

CoAP Tutorial for Raspberry Pi

For Python users, both the CoAPthon (<https://github.com/Tanganelli/CoAPthon>) and txThings (<https://github.com/mwasilak/txThings>) are usable Python implementations of CoAP. CoAPthon has better documentation and is easy to follow.

The following tutorial is for Java users. It shows how to run a CoAP server on Raspberry Pi, and view the CoAP resources through Copper, the CoAP user-agent.

1. Install Maven

Maven is used to manage the dependencies of complex projects, such as the CoAP implementation. Install Maven so that you don't need to manually deal with the dependencies of the library.

```
$ sudo apt-get install maven
```

After the installation, you may have two versions of java on your Raspberry Pi. Run the following command to set both the java JDK and java JRE to the same version:

```
$ update-java-alternatives -l
[jdk_name_1 (openjdk-6)] ## [path]
[jdk_name_2 (java-jdk-8)] ## [path]
```

This command list all java JDKs installed on the Raspberry Pi.

```
$ update-java-alternatives -s [jdk_name_2 (java-jdk-8)]
```

This command set the jdk to java version 1.8, the same as the default java JRE version on Raspberry Pi.

2. Download CoAP library

There are two ways to download CoAP core from github:

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

```
https://github.com/eclipse/californium
```

Then unzip it, use FileZilla or similar ssh file management software to transfer the library to your Raspberry Pi.

b) Or you can download it directly from the Raspberry Pi. From the terminal, run the following command to get a copy of the files (a clone of the repository):

```
$ cd [your_destination_folder]
$ git clone https://github.com/eclipse/californium.git myCoAP
```

3. Compile the CoAP library and examples:

This is the basic way of how you compile your CoAP projects. Enter the root of CoAP library:

```
$ cd myCoAP/demo-apps/cf-helloworld-server
$ sudo mvn clean install
```

All the CoAP projects will be compiled as .JAR file and locate in myCoAP/demo-apps/run/.

4. Run “Hello World” server application

Run the following command to start a CoAP server:

```
$ cd myCoAP/demo-apps/run
$ java -jar cf-helloworld-server-1.0.0-SNAPSHOT.jar
```

```
pi@raspberrypi /opt/myCoAP/run $ java -jar cf-helloworld-server-1.0.0-SNAPSHOT.jar
Oct 20, 2015 11:52:35 PM org.eclipse.californium.core.network.config.NetworkConfig createStandardWithFile
INFO: Loading standard properties from file Californium.properties
Oct 20, 2015 11:52:35 PM org.eclipse.californium.core.CoapServer start
INFO: Starting server
Oct 20, 2015 11:52:35 PM org.eclipse.californium.core.CoapServer start
INFO: No endpoints have been defined for server, setting up server endpoint on default port 5683
Oct 20, 2015 11:52:36 PM org.eclipse.californium.core.network.CoapEndpoint start
INFO: Starting endpoint at 0.0.0.0/0.0.0.0:5683
```

The source code of the “Hello World” is located at myCoAP/demo-apps/cf-helloworld-server.

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

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

Open your Firefox browser, and load the following link from Firefox:

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



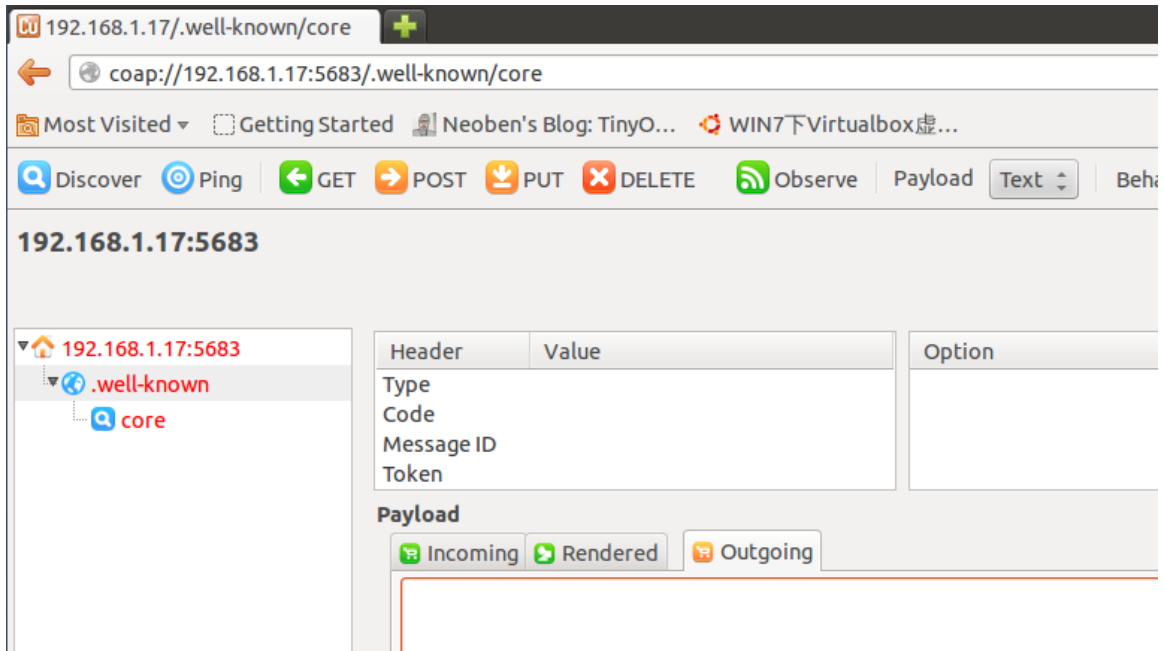
The screenshot shows the Firefox Add-ons page for the Copper (Cu) CoAP user-agent. It features a logo with the letters 'CU' in white on an orange square background. To the right of the logo, the text reads 'Copper (Cu) 0.18.4.1-signed' followed by 'by Matthias Kovatsch'. Below this, a description states: '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 plus sign and the text '+ Add to Firefox' is centered below the description. At the bottom of the card, a small line of text says 'This add-on has been preliminarily reviewed by Mozilla. [Learn more](#)'.


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

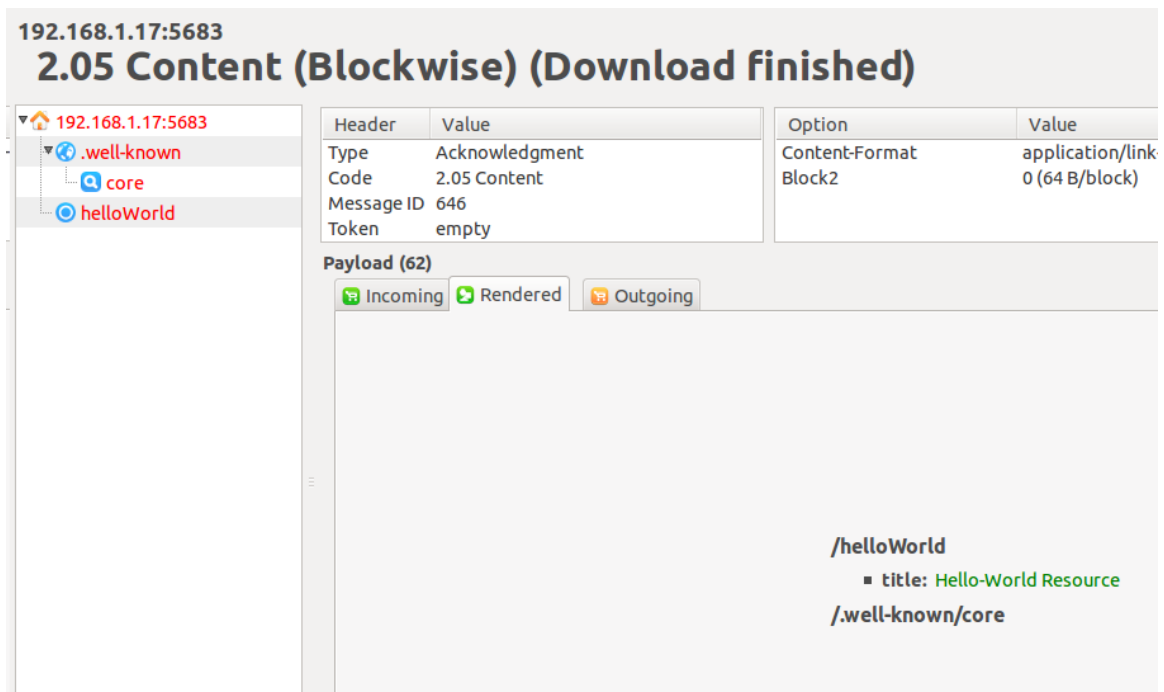
6. Access the “HelloWorld” server through Copper


Type the following CoAP address to access the “HelloWorld” server. Suppose the ip address of your Raspberry Pi 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)

Header	Value	Option
Type	Acknowledgment	Content-Format
Code	2.05 Content	Block2
Message ID	56503	
Token	empty	

Payload (12)

Incoming Rendered Outgoing

Hello World!

7. Include Pi4J dependency for your CoAP project using the POM.xml file

Edit the `/demo-apps/cf-helloworld-server/pom.xml` to include the Pi4J library. Add the following between the `<dependencies>` flag. Change the version according to the Pi4J version you use. For example, if you use "pi4j-1.1", then change the `<version>` to 1.1.

```
<dependency>
  <groupId>com.pi4j</groupId>
  <artifactId>pi4j-core</artifactId>
  <version>1.0</version>
</dependency>
```