAB": "N", "STAMP": "2015-01-23-16.41.16.72"}, {"ID": "4", "LABEL": "item4", "UNIT_PRICE": ".91", "IN_STOCK": "102000", "DISCOUNTAB": "Y", "STAMP": "2015-01-29-11.37.50"}, {"ID": "5", "LABEL": "item5", "UNIT_PRICE": "693.81", "IN_STOCK": "950", "DISCOUNTAB": "Y", "STAMP": "2015-02-06-15.57.37"}, {"ID": "6", "LABEL": "item6", "UNIT_PRICE": "55.25", "IN_STOCK": "536", "DISCOUNTAB": "Y", "STAMP": "2015-02-06-15.57.52"}, {"ID": "7", "LABEL": "item7", "UNIT_PRICE": "26.31", "IN_STOCK": "102", "DISCOUNTAB": "N", "STAMP": "2015-02-06-15.57.53"}, {"ID": "8", "LABEL": "item8", "UNIT_PRICE": "6.26", "IN_STOCK": "25", "DISCOUNTAB": "N", "STAMP": "2015-02-06-15.57.53.51"}, {"ID": "9", "LABEL": "item9", "UNIT_PRICE": "63.28", "IN_STOCK": "96", "DISCOUNTAB": "Y", "STAMP": "2015-02-06-15.57.53.8"}, {"ID": "10", "LABEL": "item10", "UNIT_PRICE": "251.17", "IN_STOCK": "107", "DISCOUNTAB": "Y", "STAMP": "2015-02-06-15.57.54"}, {"ID": "11", "LABEL": "item11", "UNIT_PRICE": "10826.55", "IN_STOCK": "9", "DISCOUNTAB": "N", "STAMP": "2015-02-06-15.57.54"}, {"ID": "12", "LABEL": "item12", "UNIT_PRICE": "520.36", "IN_STOCK": "0", "DISCOUNTAB": "N", "STAMP": "2015-02-17-12.03.20.5"}, {"ID": "13", "LABEL": "item13", "UNIT_PRICE": "1.37", "IN_STOCK": "0", "DISCOUNTAB": "N", "STAMP": "2015-02-06-15.57.55.029"}, {"ID": "14", "LABEL": "item14", "UNIT_PRICE": "1.37", "IN_STOCK": "0", "DISCOUNTAB": "N", "STAMP": "2015-02-06-15.57.55.477"}, {"ID": "15", "LABEL": "item15", "UNIT_PRICE": "1.37", "IN_STOCK": "0", "DISCOUNTAB": "N", "STAMP": "2015-02-06-15.57.55.956"}]
```

# Exécution de commandes

- L'utilisation d'objets natifs nécessite **XMLService**
- On utilise l'extension
  - **i toolkit**
  - APIs disponibles
    - <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/Toolkit%20for%20i%20APIs>

# Exécution de commandes

```
var xt =  
require("/QOpenSys/QIBM/ProdData/Node/os400/xstoolkit/lib/itoolkit");  
  
var option = {xslib : "XMLSERVICE"} ;  
var conn = new xt.iConn("*LOCAL", "NB", "xxxxxxx", option);  
  
conn.add(xt.iCmd("RTVJOBA USRLIBL(?) SYSLIBL(?)"));  
conn.add(xt.iSh("system -i DSPSYSSTS"));  
  
function cb(str) {  
    console.log("==> RTVJOBA USRLIBL(?) SYSLIBL(?) : ");  
    console.log(xt.xmlToJson(str)[0].data);  
    console.log("==> DSPSYSSTS");  
    console.log(xt.xmlToJson(str)[1].data);  
}  
  
conn.run(cb);
```

# Exécution de commandes

- A l'appel
  - Les CCSID sont à régler ici ...

```
> node /home/nb/nodejs/pgme.js
==> RTVJOBA USRLIBL(?) SYSLIBL(?) :
[ { name: 'USRLIBL',
  value: 'QTEMP      OUTILS      EXPLOIT      QGPL' },
  { name: 'SYSLIBL',
  value: 'QSYS       QSYS2       QHLPSYS     QUSR SYS' } ]
==> DSPSYSSTS

                                         Informations sur l'état du système
                                         Page      1
me
5770SS1 V7R2M0 140418
```

		NEPTUNE	03/04/15 10:33:51 CEST		
% UC utilisée	time	0,6		ASP sys	%
% capacité UC débouchée utilisée	time	343,6 G			
ASP système utilisée	time	31,7767			
Intervalle secondaire totale	time	00:00:01		Mémoire	
Travaux connus du système temporaire utilisée	time	351		Mémoire	
% adresses permanentes utilisées	time	12888 M			
% adresses permanentes max	time	0,010		Mémoire max	

# Exécution de commandes

## ■ A l'appel

- L'utilisation du itoolkit nécessite le code

```
var xt =  
require("/QOpenSys/QIBM/ProdData/Node/os400/xstoolkit/lib/it  
oolkit");
```

- Indiquer la bibliothèque d'installation de XMLService

- itoolkit chercher QXMLSRV par défaut

```
var option = {xslib : "XMLSERVICE"} ;  
var conn = new xt.iConn("*LOCAL", "NB", "xxxxxxxx", option);
```

- Exécuter des commandes CL ou Shell

```
conn.add(xt.iCmd("RTVJOBA USRLIBL(?) SYSLIBL(?)"));  
conn.add(xt.iSh("system -i DSPSYSSTS"));
```

# Appel ILE

## ■ Script js

```
var http = require('http');
var xt = require('/QOpenSys/QIBM/ProdData/Node/os400/xstoolkit/lib/itoolkit');

http.createServer(function (req, res) {
// Connexion
var conn = new xt.iConn("*LOCAL");
var pgm = new xt.iPgm("WS_SRV", {"lib":"NB_UNIV_16", "func":"GETOPERATIONARRAY"});
// Préparation des paramètres
pgm.addParam("1","10i0");      // id
pgm.addParam("20","5u0")       // maxResult
pgm.addParam([
              // operationArray
              [0, "10i0"],
              [0, "10i0"],
              [0, "10i0"],
              [ "", "25A"],
              [0, "10p2"],
              ["0001-01-01-00.00.00.000000", "26a"]
            ], {"dim":"20"});
pgm.addParam("0","10i0")      // oprArray_length
pgm.addParam("", "3a");       // rc

conn.add(pgm.toXML());
```

# Appel ILE

```
// Appel du programme
conn.run(function (rsp) {
// Affichage du résultat
var results = xt.xmlToJson(rsp);
res.writeHead(200, {'Content-Type': 'text/plain'});
results.forEach(function(result, index){
    result.data.forEach(function(data, index2){
        res.write("type:" + data.type + " value:" + data.value + "\n");
    });
});
res.end();
})

}).listen(8085, '10.2.0.1');
console.log('Server running at http://10.2.0.1:8085/');
```

## ■ Lancement

```
SBMJOB CMD(QSH CMD('node /home/nb/ibm2016/getOperationArray.js'))
```

# Appel ILE

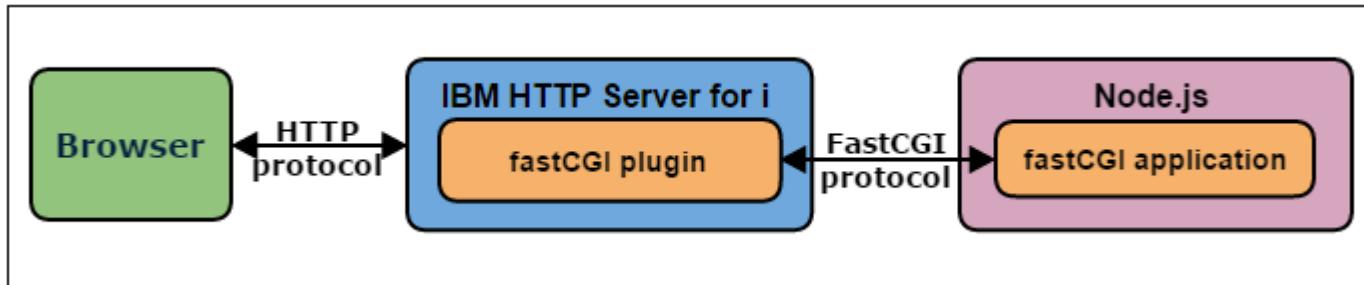
## ■ Résultat

— URL : <http://10.2.0.1:8085/>

```
type:10i0 value:1
type:5u0 value:20
type:10i0 value:1
type:10i0 value:1
type:10i0 value:1
type:25A value:Salaire
type:10p2 value:2763,25
type:26a value:2015-04-20-10.49.39.720153
type:10i0 value:2
type:10i0 value:1
type:10i0 value:1
type:25A value:Prêt
type:10p2 value:-1124,35
type:26a value:2015-04-18-10.49.39.720153
type:10i0 value:3
type:10i0 value:1
type:10i0 value:1
type:25A value:Prêt auto
type:10p2 value:-175,10
type:26a value:2015-04-16-10.49.39.720153
type:10i0 value:4
type:10i0 value:1
type:10i0 value:1
type:25A value:Frais déplacement
type:10p2 value:254,25
type:26a value:2015-04-14-10.49.39.720153
```

# Node.js via un serveur HTTP for i - FastCGI

- L'utilisation de FastCGI permet également d'exécuter Node.js via un serveur HTTP Apache en frontal



- Prérequis
  - Produits
    - 57330PS option 1 - IBM i open source solutions**
    - 5770SS1 option 33 - Portable app solutions environment**
    - 5733SC1 option 1 - OpenSSH, OpenSSL, zlib**
    - 5770DG1 \*BASE - IBM HTTP Server for i**
  - Group PTF
    - IBM i 7.1 PTF Group SF99368 - niveau 34**
    - IBM i 7.2 PTF Group SF99713 - niveau 8**

# Créer une instance HTTP

## ■ Instance de serveur HTTP

- Vous devez tout d'abord disposer d'une instance de serveur HTTP
  - Par l'outil d'administration : <http://partition:2001/HTTPAdmin>
  - La racine de votre serveur est par défaut

**/www/*instance***

[NODEJS](#) > General Server Configuration

### General Server Configuration

[General Settings](#) [Welcome Pages](#) [Configuration Includes](#) [Advanced](#)

Autostart: [Global](#) 

Server root directory: /www/nodejs

Configuration file: conf/httpd.conf

Document root:

Server name:

Fully qualified server host name:

Port:  

Server IP addresses and ports to listen on: 

	IP address	Port	Protocol
Example	All IP addresses	80	http
	*	12080	



# Configuration HTTP

## ■ Modifier la configuration du serveur

- Éditer le fichier de configuration **httpd.conf** (via l'interface web ou autre)
- Ajouter les lignes

```
# Module FastCGI  
LoadModule zend_enabler_module /QSYS.LIB/QHTTPSVR.LIB/QZFAST.SRVM  
...
```

```
# Ajout d'un handler fastCGI pour les requêtes avec extension *.jsx  
AddType application/x-httpd-javascript .jsx  
AddHandler fastcgi-script .jsx
```

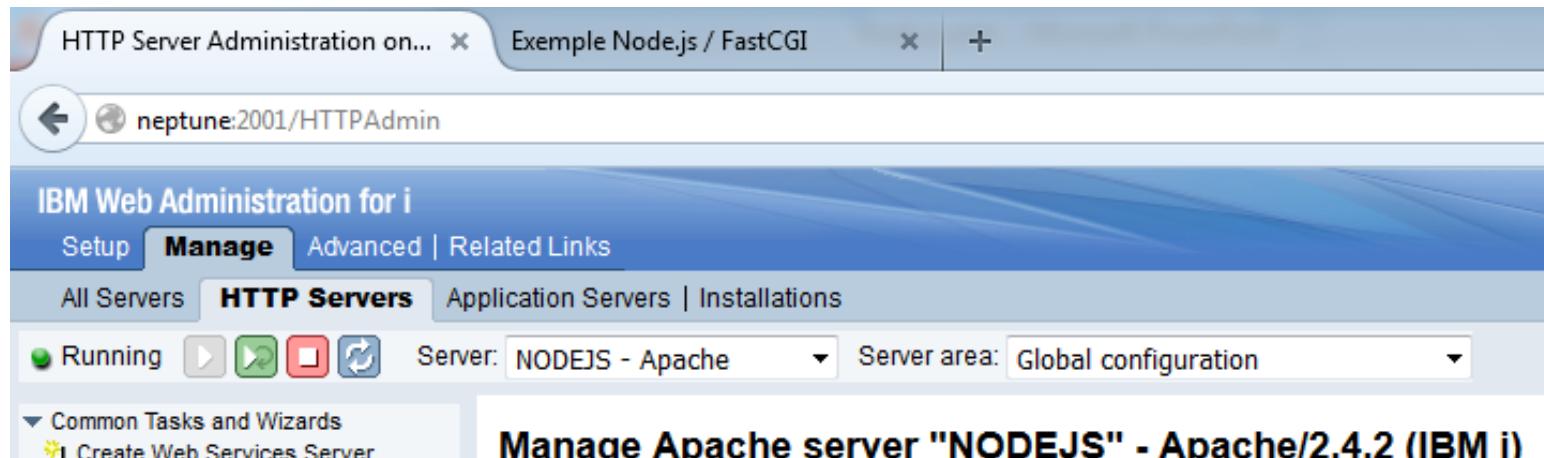
### Display Configuration File

HTTP server: NODEJS  
Selected file: /www/nodejs/conf/httpd.conf

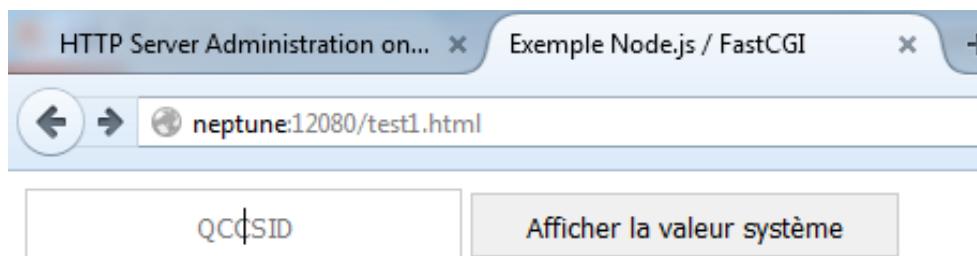
```
1 # Configuration originally created by Create HTTP Server wizard on Mon Sep 28 17:23:18 CEST 2015  
2  
3 # Module FastCGI  
4 LoadModule zend_enabler_module /QSYS.LIB/QHTTPSVR.LIB/QZFAST.SRVM  
5  
6 Listen *:12080  
7 DocumentRoot /www/nodejs/htdocs  
8 TraceEnable Off  
9 Options -FollowSymLinks  
10  
11 # Ajout d'un handler fastCGI pour les requêtes avec extension *.jsx  
12 AddType application/x-httpd-javascript.jsx  
13 AddHandler fastcgi-script.jsx  
14  
15 LogFormat "%h %T %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"\" combined  
16 LogFormat "%{Cookie}i \"%r\" %t\" cookie
```

# Démarrage

- Démarrer le serveur et vérifier l'affichage de la page créée



The screenshot shows the IBM Web Administration for i interface. The title bar says "HTTP Server Administration on... Exemple Node.js / FastCGI". The browser address bar shows "neptune:2001/HTTPAdmin". The main menu has tabs for "Setup", "Manage" (which is selected), "Advanced", and "Related Links". Below the tabs, there are links for "All Servers", "HTTP Servers" (which is selected), "Application Servers", and "Installations". A toolbar below the menu includes icons for running, stopping, and refreshing servers. The "Server:" dropdown is set to "NODEJS - Apache" and the "Server area:" dropdown is set to "Global configuration". On the left, a sidebar under "Common Tasks and Wizards" shows "Create Web Services Server". The main content area is titled "Manage Apache server \"NODEJS\" - Apache/2.4.2 (IBM i)".



The screenshot shows a browser window with the title bar "HTTP Server Administration on... Exemple Node.js / FastCGI" and the address bar "neptune:12080/test1.html". The page content displays the text "QCQSID" and a button labeled "Afficher la valeur système".

# fastCGI

- Le plugin open source fastCGI pour node.js est dans le répertoire

**/QOpenSys/QIBM/ProdData/Node/os400/node-fastcgi**

- Qui doit contenir deux fichiers modèle

**fastcgi.conf**

**fcgi.js**

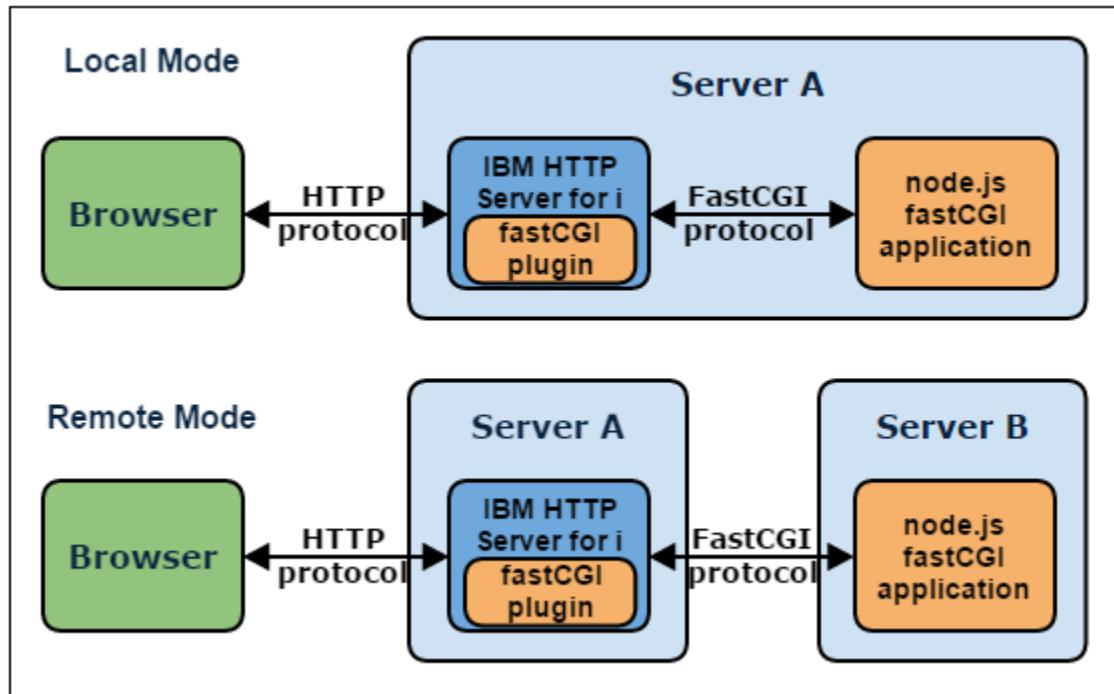
```
Répertoire . . . : /QOpenSys/QIBM/ProdData/Node/os400/node-fastcgi

Indiquez vos options et appuyez sur ENTREE.
 2=Réviser   3=Copier      4=Enlever   5=Afficher   7=Rebaptiser
 8=Afficher les attributs  11=Modifier répertoire en cours...

Opt    lien objet          Type        Attribut      Texte
_____
|   fastcgi.conf           STMF
|   fcgi.js                STMF
|   index.js               STMF
```

# Modes local et distant

- Deux modes de fonctionnement possibles
  - L'application **Node.js** peut s'exécuter sur la même machine, ou non



# Fichiers de configuration

- Copier les fichiers modèles

- Dans le répertoire de l'application fastCGI
  - Commandes QSH

```
cp -p /QOpenSys/QIBM/ProdData/Node/os400/node-fastcgi/fastcgi.conf  
      /www/nodejs/conf/
```

```
cp -p /QOpenSys/QIBM/ProdData/Node/os400/node-fastcgi/fcgi.js  
      /www/nodejs/conf/
```

- Il est conseillé

- De ne pas modifier les fichiers modèles fournis
  - Assurez-vous que QTMHHTTP a les droits \*RX sur les fichiers copiés

# fastcgi.conf

- Le fichier **fastcgi.conf** contient les directives suivantes
  - ; Native Mode --the HTTP server should run at the same IP address with the FastCGI server.

```
Server type="application/x-httpd-javascript"
CommandLine="/QOpenSys/QIBM/ProdData/Node/os400/node-
fastcgi/fcgi.js" StartProcesses="1"
```

- Indique que l'application FastCGI fonctionne en mode local, c'est à dire sur la même machine que le serveur HTTP
- Lors du démarrage du serveur HTTP, le script indiqué par **CommandeLine** est lancé

# fastcgi.conf

```
; Remote Mode --the FastCGI server can run at different IP  
addresses. Set 'Binding=' to the IP/port that the FastCGI  
server listens to.
```

```
; ExternalServer type="application/x-httdp-javascript"  
Binding="127.0.0.1:8080" ConnectionTimeout="300"  
RequestTimeout="300"
```

- Indique que l'application FastCGI et le serveur HTTP sont sur des machines distinctes
- **Binding** permet d'indiquer l'emplacement du serveur FastCGI

```
; Where to place socket files  
IpcDir /www/fastcgi/logs;
```

- **IpcDir** indique l'emplacement du fichier stream socket de communication avec FastCGI. Le répertoire doit exister

# fastcgi.conf

- Modifications à apporter à **fastcgi.conf**
  - Pour une configuration FastCGI local
    - Directive **Server**
    - **CommandLine="/www/nodejs/conf/fcgi.js"**
  - Pour une configuration FastCGI distant
    - Directive **ExternalServer**
    - **Binding= "127.0.0.1:8080"**
  - Directive **IpcDir**
    - **/www/nodjs/logs**

# fcgi.js

- Le fichier **fcgi.js** indique comment gérer les requêtes FastCGI
  - Par défaut

```
#!/QOpenSys/QIBM/ProdData/Node/bin/node
// This is a FastCGI code template to handle requests from HTTP server.
var fcgi = require('/QOpenSys/QIBM/ProdData/Node/os400/node-fastcgi');
var fcgiServer = fcgi.createServer(function(req, res) {
    if (req.method === 'GET') {
        res.writeHead(200, { 'Content-Type': 'text/plain' });
        res.end('GET: ' + req.url);
    } else {
        res.writeHead(501);
        res.end();
    }
});
// Native Mode -- Start the HTTP server and it will run the FastCGI server(this program)
// automatically.
// In this mode, set a valide 'CommandLine=' parameter in the fastcgi.conf
file.fcgiserver.listen();
// Remote Mode -- Run this program first. Then start the HTTP server. The HTTP server
// communicates with the FastCGI server remotely.
// In this mode, set the 'Binding=' parameter to the IP/port that the FastCGI server listens to
// in the fastcgi.conf file.
// fcgiServer.listen(8080);
```

# Travaux du serveur HTTP

- Après redémarrage du serveur web

Opt	S-syst/trav	Util en	cours	Type	% UC	Fonction	Etat
—	NODEJS	QTMHHTTP	BCH	0,0	PGM-QZHBMAIN	SIGW	
—	NODEJS	QTMHHTTP	BCI	0,0	PGM-QZSRLOG	SIGW	
—	NODEJS	QTMHHTTP	BCI	0,0	PGM-QZSRLOG	SIGW	
—	NODE.JS	QTMHHTTP	BCI	0,0	PGM-QZSRHTTP	SIGW	
—	NODEJS	QTMHHTTP	BCI	0,0	PGM-zfcgi	SELW	
—	NODEJS	QTMHHTTP	BCI	0,0	PGM-node	SELW	

- Le lancement du serveur HTTP provoque le lancement du moteur **Node.js**

# Test

- Tester notre application



The screenshot shows a web browser window with two tabs. The active tab is titled "HTTP Server Administration on..." and has the URL "http://neptune:12080/dspsysval.jsx?key=QCCSID". The content area displays the text "QCCSID = 1147".



The screenshot shows a web browser window with two tabs. The active tab is titled "HTTP Server Administration on..." and has the URL "http://neptune:12080/dspsysval.jsx?key=QMAXJOB". The content area displays the text "QMAXJOB = 163520".

# Outils

# GCC et chroot

- **GCC : GNU Compiler Collection**
  - Compilateur du monde Open Source
  - Nécessaire à la plupart des langages pour permettre de compiler ou recompiler des projets Open Source
- L'option apporte également plusieurs scripts shell
  - En béta
  - Développés par Litmis
  - Mis à jour par PTF ou via BitBucket
    - <https://bitbucket.org/litmis/ibmichroot/downloads>
  - Scripts
    - `chroot_setup.sh`
    - `pkg_setup.sh` : installation de rpms perzl.org
  - Outils
    - `rpm.rte` : depuis AIX Toolbox
    - `wget` : télécharge un fichier depuis le web

# Mais encore

- D'autres fonctionnalités Open Source sont d'ores et déjà annoncées de façon imminente
  - Orion
    - Éditeur web de source en mode hébergé (cloud)
    - Personnalisable et extensible (fondé sur Eclipse)
    - Le serveur peut être utilisé en mode cloud, ou redistribué et installé sur un serveur de l'entreprise (Java)
    - Le client est une application javascript (node.js)
  - Git
    - Logiciel de gestion de versions le plus utilisé (devant SVN/CVS)
    - Écrit à l'origine par Linus Torvald, le père du noyau Linux
      - « git » signifiant « connard » ...

Companies & Projects Using Git

Google

facebook

Microsoft

twitter

LinkedIn

NETFLIX



ANDROID



RAILS



GNOME

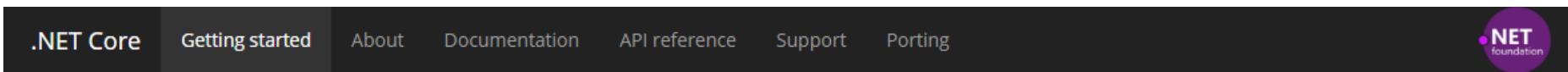
eclipse



rance 2016

# Mais encore

- Des rumeurs ...
  - .Net : le core et le framework (runtime server) ainsi qu'une partie de l'IDE sont passés sous licence Open Source en 2014
  - Un portage du runtime pour Linux et OS X



The screenshot shows the top navigation bar of the .NET Core website. It includes links for ".NET Core", "Getting started", "About", "Documentation", "API reference", "Support", and "Porting". On the right side, there is a purple circular button with the ".NET foundation" logo.

**It is very easy to get started with .NET Core on your platform of choice.**

For ASP.NET Core 1.0 + .NET Core RC instructions, please go to the [Get ASP.NET Core 1.0 website](#).

You just need a shell, a text editor and 10 minutes of your time. Ready? Set? Let's go!

Windows

Ubuntu (14.04)

OS X (10.11)

Docker

- Qui laisse envisager des possibilités via PASE ?

**Merci  
Q/R**

# Ressources

- <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/Open%20Source%20Technologies>
- <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/GCC>
- <https://developer.ibm.com/open/>
- <https://orionhub.org>
- <https://git-scm.com/>
- <https://www.dotnetfoundation.org/>
- <https://github.com/Microsoft/dotnet>
- <http://powerruby.com/>
- <https://powerruby.worketc.com/kb>
- <https://www.ruby-lang.org/fr/>
- <http://rubyonrails.org/>
- <http://iprodeveloper.com/application-development/discover-power-rails-ibm-i>
- <https://nodejs.org/>
- <http://www.fastcgi.com/>
- <http://yips.idevcloud.com/wiki/index.php/FastCGI/FastCGI>
- <https://bitbucket.org/litmisibmichroot/downloads>