Observing

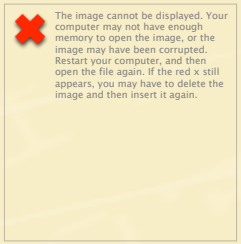
Giacomo Tanganelli
PhD student @ University of Pisa
g.tanganelli@iet.unipi.it

Contiki - Observing

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- The Contiki implementation is resource centric.
- To enable observing on a resource, the resource itself must be observable.
- Contiki defines also periodic resources. The handler is called periodically.
- Two type of observable resources:
  - EVENT_RESOURCE
  - PERIODIC_RESOURCE

Event resources



- Define the resource:
  - EVENT_RESOURCE(resource_example, "title=\"Resource\";rt=\"Text\”;obs", get_handler, NULL, NULL, NULL, event_handler);
- Define the standard handler:
  - get_handler(…)
  - Responses to GET requests
- Define the event handler:
  - void event_handler(…)
  - Response to events

Event handler

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

```
void event_handler(){
/* Do the update triggered by the event here, e.g., sampling a sensor. */

   /* Notify the registered observers which will trigger the tget_handler to create the response. */
   REST.notify_subscribers(&resource_example);
}
```

Activate event resource

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

```
rest_activate_resource(&resource_example, “path”);

 while(1) {
   PROCESS_WAIT_EVENT();
   if (ev == sensors_event && data == &button_sensor) {
        resource_example.trigger();
  }
}
```

Periodic resources

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- Define the resource:
  - PERIODIC_RESOURCE(resource_per, "title=\"Resource\";rt=\"Text\”;obs", per_get_handler, NULL, NULL, NULL,1000 * CLOCK_SECOND, per_handler);
- Define the standard handler:
  - per_get_handler(…)
  - Responses to GET requests
- Define the periodic handler:
  - void per_handler(…)
  - Response periodically

Periodic handler

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

```
void per_handler(){
/* Do the update triggered by the event here, e.g., sampling a sensor. */

   /* Notify the registered observers which will trigger the tget_handler to create the response. */
   REST.notify_subscribers(&resource_example);
}
```

Activate periodic resource

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

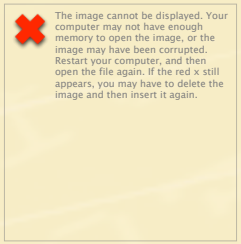
```
rest_activate_resource(&resource_per, “path”);

 while(1) {
   PROCESS_WAIT_EVENT();
}
```

Exercise 1

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- Write a CoAP server with two resources:
  - The former must be an event resource which will send a notification to observers when the server button is pressed.
  - The latter is a periodic resource which sends a notification every 10 seconds.

Content-type



- In order to exchange information correctly CoAP defines Content-type.
- A client can set an Accept option in a request.
- The server try to response with an accepted content-type.
- To retrieve accept options:
  - unsigned int accept = -1;
  - REST.get_header_accept(request, &accept);

Content-type (2)

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- Change the payload according to the Accept option:

```
 if(accept == -1 || accept == REST.type.TEXT_PLAIN) {
        /*Popolate response payload*/
        /*Set Response content-type*/
        REST.set_header_content_type(response, REST.type.TEXT_PLAIN);
} else if(accept == REST.type.APPLICATION_XML) {
        /*Popolate response payload*/
        /*Set Response content-type*/
        REST.set_header_content_type(response,
REST.type.APPLICATION_XML);
}
```

Exercise 2

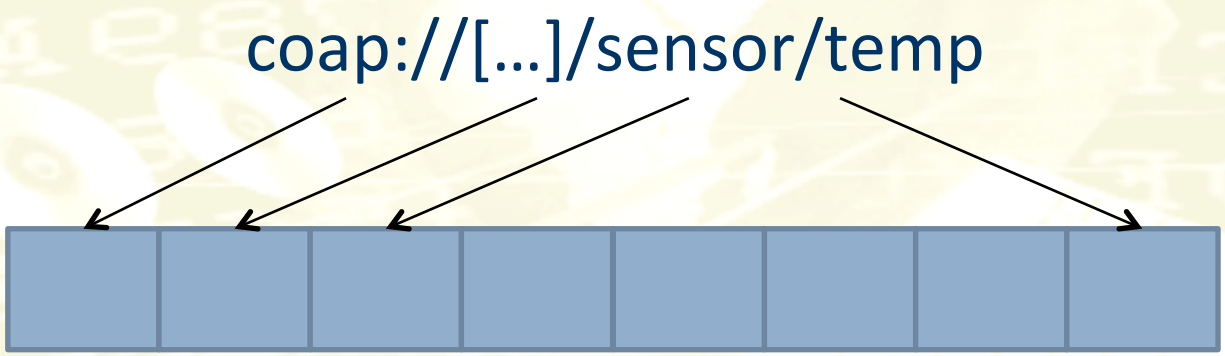
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- Modify the previous example in order to send responses in the xml format and text plain.
- Remember to not accept different type of enconding :
  - REST.set_response_status(response, REST.status.NOT_ACCEPTABLE);

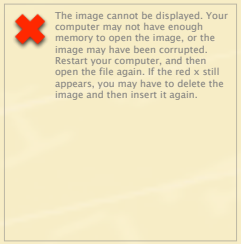
ETag

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- The Etag is used to validate responses:
  - A resource representation is characterized by an Etag.

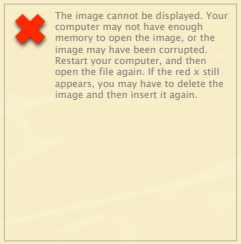


ETag



- When a client obtains a response it also obtains an Etag.
- When the client wants an update send a request with also the Etag obtained previously.
- If the resource representation is still valid the server reply with 2.03 valid without include in the payload the resource representation (the client already has the most updated one)

Contiki ETag



- To retrive the Etag contained in a request:
  - const uint8_t *bytes = NULL;
  - len = coap_get_header_etag(request, &bytes);
- To set an Etag:
  - REST.set_header_etag(response, etag, etag_len);
- To set 2.03 code:
  - REST.set_response_status(response, REST.status.NOT_MODIFIED);

Exercise 3

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- Defines a resource which reply to GETs with also an ETag. If the request has the correct Etag the server must reply wit 2.03.
- When the client sends a POST message the resource must be updated with the content of the request and the Etag must change according to the incoming Etag .