

# Preparing the environment

## 1. Install Eclipse

<https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/lunasr1>

## 2. Install Postman on Google Chrome

## 3. Install CF CLI

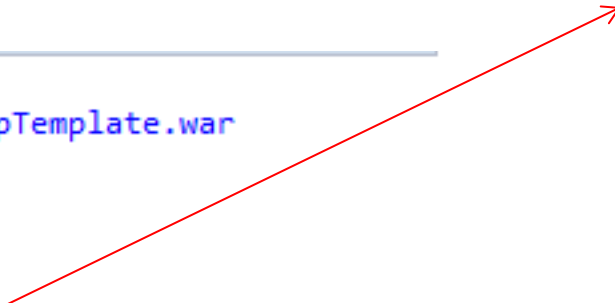
<https://github.com/cloudfoundry/cli/releases>

## Preparing your cloud application

1. Download the zip file from GitHub  
<https://github.com/rajrin/BootcampJaxRS>
2. Import the zip file in eclipse as a new project (for Node simply unzip in desired folder)
3. Edit the manifest file for the application

Change Host & Name

```
1 applications:
2   - path: target/BootcampTemplate.war
3     memory: 512M
4     instances: 1
5     domain: cfapps.io
6     name: TcsBootcampApp
7     host: TcsBootcamp
8     disk_quota: 1024M
9
```



## Managing your cloud application

1. Create an account on <http://run.pivotal.io>

Login to your account

You should see one organization with the name “Main Org”

Create a new space called “Bootcamp”

2. In the command window do the following:

Type `cf login -a` <https://api.run.pivotal.io>

Provide email/password

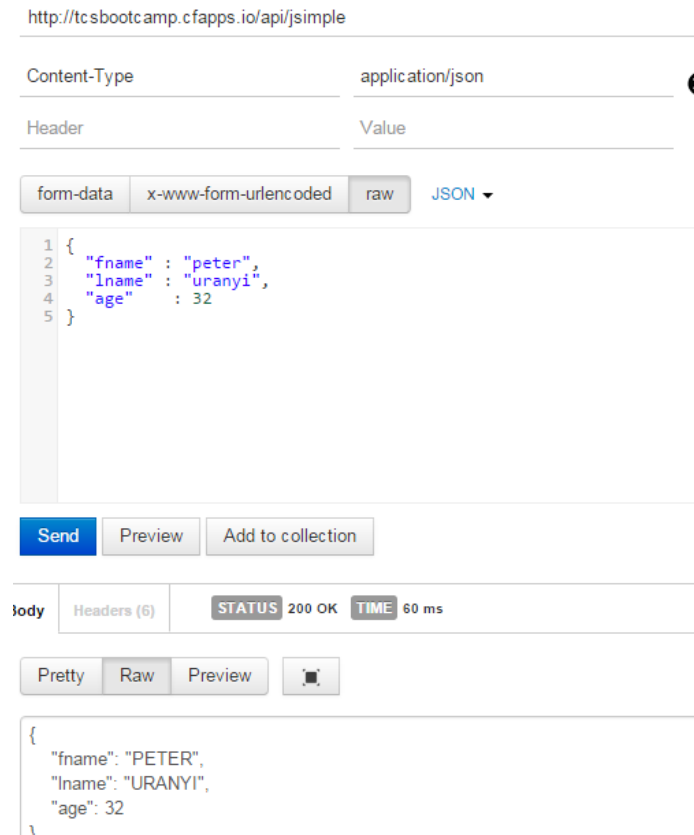
3. Use the following commands as needed

- To push your application to *Pivotal*, go to the root directory of your project and type the command **cf push**
- To **stop** your application type the command **cf stop app\_name**
- To **start** your application type the command **cf start app\_name**
- To check **logs** (tailing) **cf logs app\_name**
- Deployment target **cf target** (edit it using `-o -s` )

## Suggestion: Install the original application & test before making changes

### Test your Application

1. Use the postman on chrome
2. Make sure you are using the right method (GET, POST ...)
3. Make sure the content-type is set



## Creating single page applications

1. Download the samples from  
<https://github.com/rajrin/Single-Page-Apps-Sample/tree/master>
2. Change the API URL/Endpoint in ajax calls
3. Change the form fields/ids and populate data received from API invocation

## Mongo DB

1. Create the mongo db account on <https://mongolab.com/>
2. Create the database
3. Create the collections
4. Create the user(s)
5. Download & install mongo 3.x for local development

# NodeJS API Template

<https://github.com/rajrin/BootcampNodeJS.git>

1. Download the template from above – it has everything you need for creating a REST API with Mongo DB
2. Try out a running instance of the template @ [I may bring this app down so contact me if you need an overview – 768988]

<http://tcsbootcampnode.cfapps.io/api/v1/customer>

The screenshot shows a REST client interface with the following details:

- URL:** `http://tcsbootcampnode.cfapps.io/api/v1/customer`
- Buttons:** Send, Preview, Add to collection
- Response Headers:** 7 headers (not expanded)
- Status:** 200 OK
- Time:** 568 ms
- Body:** Pretty, Raw, Preview (Pretty is selected)
- Response Body (JSON):**

```
[
  {
    "_id": "5647ad85e4b04a6f73355611",
    "customer_id": 1000,
    "fname": "John",
    "lname": "Doe",
    "address": {
      "street": "1 main street",
      "city": "some city",
      "state": "NJ",
      "zip": "08816"
    },
    "Auto": 0,
    "Home": 1,
    "Life": 1
  },
  {
    "_id": "5647af52e4b04a6f7335562e",
    "customer_id": 1001,
    "fname": "Bharat",
    "lname": "Desai",
  }
]
```

## Git – Your code

<https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>

Create an account on GitHub

Download the Git shell on your machine

Upload project to a repo – share with *Rajeev*

*\*OR\**

You can provide the source to Rajeev – will upload to common repo

## **Download Mongo DB**

<https://www.mongodb.org/downloads#production>

## **Download Aptana studio (for Node & Single page app)**

<http://www.aptana.com/products/studio3/download.html>