Bluemix Hands-On Workshop

Section 5 - Create your app

Version:	9
Last modification date:	6-Oct-14
Owner:	IBM Ecosystem Development

Table of Contents

Exercise 5.a – Your first Node.js application

In this exercise we switch from Java to Node, which uses javascript on the server.

A typical "hello world" application is :

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(3000);
```

Create this file using a text editor installed on your system[1] in a clean directory and call it app.js. If you have a local install of Node, you could run this using `node app.js`

This will start a web server and every request will respond with Hello World. To run this on Bluemix we need to make a couple of changes. The first change is to get the application to listen on the port Bluemix has assigned for the application. Environment variables are used by Bluemix to communicate configuration to the application:

```
var http = require('http');
var appport = process.env.VCAP_APP_PORT || 3000;
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(appport);
```

The next is to add a step is to create a package.json – this is required to run a node application on Bluemix.

```
{
  "name": "BINodeTest",
  "version": "0.0.1",
  "description": "Sample Node application"
}
```

Create a new text file called package.json in the same directory as app.js and enter the content above.

To deploy the application on Bluemix we will initially use the CLI. Open a command/terminal window and change to the directory containing app.js and package.json then enter the command

```
cf push BINodeTest -c 'node app.js'
```

If you navigate to the Bluemix Web UI you should see the application running – with "Hello World" displayed in the browser when you launch the application.

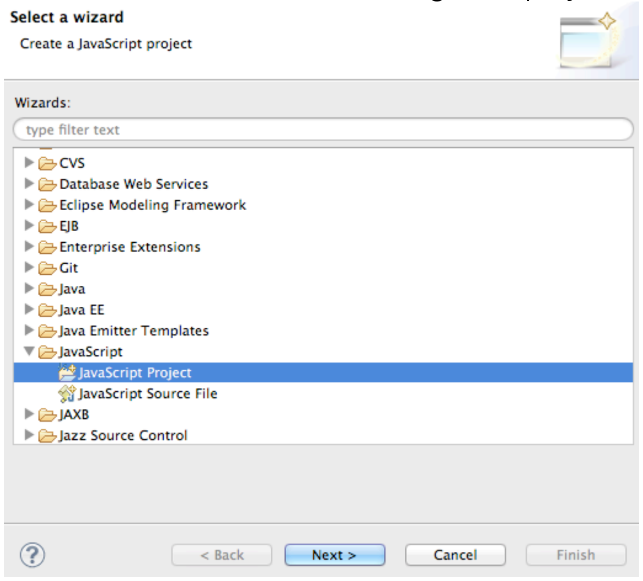
[1] notepad on windows or TextEdit on Mac – please use text mode with smart features turned off, such as smart quotes ‘’

Exercise 5.b – Working with Node using Eclipse

In this exercise you will use Eclipse to create a Node.js application then deploy it to Bluemix.
Launch Eclipse. If using a fresh workspace you need to create the Bluemix server configuration – we did this in section 3.

Create a new javascript project File -> New -> Other

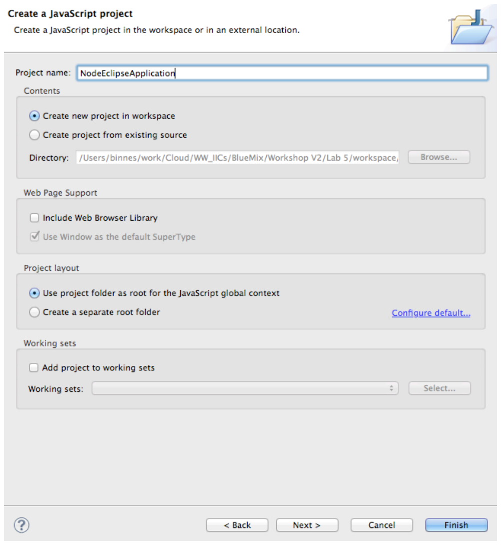
When the Select a wizard dialog is displayed select Javascript -> JavaScript Project.


Select a wizard
Create a JavaScript project

Wizards:
type filter text

- CVS
- Database Web Services
- Eclipse Modeling Framework
- EJB
- Enterprise Extensions
- Git
- Java
- Java EE
- Java Emitter Templates
- JavaScript
  - JavaScript Project
  - JavaScript Source File
- JAXB
- Jazz Source Control

When the Create a JavaScript project dialog appears. Give the project a name and remove the Web Page Support option.


Create a JavaScript project
Create a JavaScript project in the workspace or in an external location.

Project name: NodeEclipseApplication

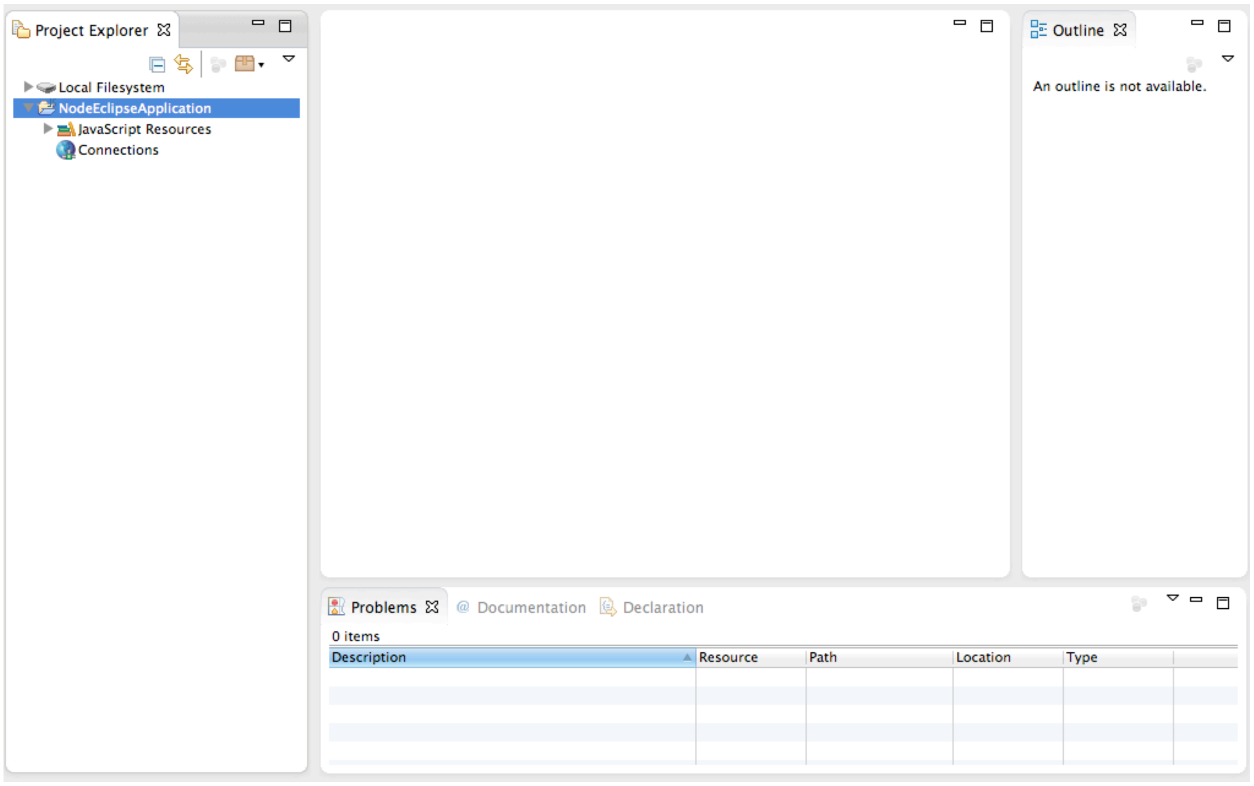
Contents
(●) Create new project in workspace
( ) Create project from existing source
Directory: /Users/binnes/work/Cloud/WW_IICs/BlueMix/Workshop V2/Lab 5/workspace,

Web Page Support
[ ] Include Web Browser Library
[✓] Use Window as the default SuperType

Project layout
(●) Use project folder as root for the JavaScript global context
( ) Create a separate root folder

Working sets
[ ] Add project to working sets
Working sets:

click Finish – will switch to JavaScipt perspective – accept this recommendation.

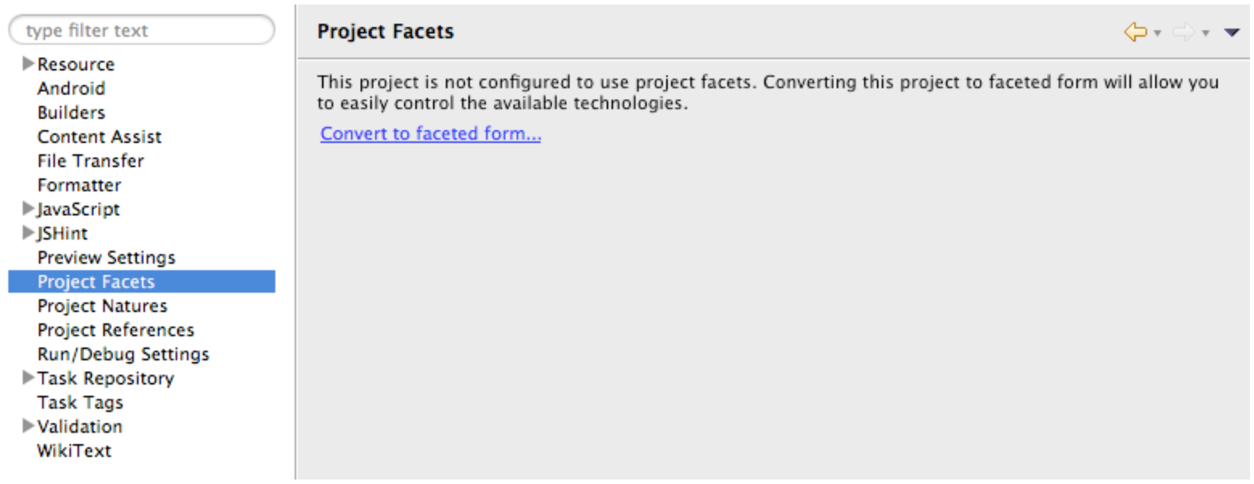


An outline is not available.

0 items

Description	Resource	Path	Location	Type

To add Bluemix support to the project you need to add a Facet to the project. Right-click the project name and select Project Facets – select 'Convert to faceted form…'

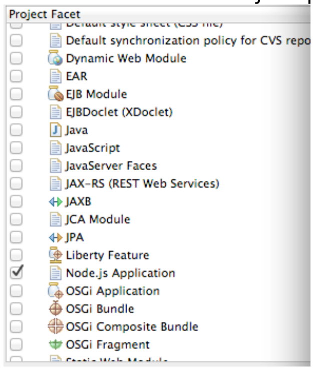


Project Facets

This project is not configured to use project facets. Converting this project to faceted form will allow you to easily control the available technologies.

Convert to faceted form...

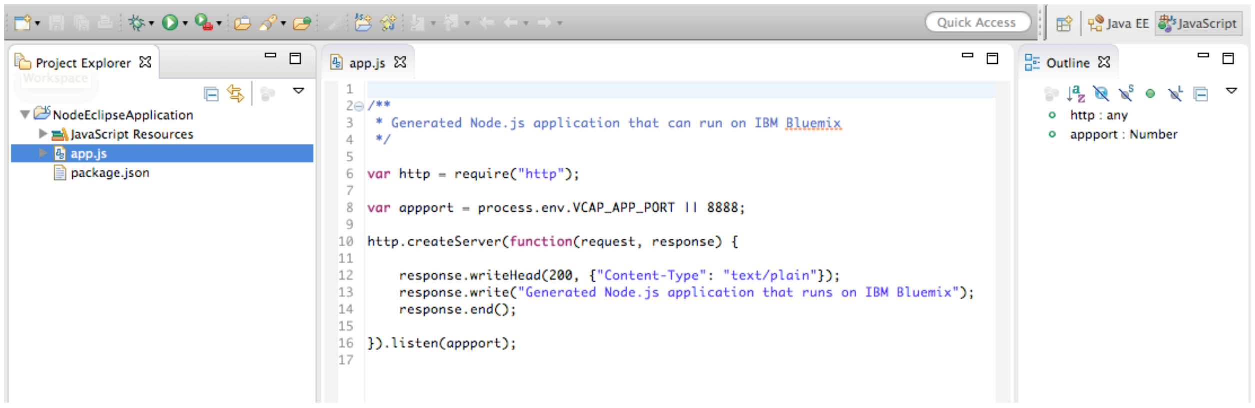
then select 'Node.js Application'



Project Facet

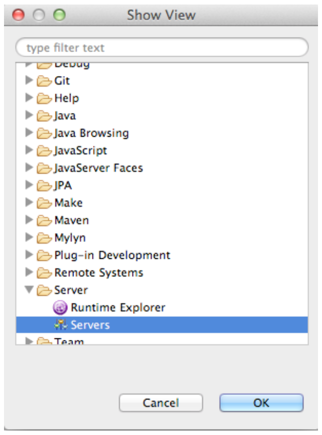
- [ ] Default synchronization policy for CVS repo
- [ ] Dynamic Web Module
- [ ] EAR
- [ ] EJB Module
- [ ] EJBDoclet (XDoclet)
- [ ] Java
- [ ] JavaScript
- [ ] JavaServer Faces
- [ ] JAX-RS (REST Web Services)
- [ ] JAXB
- [ ] JCA Module
- [ ] JPA
- [ ] Liberty Feature
- [✓] Node.js Application
- [ ] OSGi Application
- [ ] OSGi Bundle
- [ ] OSGi Composite Bundle
- [ ] OSGi Fragment

You will see that app.js and package.json are added to the project and the contents of the file will look similar.

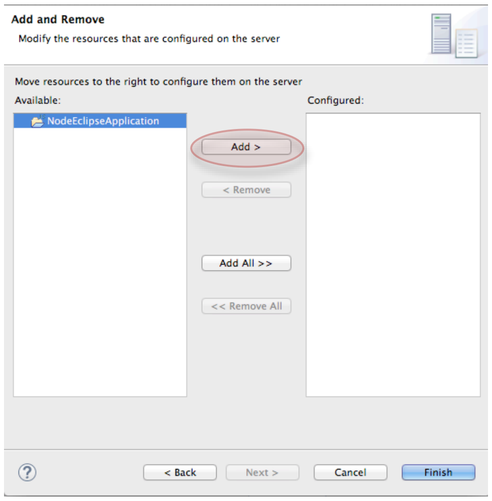


```
/**
 * Generated Node.js application that can run on IBM Bluemix
 */

var http = require("http");

var appport = process.env.VCAP_APP_PORT || 8888;

http.createServer(function(request, response) {

    response.writeHead(200, {"Content-Type": "text/plain"});
    response.write("Generated Node.js application that runs on IBM Bluemix");
    response.end();

}).listen(appport);
```

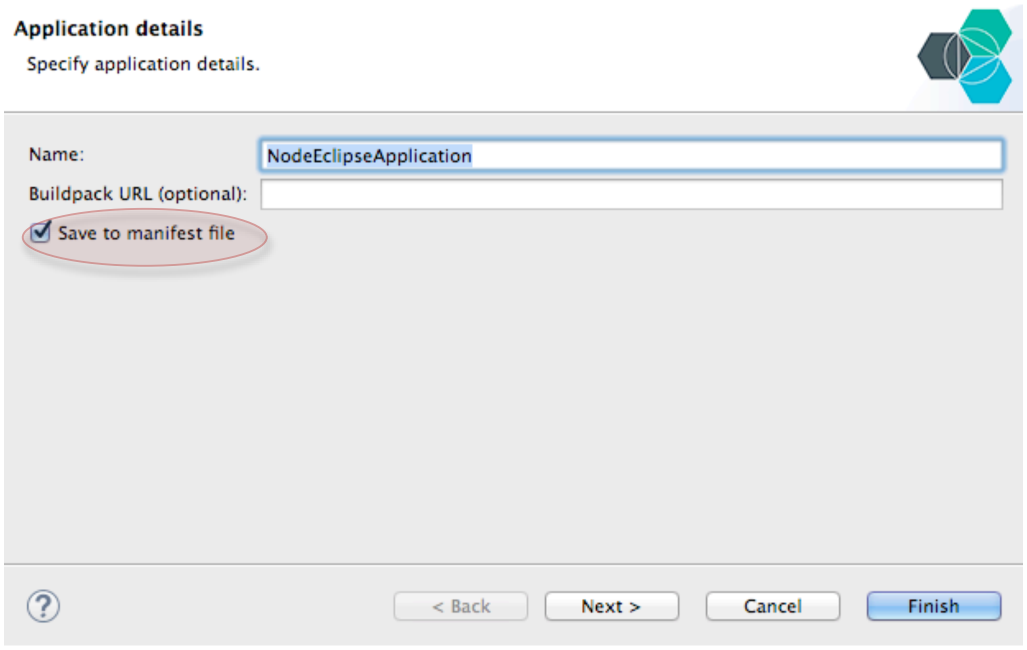
To deploy the application you need to open the Servers view Window -> Show View -> Other. Then in the dialog that opens select Server -> Servers.



Right-click Bluemix then select your Node application and move it to the Configured column.



The Bluemix deploy wizard will start. Eclipse does not use the manifest file, but you can select to generate a file as a result of the Eclipse deploy:



Application details

Specify application details.

Name: NodeEclipseApplication

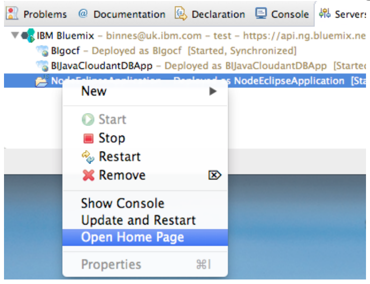
Buildpack URL (optional):

[✓] Save to manifest file

Give the application a unique name, modify the memory requirement to 128M and select to deploy 2 instances. Complete the deployment – there are no services needed for this deployment.

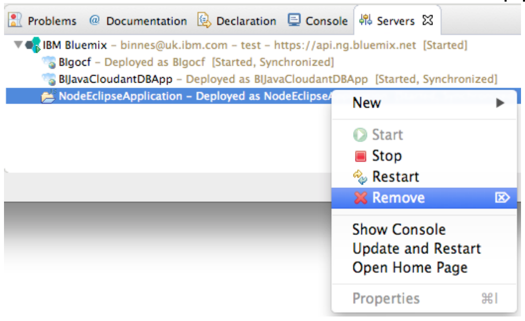
Once the application is running open the generated manifest.yml file and verify the options to selected are in the manifest file.

Once deployed you can run the Node application by right-clicking the application in the Servers view then selecting to open home page.



```
IBM Bluemix – binnes@uk.ibm.com – test – https://api.ng.bluemix.ne
  BIgocf – Deployed as BIgocf [Started, Synchronized]
  BIJavaCloudantDBApp – Deployed as BIJavaCloudantDBApp [Starte
  NodeEclipseApplication – Deployed as NodeEclipseApplication [Sta
```

Once you have tested the application right click the application under the Bluemix server definition and select to remove the application

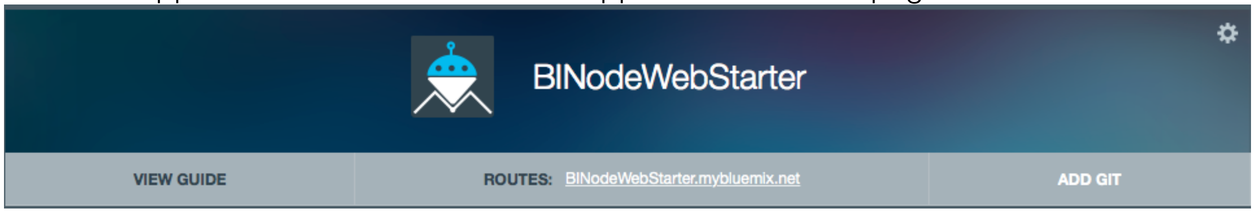


```
IBM Bluemix – binnes@uk.ibm.com – test – https://api.ng.bluemix.net [Started]
  BIgocf – Deployed as BIgocf [Started, Synchronized]
  BIJavaCloudantDBApp – Deployed as BIJavaCloudantDBApp [Started, Synchronized]
  NodeEclipseApplication – Deployed as NodeEclipse
```

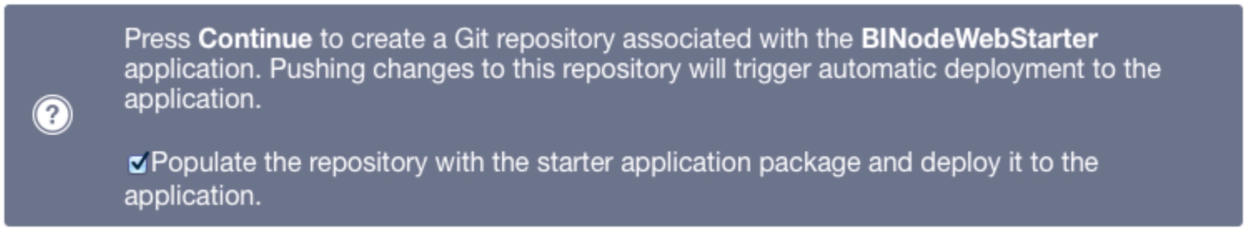
Exercise 5.c – Working with Node using DevOps Services

In Bluemix deploy the Node.js Cache Web Starter from the Catalog view in the Bluemix UI.

Once the application has started from the application overview page select 'ADD GIT'



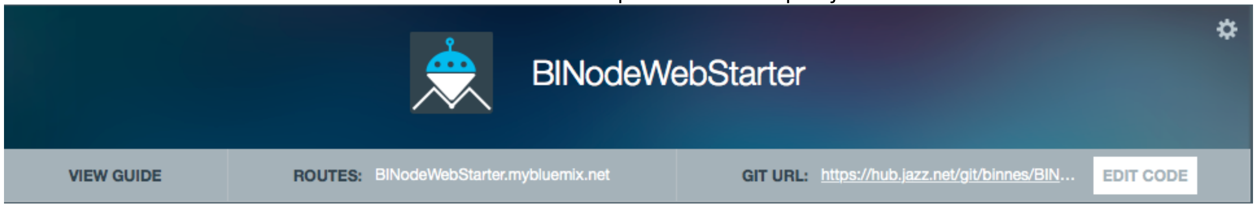
Ensure the "Populate the repository with the starter application package …" is selected



Press **Continue** to create a Git repository associated with the **BINodeWebStarter** application. Pushing changes to this repository will trigger automatic deployment to the application.

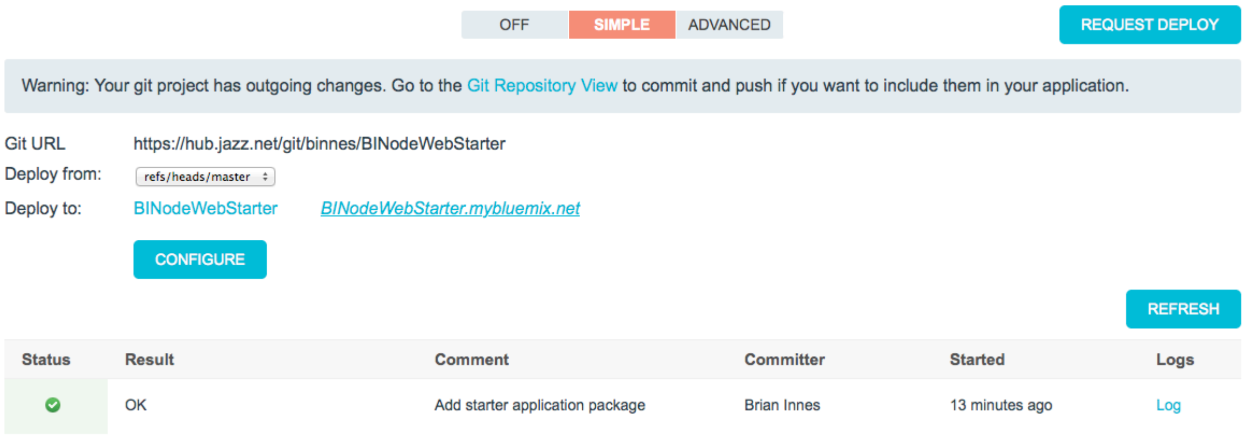
☑ Populate the repository with the starter application package and deploy it to the application.

Select 'EDIT CODE' to be taken to the DevOps Services project


BINodeWebStarter
VIEW GUIDE
ROUTES: BINodeWebStarter.mybluemix.net
GIT URL: https://hub.jazz.net/git/binnes/BIN...
EDIT CODE

The Node.js Web Starter project uses the express framework. If you open the package.json in the DevOps editor you will see the project dependencies.

Node applications do not need to be compiled before they are deployed, so the simple Build & Deploy option within DevOps Services can be used. If you select the BUILD & DEPLOY tab , you will see the project is already configured.

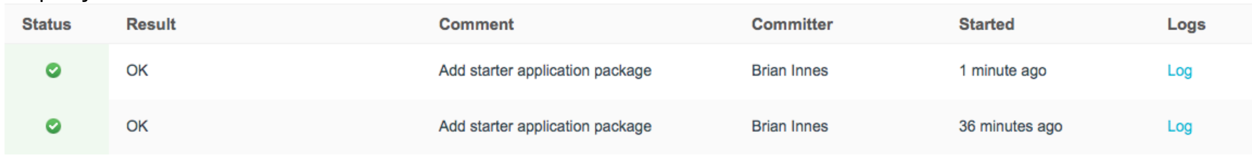


Warning: Your git project has outgoing changes. Go to the Git Repository View to commit and push if you want to include them in your application.

Git URL	https://hub.jazz.net/git/binnes/BINodeWebStarter
Deploy from:	refs/heads/master
Deploy to:	BINodeWebStarter BINodeWebStarter.mybluemix.net

Status	Result	Comment	Committer	Started	Logs
✓	OK	Add starter application package	Brian Innes	13 minutes ago	Log

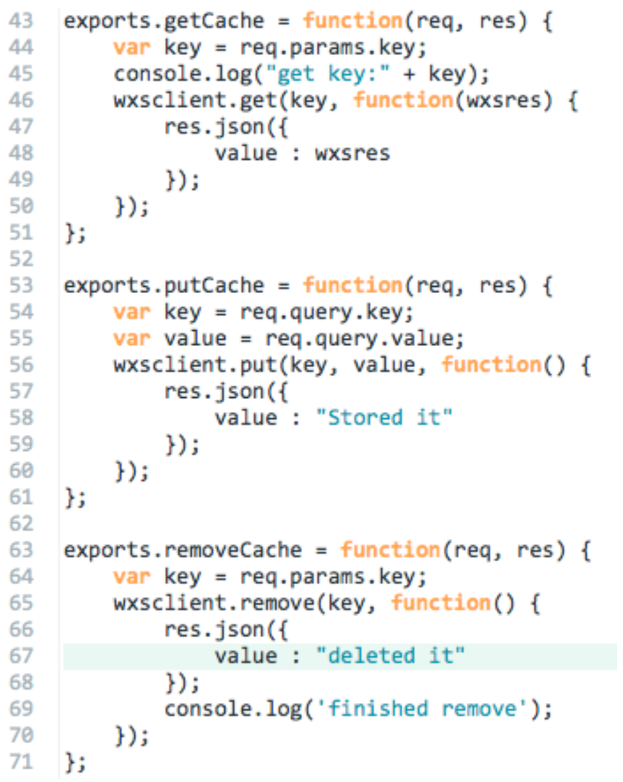
To test the configuration, select REQUEST DEPLOY. After a short while you will see the deploy result



Status	Result	Comment	Committer	Started	Logs
✓	OK	Add starter application package	Brian Innes	1 minute ago	Log
✓	OK	Add starter application package	Brian Innes	36 minutes ago	Log

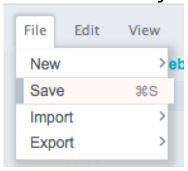
The Simple automatic deployment works the same way as the Advanced that we used in section 4. Select EDIT CODE and open file routes/cache.js in the editor. Scroll to the bottom of the file and locate the putCache and removeCache functions. Change the strings that are displayed on the web UI when a cache operation stores or removes a key:

```
Put successfully. => Stored it
Remove successfully. => deleted it
```

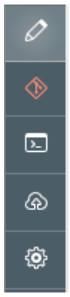


```
exports.getCache = function(req, res) {
    var key = req.params.key;
    console.log("get key:" + key);
    wxsclient.get(key, function(wxsres) {
        res.json({
            value : wxsres
        });
    });
};

exports.putCache = function(req, res) {
    var key = req.query.key;
    var value = req.query.value;
    wxsclient.put(key, value, function() {
        res.json({
            value : "Stored it"
        });
    });
};

exports.removeCache = function(req, res) {
    var key = req.params.key;
    wxsclient.remove(key, function() {
        res.json({
            value : "deleted it"
        });
        console.log('finished remove');
    });
};
```

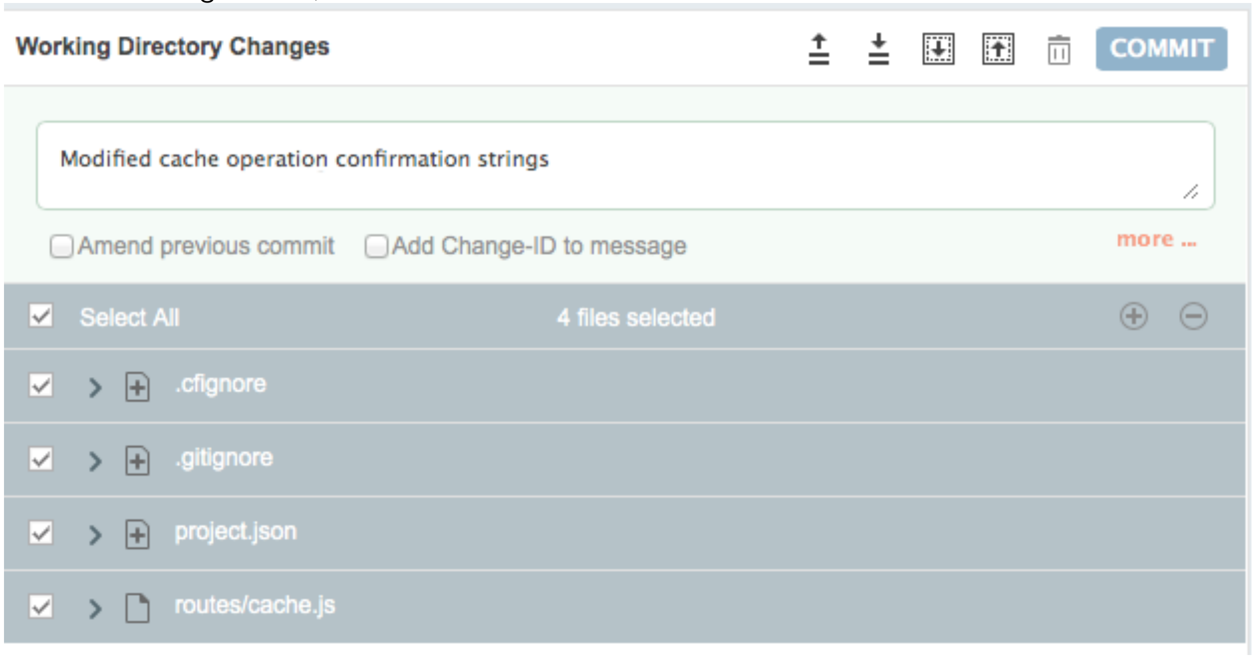
Make sure you save the file



then switch to the Git Repository



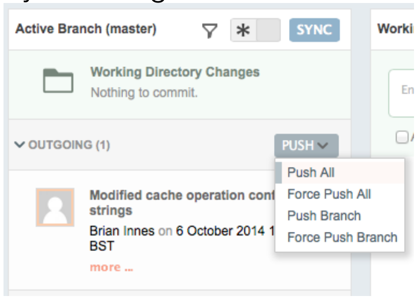
select all changed files, enter a comment for the commit



Modified cache operation confirmation strings

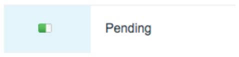
.cfignore
.gitignore
project.json
routes/cache.js

then select COMMIT. Once you committed the changes push them to the remote branch by selecting Push -> Push All.

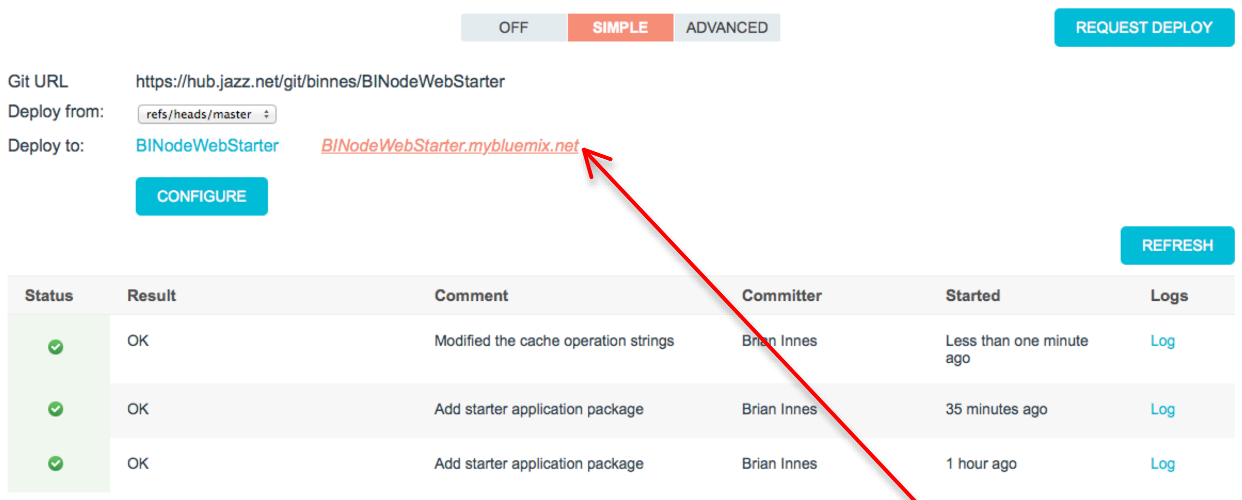


Modified cache operation conf strings
Brian Innes on 6 October 2014 1 BST

Quickly switch to the BUILD & DEPLOY tab and you will see that a deploy has been initiated.



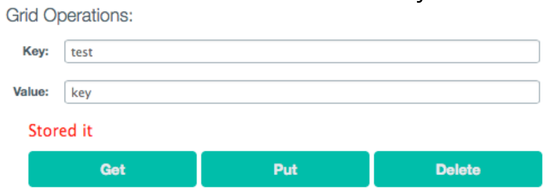
After a short while the deploy will complete and you should see a success status with the commit message displayed in the comment.



Git URL https://hub.jazz.net/git/binnes/BINodeWebStarter
Deploy from: refs/heads/master
Deploy to: BINodeWebStarter BINodeWebStarter.mybluemix.net

Status	Result	Comment	Committer	Started	Logs
✓	OK	Modified the cache operation strings	Brian Innes	Less than one minute ago	Log
✓	OK	Add starter application package	Brian Innes	35 minutes ago	Log
✓	OK	Add starter application package	Brian Innes	1 hour ago	Log

You can launch the application by selecting the link on the 'Deploy to:' line.
Add a value to the cache and you will see the modified message displayed



Grid Operations:

Key: test

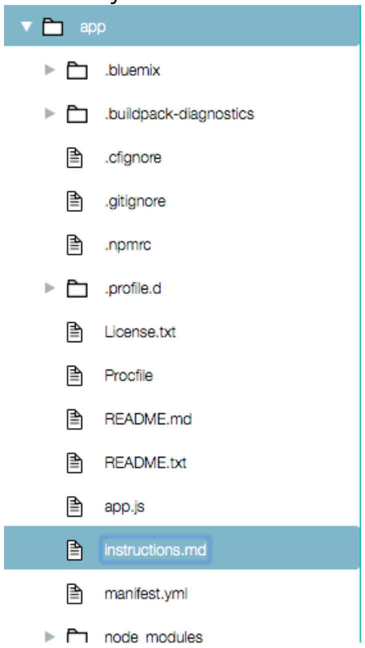
Value: key

Stored it

Exercise 5.d – Selecting files to include in your application

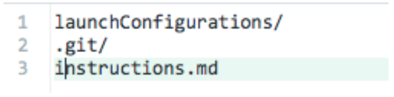
The .cfignore file controls what files are pushed to Bluemix.

In the Bluemix Web UI open the Node.js Web Starter app that was deployed in previous sections of this section. Navigate to the Files and Logs section and open the app directory



you will see the files that are deployed, including a file named instructions.md. This is a file that explains what the starter application is and is not part of the application. We want to stop this file from being deployed to Bluemix.

In DevOps Services open the '.cfignore' file in the editor and add instructions.md to the end of the file on a new line.



```
launchConfigurations/
.git/
instructions.md
```

Save the file then commit and push the changes to the remote repository. This will force the project to be redeployed.

Switch back to the Bluemix UI and open the Files and Logs section of the application. You should now see that the 'instructions.md' file is no longer included with the application deployed to Bluemix.