

CoAP Tutorial for Eclipse

This tutorial shows how to run a CoAP server using Eclipse, and view the CoAP resources through Copper, the CoAP user-agent.

1. Download Eclipse for Java developers.

<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/neon1a>

If you already have Eclipse installed, then you don't need to do this step. No need to install, unzip the package and you can run Eclipse now.

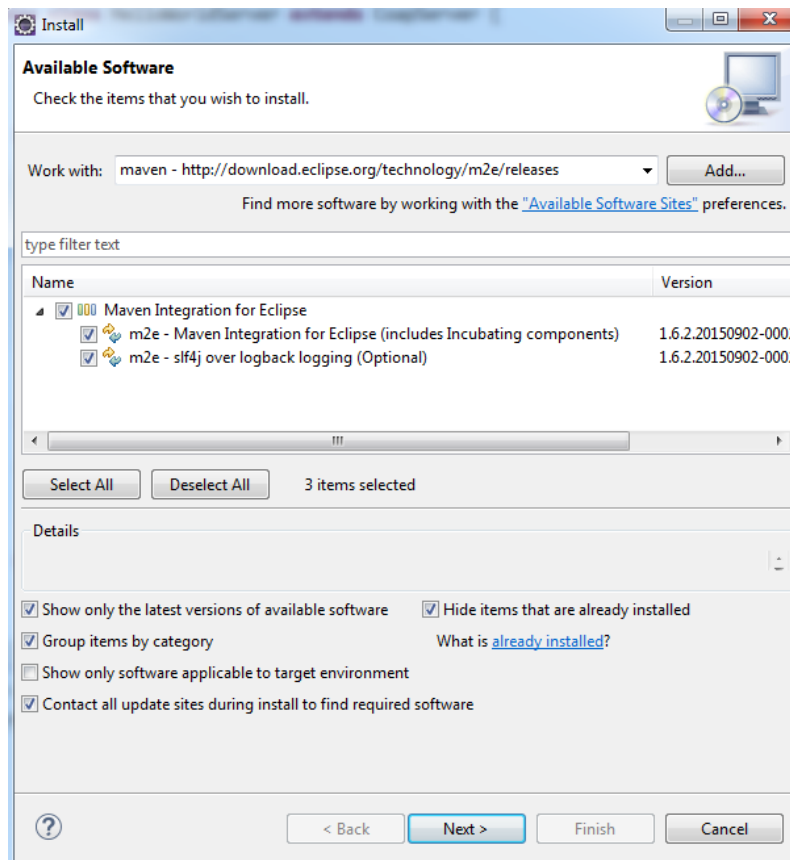
2. Install Maven plugin for Eclipse

Maven may have already come together with Eclipse, if not, install it as below.

Run Eclipse, click *Help-> Install New Software...*, in the *Work with:* field, enter the following website:

<http://download.eclipse.org/technology/m2e/releases>

In the output, select *Maven Integration for Eclipse*, click *Next*, and *Next*, until it is installed. You may need to restart Eclipse after the installation.



3. Download CoAP library

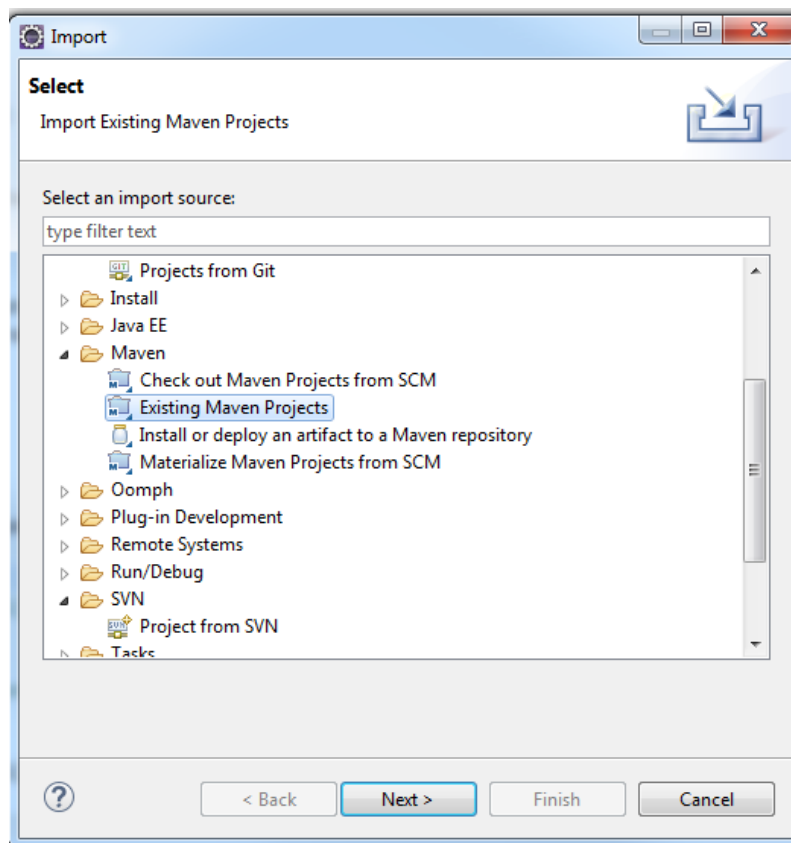
On windows, download the .zip file directly from the github page:

<https://github.com/eclipse/californium>

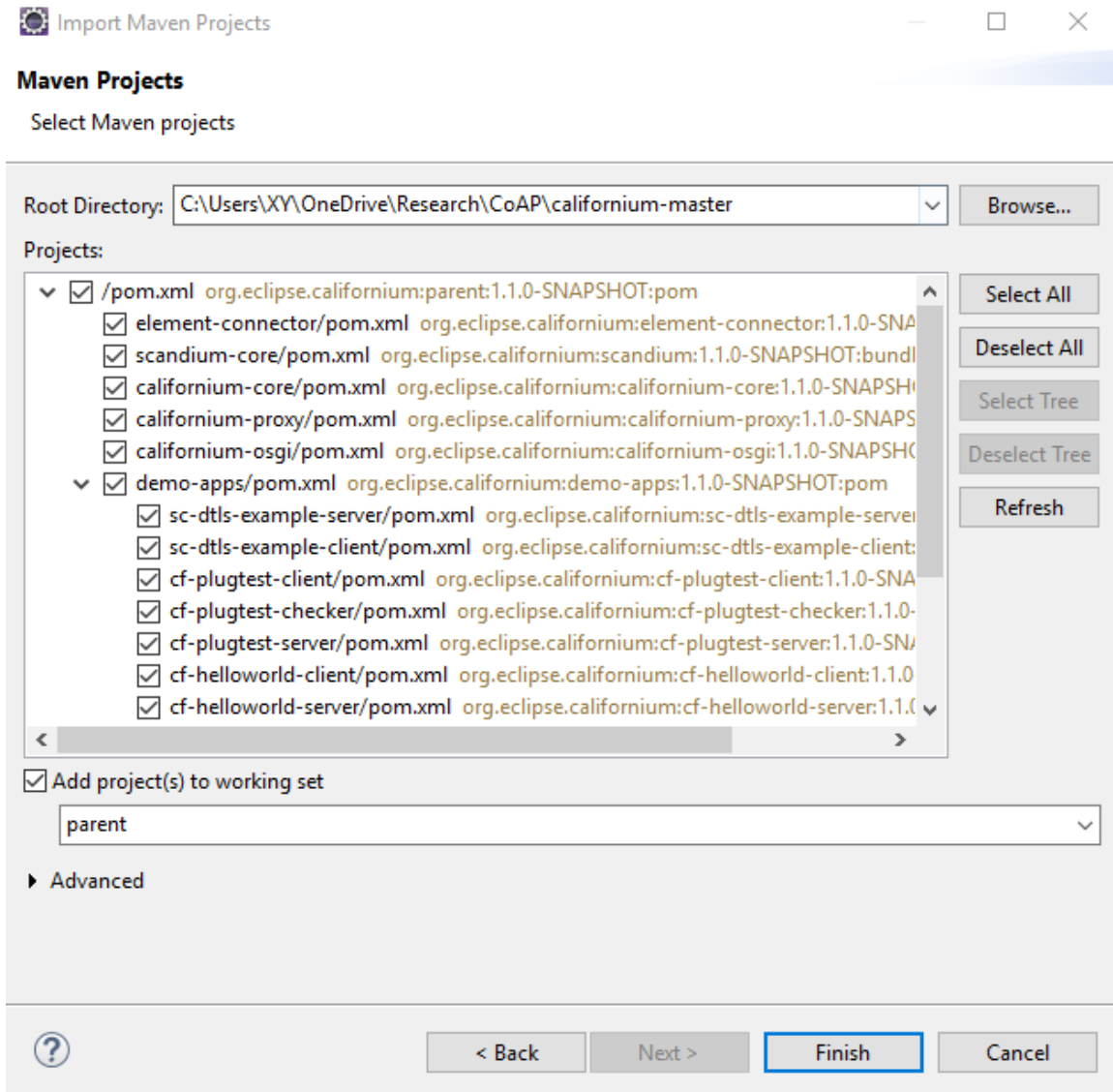
Then unzip it.

4. Import the Californium to Eclipse:

Click *File -> Import...*, you would see the following dialog.



Select *Maven->Existing Maven Project*, then click *Next*. You would see the following dialog. In the *Root Directory:*, browse to your *californium-master* folder. Check all the *pom.xml* files, then click *Finish*.

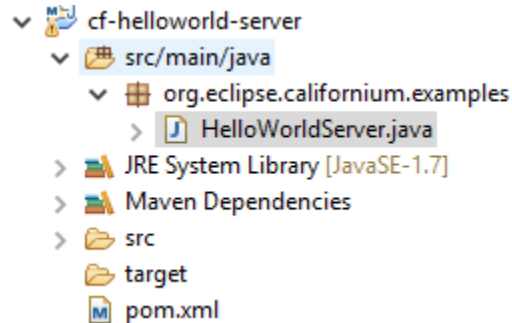


You would see all the Californium libraries and projects are listed on the left.

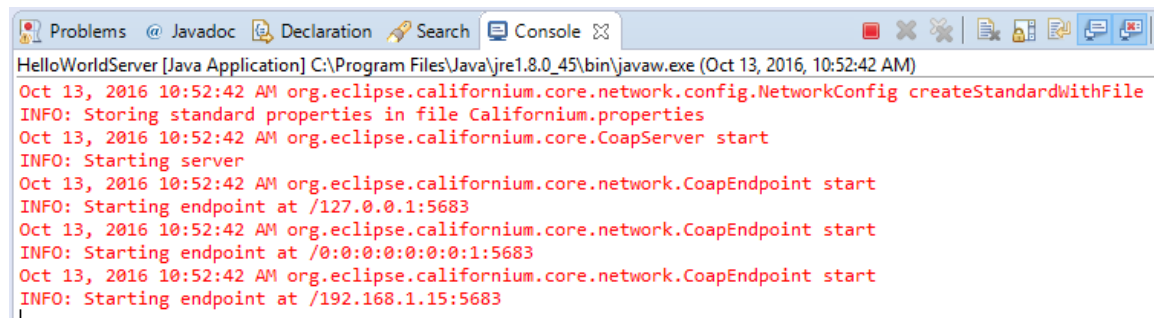
- > californium-core
- > californium-osgi
- > californium-proxy
- > cf-benchmark
- > cf-benchmark-observe
- > cf-cocoa
- > cf-helloworld-client
- > cf-helloworld-server
- > cf-plugtest-checker
- > cf-plugtest-client
- > cf-plugtest-server
- > cf-proxy
- > cf-secure
- > demo-apps
- > demo-certs
- > element-connector

5. Run Hello World

Select the Hello World example, double click it, the code is already compiled automatically by Eclipse. Click Run -> Run, you will have a Coap server running on your computer.



You will see the output as the following:



To stop the server, click the **RED** square near the Console.

Basically, you will modify this example for your Project 2. Add sensor readings as “resources” in CoAP server. Refer to class “HelloWorldResource” in the hello-world-server example.

```
class HelloWorldResource extends CoapResource {  
  
    public HelloWorldResource() {  
        // set resource identifier  
        super("helloWorld");  
  
        // set display name  
        getAttributes().setTitle("Hello-World Resource");  
    }  
  
    @Override  
    public void handleGET(CoapExchange exchange) {  
        // respond to the request  
        exchange.respond("Hello World!");  
    }  
}
```

You can also check the examples from the following Github repository:

<https://github.com/jvermillard/hands-on-coap>

7. Install Copper user-agent add-ons on Firefox.

You must have Firefox browser installed to run this CoAP client. Open your Firefox browser, and load the following link from Firefox:

<https://addons.mozilla.org/en-US/firefox/addon/copper-270430/>



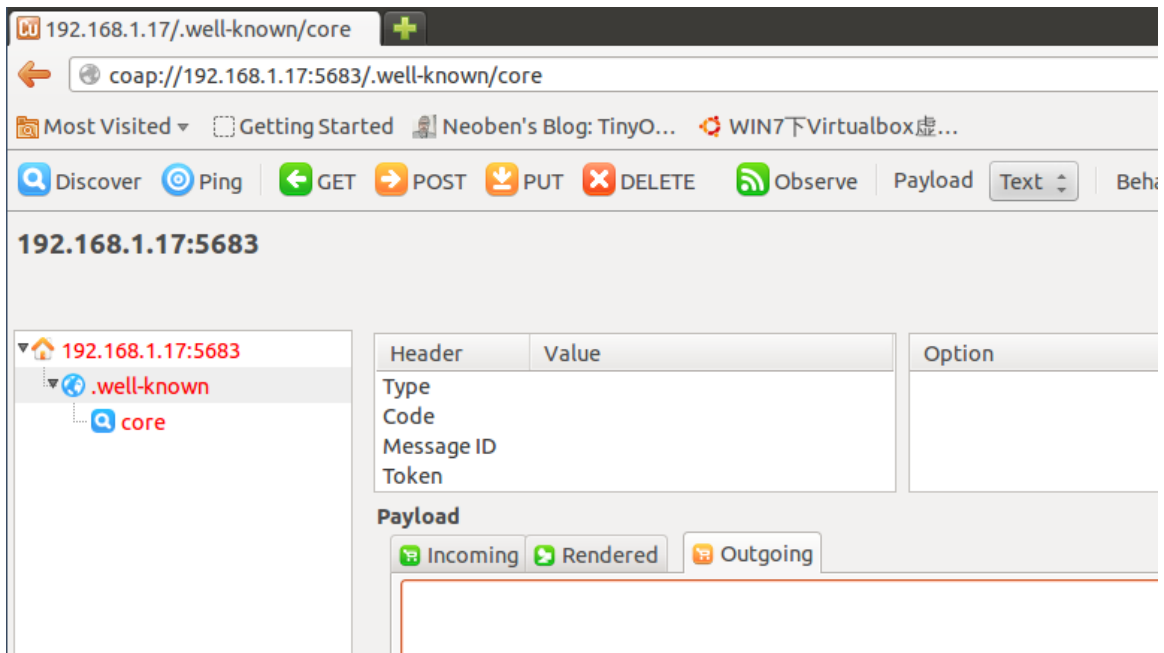
The screenshot shows the Firefox Add-ons page for the 'Copper (Cu)' add-on. On the left is an orange square icon with the letters 'Cu' in white. To the right of the icon, the text reads 'Copper (Cu) 0.18.4.1-signed' in a bold, serif font, followed by 'by Matthias Kovatsch' in a smaller, blue, sans-serif font. Below this, a paragraph describes the add-on: 'The Copper (Cu) CoAP user-agent for Firefox installs a handler for the 'coap' URI scheme and allows users to browse and interact with Internet of Things devices.' A prominent yellow button with a black plus sign and the text '+ Add to Firefox' is centered below the description. At the bottom, a small line of text states 'This add-on has been preliminarily reviewed by Mozilla. [Learn more](#)'.


Then simply click the “Add to Firefox” button to install Copper.

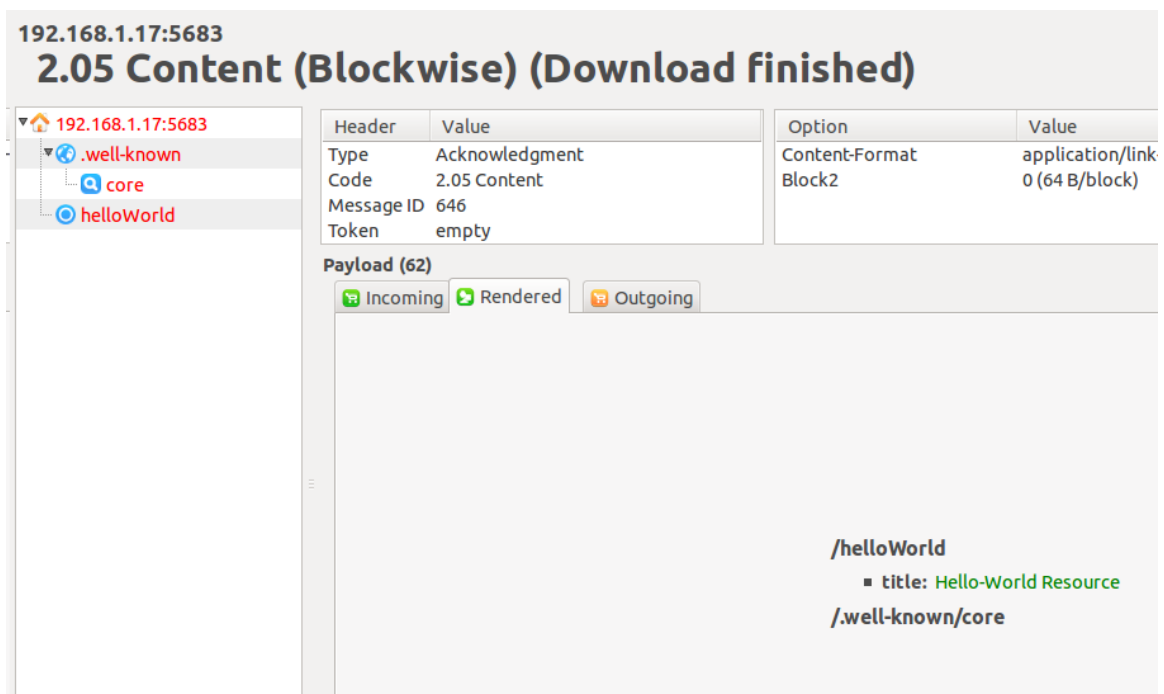
8. Access the “HelloWorld” server through Copper


Type the following CoAP address to access the “HelloWorld” server. Suppose the ip address of your computer is 192.168.1.17:

```
coap://192.168.1.17:5683/
```



Click “core”, then click , you will get all the **resources** on the server. In this example, the resource name is “helloworld”:



Then click “helloworld” and , you will get the response which is sent back by the “helloworld” resource, which is the string “Hello World!”:

192.168.1.17:5683

2.05 Content (Blockwise) (Download finished)

▼ 192.168.1.17:5683

• .well-known

• core

• helloWorld

Header	Value
Type	Acknowledgment
Code	2.05 Content
Message ID	56503
Token	empty

Option
Content-Format
Block2

Payload (12)

Incoming

Rendered

Outgoing

Hello World!