

RELAZIONE PROGETTO OPEN DATA

Federico Frasca Polara W82/000082

Parte 1) Generazione di un file .ics da un insieme di dati in RDF.

La prima parte del progetto si è basata sull'aggiunta di una nuova funzionalità al progetto Agenda Unica, che permette di generare e scaricare un file .ics contenente tutti gli eventi.

1.1) iCalendar

iCalendar è un formato di file che permette agli utenti di poter inviare richieste di riunione e di altre attività ad altri utenti, tramite e-mail, o la condivisione di file con estensione .ics. I destinatari del file, con opportuni software di supporto, possono anche rispondere facilmente al mittente e contro-proporre un'altra data e ora per l'attività.

I file iCalendar sono formati da normale testo, hanno un proprio MIME: *text/calendar* e l'ultimo standard per il formato dei dati è stato definito nel 2009 dal RFC 5545. Quest'ultimo è stato successivamente esteso con vari aggiornamenti fino al 2015 con RFC 7265. Anche se lo standard è molto completo le vari applicazioni che lo utilizzano come : *Google Calendar, Apple Calendario, Microsoft Outlook, Yahoo! Calendario, ecc.* lo supportano solo in parte, queste però sono ancora in continuo sviluppo e sicuramente diventeranno sempre più compatibili con lo standard.

1.2) Esempio di file Calendar

```
BEGIN:VCALENDAR
PRODID:-//Opendatahacklab Catania//Agenda Unica//IT
VERSION:2.0
CALSCALE:GREGORIAN
BEGIN:VEVENT
DTSTART:20160827T090000Z
DTEND:20160827T090000Z
DTSTAMP:20160828T174851Z
UID:http://dydra.com/free-agenda/20160827/CACCIA+AL+TESORO+CULTURALE/
LOCATION:, Via Torre del Vescovo, Catania, Italy
STATUS:CONFIRMED
SUMMARY:CACCIA AL TESORO CULTURALE
END:VEVENT
END:VCALENDAR
```

La prima linea deve essere `BEGIN:VCALENDAR` e l'ultima `END:VCALENDAR` che definiscono l'inizio e la fine del file, il contenuto tra le due righe è chiamato "icalbody", le righe successive aggiungono ulteriori informazioni al file come la versione (`VERSION:2.0`) del formato iCalendar, l'id dell'autore del iCalendar (`PRODID:`) e il formato da usare per le date del calendario (`CALSCALE:GREGORIAN`).

Per descrivere un nuovo evento bisogna aggiungere la riga `BEGIN:VEVENT` che identifica l'inizio delle informazioni dell'evento per poi chiudere con `END:VEVENT`.

Nelle specifiche sono state pensate molte informazioni aggiuntive per gli eventi tra le più usate abbiamo le date di inizio (`DTSTART:`) e di fine (`DTEND:`) evento, un timestamp che indica l'ultima modifica fatta all'evento (`DTSTAMP:`), l'UID che deve identificare in modo univoco un evento, la locazione (`LOCATION:`) e il (`SUMMARY:`) usato come titolo dell'evento.

1.3) Implementazione su Agenda Unica

Librerie Usate:

sparqllib.php: Usata per eseguire le query SPARQL su dati rappresentati in RDF.

calendar.php: Piccola libreria scritta appositamente per l'applicazione, per la generazione di file .ics.

```
<?php
class Calendar {
    var $data;
    var $name;

    function Calendar($name) {
        $this->name=$name;
        $this->data = "BEGIN:VCALENDAR\r\nPRODID:-//Opendatahacklab
                    Catania//Agenda Unica//IT\r\nVERSION:2.0\r\n
                    CALSCALE:GREGORIAN\r\nMETHOD:PUBLISH\r\n";
    }

    //FUNZIONE PER AGGIUNGERE EVENTI
    function add_event($start,$end,$uid,$location,$summary) {

        $this->data .= "BEGIN:VEVENT\r\nDTSTART:"
            .date("Ymd\THis\Z",strtotime($start))."\r\nDTEND:"
            .date("Ymd\THis\Z",strtotime($end))."\r\nDTSTAMP:"
            .date("Ymd\THis\Z")."\r\nUID:"
            . $uid."\r\nDESCRIPTION:\r\nLOCATION:"
            . $location."\r\nSEQUENCE:0\r\nSTATUS:CONFIRMED\r\nSUMMARY:"
            . $summary."\r\nTRANSP:OPAQUE\r\nEND:VEVENT\r\n";
    }
}
```

```

        //FUNZIONE PER CHIUDERE IL FILE E SCARICARLO
        function show() {
            $this->data .= "END:VCALENDAR";
            header('Content-Disposition: attachment; filename="'
                . $this->name . '.ics"');
            Header('Content-Length: ' . strlen($this->data));
            Header('Connection: close');
            echo $this->data;
        }
    }

    ?>

```

La libreria è molto semplice ed è formata da una sola classe, le proprietà `$name` e `$data` definiscono solo il nome utilizzato per nominare il file .ics e il testo da inserire nel file. Il primo metodo **function** `Calendar($name)` che fa anche da costruttore, assegna il nome del calendario e aggiunge le prime informazioni necessarie al file alla proprietà `data`. Il metodo **function** `add_event($start,$end,$uid,$location,$summary)` serve per aggiungere un nuovo evento concatenando a `data` tutte le informazioni. Infine il metodo **function** `show()` chiude il calendario e aggiunge ulteriori informazioni all'header del file, come il nome e la lunghezza, e stamperà le informazioni di `data`, che il browser tradurrà nell'apertura di una finestra per lo scaricamento del file .ics.

Script finale:

RDFEvents2Calendar.php: script che dopo aver prelevato i dati da un RDF tramite una query in SPARQL creerà e aggiungerà gli eventi al file .ics.

```

<?php
header("Content-type:text/calendar");

require_once( "sparqllib.php" );
require_once( 'Calendar.php' );

//Mi collego al database e in caso di errore esco.
$db = sparql_connect( "http://dydra.com/cristianolongo/agenda-unica-ct/sparql" );

if( !$db ) {
    print $db->errno() . ": " . $db->error() . "\n";
    exit;
}

$db->ns( "event","http://purl.org/NET/c4dm/event.owl#" );
$db->ns( "locn","http://www.w3.org/ns/locn#" );
$db->ns( "time","http://www.w3.org/2006/time#" );
$db->ns( "dcterm","http://purl.org/dc/terms/" );
$db->ns( "foaf","http://xmlns.com/foaf/0.1/" );
$db->ns( "locn","http://www.w3.org/ns/locn#" );
$db->ns( "wgs84","http://www.w3.org/2003/01/geo/wgs84_pos#" );
$db->ns( "rdfs","http://www.w3.org/2000/01/rdf-schema#" );

```

```

$db->ns( "sioc","http://rdfs.org/sioc/ns#");
$db->ns( "dc","http://purl.org/dc/elements/1.1/");

//Imposto ed eseguo la query per estrarre tutti gli eventi, in caso di errore esco
$query = "SELECT DISTINCT ?item ?itemlabel ?timeStart ?address WHERE {
    ?item locn:location ?site .
    ?item rdfs:label ?itemlabel .
    ?item event:time ?t .
    ?t time:hasBeginning ?hasB .
    ?hasB time:inXSDDateTime ?timeStart .
    ?site locn:address ?a .
    ?a locn:fullAddress ?address .

} ORDER BY DESC(?timeStart) ?item";

$result = $db->query( $query );
if( !$result ) {
    print $db->errno() . ": " . $db->error(). "\n";
    exit;
}

//Creazione di un nuovo calendario
$calendar = new Calendar("agenda-unica-calendar");

//Aggiungo gli eventi al calendario
while( $row = $result->fetch_array() ) {
    $uid=rtrim(chunk_split( $row['item'], 65, "\r\n "),"\r\n ");
    $summary=rtrim(chunk_split( $row['itemlabel'], 65, "\r\n "),"\r\n ");
    $location=rtrim(chunk_split( $row['address'], 65, "\r\n "),"\r\n ");
    $calendar->add_event
($row['timeStart'],$row['timeStart'],$uid,$location,$summary);
}

//Chiusura e download del calendario
$calendar->show();

?>

```

Lo script dopo aver incluso le librerie necessarie preleva i dati dall' RDF, se la query va a buon fine viene creato un oggetto `$calendar = new Calendar()` al quale vengono aggiunti con un'iterazione i vari eventi tramite il metodo `add_event()`. Infine il calendario viene chiuso e scaricato tramite il metodo `show()`.

Parte 2) Filtraggio dei dati tramite date nella libreria sparql_suite.

In questa seconda parte sono state fatte delle modifiche alla libreria *sparql_suite* che permette di eseguire query a sparql endpoint, in particolare è stata modificata la funzione `function EventQueryProcessor()` aggiungendo la possibilità di filtrare i dati tramite un range di date.

Funzione `EventQueryProcessor()` iniziale:

```
function EventQueryProcessor(eventQueryProcessor, currentDate){
    this.query = "PREFIX locn:<http://www.w3.org/ns/locn#>\n"+
    "PREFIX wsg84:<http://www.w3.org/2003/01/geo/wgs84_pos#>\n"+
    "PREFIX foaf:<http://xmlns.com/foaf/0.1/>\n"+
    "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>\n"+
    "PREFIX event:<http://purl.org/NET/c4dm/event.owl#>\n"+
    "PREFIX time:<http://www.w3.org/2006/time#>\n"+
    "PREFIX sioc:<http://rdfs.org/sioc/ns#>\n"+
    "PREFIX dc:<http://purl.org/dc/elements/1.1/>\n";

    if (eventQueryProcessor.additionalPrefixes!=null)
        this.query+=eventQueryProcessor.additionalPrefixes+"\n";

    this.query+="SELECT DISTINCT ?item ?agent ?post ?eventPlace ?depiction
?itemlabel ?site ?logo ?timeStart ?address ?description ?homepage ?partname ?ptitle
?plabel ?pcreat WHERE {\n";

    if (eventQueryProcessor.additionalConstraints!=null)
        this.query+="\t"+eventQueryProcessor.additionalConstraints+" .\n";

    this.query+="\t?item locn:location ?site .\n"+
    "\t?item rdfs:label ?itemlabel .\n" +
    "\t?item event:time ?t .\n"+
    "\tOPTIONAL {?item event:agent ?agent} .\n"+
    "\tOPTIONAL {?site rdfs:label ?eventPlace} .\n"+ //Optional because the
event could not have a event place
    "\t?t time:hasBeginning ?hasB .\n"+
    "\t?hasB time:inXSDDateTime ?timeStart .\n"+
    "\t?site locn:address ?a .\n"+
    "\t?a locn:fullAddress ?address .\n"+
    "\tOPTIONAL {?item rdfs:comment ?description} .\n"+
    "\tOPTIONAL {?item foaf:homepage ?homepage} .\n"+
    "\tOPTIONAL {?agent rdfs:label ?partname} .\n"+
    "\tOPTIONAL {?item foaf:depiction ?depiction} .\n"+
    "\tOPTIONAL {?hasB time:xsdDateTime ?timeEnd} .\n"+
    "\tOPTIONAL {?item foaf:logo ?logo} .\n"+
    "\tOPTIONAL {?post sioc:about ?item} .\n"+
    "\t\t?post dc:title ?ptitle .\n"+
    "\t\t?post rdfs:label ?plabel .\n"+
    "\t\t?post sioc:has_creator ?pc .\n"+
    "\t\t?pc rdfs:label ?pcreat .\n"+
    "\t} .\n";
    ORDER BY DESC(?timeStart) ?item";
```

```

    this.event=null;
    this.processor=eventQueryProcessor;
    this.currentDate=currentDate ==null ? new Date() : currentDate;
    this.isNextEvent=false;
}

```

Funzione EventQueryProcessor() finale:

```

function EventQueryProcessor(eventQueryProcessor, currentDate, minDate, maxDate){
    this.query = "PREFIX locn:<http://www.w3.org/ns/locn#>\n"+
    "PREFIX wgs84:<http://www.w3.org/2003/01/geo/wgs84_pos#>\n"+
    "PREFIX foaf:<http://xmlns.com/foaf/0.1/>\n"+
    "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>\n"+
    "PREFIX event:<http://purl.org/NET/c4dm/event.owl#>\n"+
    "PREFIX time:<http://www.w3.org/2006/time#>\n"+
    "PREFIX sioc:<http://rdfs.org/sioc/ns#>\n"+
    "PREFIX dc:<http://purl.org/dc/elements/1.1/>\n";
    "PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>";

    if(minDate==null || minDate=='')
    {
        minDate='2000-01-01T00:00:00Z';
    }
    if(maxDate==null || maxDate=='')
    {
        maxDate='3000-01-01T00:00:00Z';
    }

    if (eventQueryProcessor.additionalPrefixes!=null)
        this.query+=eventQueryProcessor.additionalPrefixes+"\n";

    this.query+="SELECT DISTINCT ?item ?agent ?post ?eventPlace ?depiction
?itemlabel ?site ?logo ?timeStart ?address ?description ?homepage ?partname ?ptitle
?plabel ?pcreat WHERE {\n";

    if (eventQueryProcessor.additionalConstraints!=null)
        this.query+="\t"+eventQueryProcessor.additionalConstraints+" .\n";

    this.query+="\t?item locn:location ?site .\n"+
    "\t?item rdfs:label ?itemlabel .\n" +
    "\t?item event:time ?t .\n"+
    "\tOPTIONAL {?item event:agent ?agent} .\n"+
    "\tOPTIONAL {?site rdfs:label ?eventPlace} .\n"+ //Optional because the
event could not have a event place
    "\t?t time:hasBeginning ?hasB .\n"+
    "\t?hasB time:inXSDDateTime ?timeStart .\n"+
    "\t?site locn:address ?a .\n"+
    "\t?a locn:fullAddress ?address .\n"+
    "\tOPTIONAL {?item rdfs:comment ?description} .\n"+
    "\tOPTIONAL {?item foaf:homepage ?homepage} .\n"+
    "\tOPTIONAL {?agent rdfs:label ?partname} .\n"+
    "\tOPTIONAL {?item foaf:depiction ?depiction} .\n"+
    "\tOPTIONAL {?hasB time:xsdDateTime ?timeEnd} .\n"+
    "\tOPTIONAL {?item foaf:logo ?logo} .\n"+
    "\tOPTIONAL {?post sioc:about ?item .\n"+
    "\t\t?post dc:title ?ptitle .\n"+

```

```

"\t\t?post rdfs:label ?plabel .\n"+
"\t\t?post sioc:has_creator ?pc .\n"+
"\t\t?pc rdfs:label ?pcreat .\n"+
"\t} .\n"+
"FILTER (xsd:dateTime(?timeStart) > '"+minDate+"'^^xsd:dateTime &&
xsd:dateTime(?timeStart) < '"+maxDate+"'^^xsd:dateTime) .\n"+
"} ORDER BY DESC(?timeStart) ?item";

this.event=null;
this.processor=eventQueryProcessor;
this.currentDate=currentDate ==null ? new Date() : currentDate;
this.isNextEvent=false;
}

```

Le modifiche fatte riguardano innanzitutto gli argomenti della funzione aggiungendo sia una dataMin (data minima) e una dataMax(data massima), per la massima dinamicità della funzione è stato aggiunto un piccolo blocco di codice per gestire le eventualità che dataMin o dataMax non vengano passati, o ne venga passato uno solo.

```

if(minDate==null || minDate=='')
{
    minDate='2000-01-01T00:00:00Z';
}
if(maxDate==null || maxDate=='')
{
    maxDate='3000-01-01T00:00:00Z';
}

```

Infine è stata modifica la query per poter filtrare i dati in base alle date tramite l'aggiunta di:

```

FILTER (xsd:dateTime(?timeStart) > '"+minDate+"'^^xsd:dateTime &&
xsd:dateTime(?timeStart) < '"+maxDate+"'^^xsd:dateTime

```