Code explanation

```
package org.mo zil la.i ot.we bth ing.ex ample;
import org.js on.J SO NArray;
import org.js on.J SO NOb ject;
import org.mo zil la.i ot.we bth ing.Ac tion;
import org.mo zil la.i ot.we bth ing.Event;
import org.mo zil la.i ot.we bth ing.Pr operty;
import org.mo zil la.i ot.we bth ing.Thing;
import org.mo zil la.i ot.we bth ing.Value;
import org.mo zil la.i ot.we bth ing.We bTh ing Server;
import org.mo zil la.i ot.we bth ing.er ror s.P rop ert yError;
import java.i o.I OEx cep tion;
import java.u til.Ar rays;
import java.u til.UUID;
//Decl aration of class
public class Single Thing {
        //C reates an object of class Thing which used to control a lamp
        public static Thing makeTh ing() {
                Thing thing = new Thing( " urn :de v:o ps: my- lam p-1 234 ",
                                                        "My Lamp",
                                                       new JSONAr ray (Ar ray s.a sLi st( " OnO -
ffS wit ch",
                                                                                                -
              " Lig ht")),
                                                        "A web connected lamp");
                //C reates a json structure
                JSO NObject onDesc ription = new JSONOb ject();
                //Adds entries to the json structure
                onD esc rip tio n.p ut( " @ty pe", " OnO ffP rop ert y");
                onD esc rip tio n.p ut( " tit le", " On/ Off ");
                onD esc rip tio n.p ut( " typ e", " boo lea n");
                onD esc rip tio n.p ut( " des cri pti on", " Whether the lamp is turned on");
                //Adds the json structure to thing object as a property
                thi ng.a dd Pro per ty(new Proper ty( thing,
                                                                        " on",
                                                                      new Value( true),
                                                                       onD esc rip tion));
                  //C reates a json structure
                JSO NObject bright nes sDe scr iption = new JSONOb ject();
                //Adds entries to the json structure
                bri ght nes sDe scr ipt ion.pu t("@ typ e", " Bri ght nes sPr ope rty ");
                bri ght nes sDe scr ipt ion.pu t("t itl e", " Bri ght nes s");
                bri ght nes sDe scr ipt ion.pu t("t ype ", " int ege r");
                bri ght nes sDe scr ipt ion.pu t("d esc rip tio n",
                                                                "The level of light from 0-100");
```


Sponsored by Readable.com
Measure your website readability!
https://readable.com

Code explanation (cont)

```
>       brightnessDescription.put("minimum", 0);
      brightnessDescription.put("maximum", 100);
      brightnessDescription.put("unit", "percent");
      //Adds the json structure to thing object as a property
      thing.addProperty(new Property(thing,
                                     "brightness",
                                     new Value(50),
                                     brightnessDescription));
       //Creates a json structure
      JSONObject fadeMetadata = new JSONObject();
      JSONObject fadeInput = new JSONObject();
      JSONObject fadeProperties = new JSONObject();
      JSONObject fadeBrightness = new JSONObject();
      JSONObject fadeDuration = new JSONObject();
      //Adds entries to the json structure
      fadeMetadata.put("title", "Fade");
      fadeMetadata.put("description", "Fade the lamp to a given level");
      fadeInput.put("type", "object");
      fadeInput.put("required",
                    new JSONArray(Arrays.asList("brightness", "duration")));
      fadeBrightness.put("type", "integer");
      fadeBrightness.put("minimum", 0);
      fadeBrightness.put("maximum", 100);
      fadeBrightness.put("unit", "percent");
      fadeDuration.put("type", "integer");
      fadeDuration.put("minimum", 1);
      fadeDuration.put("unit", "milliseconds");
      fadeProperties.put("brightness", fadeBrightness);
      fadeProperties.put("duration", fadeDuration);
      fadeInput.put("properties", fadeProperties);
      fadeMetadata.put("input", fadeInput);
      //Adds the json structure to thing object as an action
      thing.addAvailableAction("fade", fadeMetadata, FadeAction.class);
      JSONObject overheatedMetadata = new JSONObject();
      overheatedMetadata.put("description",
                             "The lamp has exceeded its safe operating temperature");
      overheatedMetadata.put("type", "number");
      overheatedMetadata.put("unit", "degree celsius");
      thing.addAvailableEvent("overheated", overheatedMetadata);
      //returns the thing object with the added project
      return thing;
    }
```


Sponsored by Readable.com
Measure your website readability!
https://readable.com

Code explanation (cont)

```
>     //Startpoint for the program
    public static void main(String[] args) {
        //Creates an object of thing and the needed properties are added
        Thing thing = makeThing();
        WebThingServer server;
        try {
            // If adding more than one thing, use MultipleThings() with a name.
            // In the single thing case, the thing's name will be broadcast.
            server = new WebThingServer(new WebThingServer.SingleThing(thing),
                                        8888);
            //Hook is added so the program can be shut down
            Runtime.getRuntime().addShutdownHook(new Thread() {
                public void run() {
                    server.stop();
                }
            });
            //Starts the server and if something unexpected happens it prints the error and then exits
              the program
            server.start(false);
        } catch (IOException e) {
            System.out.println(e);
            System.exit(1);
        }
    }
```

Comments are in code

Packet Tracer

1) Set up the wired network with connecting FashEthernet0 on PC to Home Switch
2) Set up the wireless network with DHCP on the Home Gateway
3) Connect IoT devices to the network; Connect the Coffee Pot with FastEthernet0 by DHCP to the existing network
4) To access IoT remotely it needs a server address, user name and password.
5) IoT can be accessed remotely with a web browser, the address, user name and password

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com