

# Culture24 REST API Developer's Guide

Revision 1.4 – August 2013

System Simulation

<http://www.ssl.co.uk>



## TABLE OF CONTENTS

Document history.....	4
Introduction.....	5
Culture24 data model.....	5
API basics.....	5
Base URL.....	5
API key.....	5
REST.....	6
Some API terminology.....	6
API response.....	6
Format.....	6
Return values.....	7
Error handling.....	8
API error codes.....	8
Fetching data – GET requests.....	9
Fetching all the items in a set.....	9
Traversing the set.....	9
Limiting the item data returned.....	10
Including elements from related items.....	10
Sorting.....	10
Event date sorting.....	11
Distance sorting.....	11
Querying sets (searching).....	11
Free text search.....	11
Element-specific search.....	12
Relevance.....	12
Fetching an item by unique ID.....	13
Examples and common recipes.....	14
Set element reference.....	16
Events.....	16
Data elements.....	16
Link elements.....	17
Query elements.....	17
Sort elements.....	17
Venues.....	19
Data elements.....	19
Link elements.....	20
Query elements.....	20
Sort elements.....	21

## Document history

Revision	API version	Date	Notes
1.4	V1	2013-08-30	Updated list of sort elements to reflect changes arising during acceptance testing. Added note about empty elements sorting to end rather than front.
1.3	V1	2013-08-07	Changes for July enhancements (6891): Documented the new 'branch' query operator Expanded section on sorting Added section on relevance Added some examples for sorting and relevance Updated list of elements
1.2	v1	2013-06-25	Added note about sort order
1.1	v1	2013-06-05	Added the <code>position</code> query element and its <code>box</code> operator. Deprecated the <code>latitude &amp; longitude</code> query elements.
1.0	v1	2013-05-16	First issue.

## Introduction

Culture24 maintains a database of heritage institutions and events across the UK. The information held is updated directly by institutions via Culture24's Direct Data Entry (DDE) website.

A portion of the database is accessible via a URL-based "REST"-like API. This document describes how to access and use API.

Some familiarity with the HTTP protocol and with JSON is assumed.

The API version at the time of writing is v1.

If you want to dive straight in with some examples, skip to the section at the end.

## Culture24 data model

The data in the Culture24 database is partitioned into resource sets (items of different types). There are currently two sets available via the API:

- venues

- events

Events are always "owned" by a single venue, i.e. there is a one-to-many relationship between venues and events.

Events generally take place at the venue, but not necessarily (i.e. may have their own location information).

A single event may span a range of dates (e.g. an exhibition) and may have multiple occurrences at different dates or times.

## API basics

### Base URL

The API is hosted at the following URL:

```
http://www.culture24.org.uk/api/rest/v1/
```

All your URLs should begin with this.

### API key

Before you can use the API, you'll need Culture24 to issue you an API key. An API key is a 9-character alphanumeric string intended for use only by you.

You will always need to supply this key as a URL parameter, e.g.

```
?key=i8sS7p3PE
```

(This is just an example, not a valid key.)

Be aware that the examples throughout this document don't include the key.

## REST

A REST API uses HTTP requests to interact with a database. The HTTP request methods correspond to “CRUD” database operations:

HTTP method	Database operation
POST	Create
GET	Read
PUT	Update
DELETE	Delete

At the time of writing, the Culture24 REST API is a read-only API and supports only GET requests to retrieve data from the database.

If you are unable to phrase your request as a GET request (e.g. because platform/toolkit limitations mean you have to POST) you can include an explicit `method` URL parameter, e.g.

```
?method=GET
```

## Some API terminology

**Sets** (tables, record types) contain **items** (rows, records) which have **elements** (columns, fields).

## API response

### Format

By default, API return values are returned as a JSON-encoded object in the body of the HTTP response. However, the API also supports JSON-P and XML encodings.

Use the `format` URL parameter to explicitly request a particular format:

```
?format=json      (the default if omitted)
?format=jsonp&callback=yourFunction
?format=xml
```

The root element for an XML return object is `<return>`.

In XML responses, array values are returned simply as repeating elements. E.g.

JSON:

```
foo: [ "a", "b", "c" ]
```

XML:

```
<foo>a</foo>
<foo>b</foo>
<foo>c</foo>
```

Regardless of your choice of format, the HTTP Content-Type header will always be `text/html` to avoid issues with certain browsers.

## Return values

The response object always includes the following members:

success	Boolean (true or false)	Whether or not your API call has worked.
result	Object	The return value of the API call, if successful. The structure will depend on the particular call.

E.g.

Request:

```
/venues/AM40
```

Return object:

```
{
  success: true,
  result: {
    uniqueID: "AM40",
    name: "Culture24",
    ...
  }
}
```

If the `success` member is false, the `result` member will be an object containing the following members:

errorCode	Integer	A unique code for each kind of error the API can generate. (See "Error handling" below)
errorMessage	String	A description of the error which may reference the particulars of your request

E.g.

Request:

```
/venues/AM40?key=BADKEY
```

Return object:

```
{
  success: false,
  result: {
    errorCode: 101,
    errorMessage: "The supplied key is not valid ↵
                  for this API"
  }
}
```

## Error handling

You should first check the response's HTTP status code. This will be:

HTTP status code	What it means
500 Internal Server Error	There was a server error or misconfiguration. The body of the response will be an HTML page.
405 Method Not Allowed	You attempted something other than a GET request.
404 Not Found	<p>Possible causes are that your request URL</p> <ul style="list-style-type: none"> <li>was incomplete (e.g. missing the v1 version number)</li> <li>was incorrect (e.g. a typo, the wrong version number)</li> <li>referenced an unknown resource set (something other than 'venues' or 'events')</li> <li>referenced a non-existent item ID</li> </ul> <p>The body of the response will be an HTML page.</p>
400 Bad Request	You did something wrong in your API call. The <code>success</code> member of the response object should be false, and the <code>result</code> member will be an object containing error information as described in “Return values” above.
200 OK	Your API call completed. The <code>success</code> member of the response object should be true.

Ideally you should cope with redirection codes (302, 303) in case the API moves in the future. All other status codes should be treated as errors.

## API error codes

Code	What it means, and troubleshooting
101	<p><i>Missing or bad key</i></p> <p>Check you've supplied an API key, and make sure it's the one Culture24 issued you. If it all looks correct, get in touch with Culture24.</p>
102	<p><i>Unknown data element</i></p> <p>You've asked for a data element that doesn't exist. Check for typos, check you're querying the right set, and check against the list of elements in “Set element reference”. Note that element names are case-sensitive.</p>
103	<p><i>Unsupported response format</i></p> <p>You've asked for a <code>format</code> other than <code>json</code>, <code>jsonp</code>, or <code>xml</code>.</p>
104	<p><i>Missing JSON-P callback</i></p> <p>If you've specified <code>method=jsonp</code>, make sure you also specify a <code>callback</code>.</p>
105	<p><i>Bad query element</i></p> <p>You're trying to query into an unknown query element, or you're using an invalid operator for that query element. Check for typos, check you're querying the right set, make sure it's in the list in “Set element reference” and that it supports the operator you're using.</p>

106	<i>Bad argument to query element</i> In practice, this means you didn't supply a valid range to the <code>range</code> operator. Make sure you give a comma-separated pair of values.
107	<i>Attempt to query an item</i> If you've made a unique item request (e.g. <code>/venues/AM40</code> ), you can't include query parameters in the URL.
108	<i>Invalid result offset</i> You've supplied a negative <code>offset</code> , or you've exceeded the number of items in the query result. (Note that this could happen even if your maths is correct, because the size of the result could conceivably change across requests.)
109	<i>Invalid result limit</i> You've specified a negative <code>limit</code> . (Note that a limit of zero is ok, but will of course always yield no matches.)
110	<i>Bad sort</i> Either the specified sort element is not recognised, or it requires a value that hasn't been specified (e.g. a coordinate point, when sorting by distance)

## Fetching data – GET requests

### Fetching all the items in a set

The URL to fetch all the items in a set is

`/{set}/`

(Include the trailing slash – it's important.)

For example

`http://www.culture24.org.uk/api/rest/v1/venues/`

will fetch all the venue items in the database. Note that items are returned in batches (see “Traversing the set” below).

The response object's `result` member will be an object with the following members:

<code>found</code>	Integer	The total number of items in the set.
<code>items</code>	Array	The list of items in the batch. Each item is an object whose elements depend on the type of item – see “Set element reference” for reference.

### Traversing the set

Set items are returned in batches (of 10 by default). You can control the batch size, and the starting point in the set, using the following URL parameters, both of which are optional:

<code>offset</code>	Integer	The zero-based index into the set of the first item to be returned.	Default: 0 (i.e. start at first item)
<code>limit</code>	Integer	The maximum number of items to be returned.	Default: 10

E.g.

```
/venues/?offset=100&limit=50
```

(Note the / before the ? - it's important.)

### *Limiting the item data returned*

By default, the API will include all the available elements for each item returned. You may only be interested in certain elements, in which case you can use the `elements` URL parameter:

<code>elements</code>	Comma-separated list	Limits the elements returned for each item to those specified in the list
-----------------------	----------------------	---

E.g.

Request:

```
/venues/?elements=uniqueID,name,link
```

Item in response:

```
{
  uniqueID: "AM40",
  name: "Culture24",
  link: "http://www.culture24.org.uk/am40"
}
```

See "Set element reference" for the full set of elements available for each set.

### *Including elements from related items*

Items in one set may be related to items in another set. For example, events belong to a venue. When specifying the elements to return, you can pull in data from across such a relationship by using dot notation.

E.g.

Request:

```
/events/?elements=uniqueID,name,venue.name
```

Item in response:

```
{
  uniqueID: "EVENT423226",
  name: "Made in the Middle exhibition"
  venue: {
    name: "Northampton Museum & Art Gallery"
  }
}
```

This mechanism helps you avoid making multiple API requests to gather data from multiple sets.

Link elements (such as "venue" in the above example) are indicated in the "Set element reference" section of this document.

### *Sorting*

By default, events are returned in ascending date order. Where an event occurs on multiple dates, the date used for sort comparison is the nearest future date.

Venues are returned in ascending name order (i.e. A-Z).

Query results (see next section) are also returned in this order by default.

The sort order can be altered by specifying a `sort` URL parameter:

<code>sort</code>	Single element	Sorts the result on the named element. Note that not all elements are sortable, and there are some sort elements which are not data elements (e.g. <code>distance</code> )
-------------------	----------------	--

E.g.

`/events/?elements=uniqueID,name&sort=publicationDate`

The default search direction is ascending, but you can change it to descending by qualifying the sort parameter with `.desc`:

`/events/?elements=uniqueID,name&sort.desc=publicationDate`

(You can also use `.asc` if you want the default sort order to be explicit.)

The available sort elements are given in the appendix.

Note: we've generally taken pains to ensure empty elements sort to the end of an ascending sort rather than the front.

### Event date sorting

Events can be sorted by date using the `date` sort element. This sorts events by the nearest date to today.

`/events/?sort=date`

Where a date range has been specified as a query parameter, the earliest date in the range will be used in preference to today:

`/events/?q.date.range=01/11/2013,30/11/2013&sort=date`

This mechanism ensures that where events are limited to a particular date range, they are sorted by proximity to that date even if instances of the events are also occurring sooner.

### Distance sorting

Events and venues can also be sorted by distance from a point using the `distance` sort element. The point (a WGS84 lat/long pair) is specified using a parameter `s.distance`:

`/events/?sort=distance&s.distance=51.2,-1.03`

## Querying sets (searching)

You can filter sets by including query parameters in your request URL. The response object and traversal logic is the same as for a plain set request, except of course that you'll only be seeing the matching subset.

### Free text search

The URL parameter `q` will perform a free text search across all the elements in the set and return only matching items.

For example:

`/events/?q=workshop`

`/venues/?q=british+museum`

### Element-specific search

For each set there are a range of *query elements* which let you search into specific data. To use a query element, include it in your request URL with a prefix of “q.”.

E.g. to search for the venue with unique ID AM40

`/venues/?q.uniqueID=AM40`

The list of available query elements is given in the “Set element reference” section of this document. (Note that although these often share the same name as item elements, they aren't the same, and you can't therefore search into every item element.)

Some query elements support multiple *operators* which control how the query is interpreted. For example, the event *date* query element supports both “exact” searching (find events occurring on a specific date) and “range” searching (find events occurring between two comma-separated dates). The operator is appended to the query element in your request:

`/events/q.date.range=01/06/2013,30/06/2013`

The full set of operators is as follows (although consult the “Set element reference” section to see which query elements support which operators)

text	Perform a free text search into the element. For example, “york” would match “new york”
exact	Return only items where the value is found exactly. For example, “york” would match “York” but not “New York”
range	Return only items where the value is within the range specified (as a comma-separated pair)
box	Return only items within the bounding box (x1,y1) – (x2,y2) where (x1,y1) is the lower left corner of the box and (x2,y2) the upper right corner. Specify as a comma-separated list: x1, y1, x2, y2
branch	The values in some data elements are taken from a hierarchical authority. The branch operator requests that not only items with the query term be found, but also any items using terms below the query term in the hierarchy. For example, “London” sits below “England” in the tagging hierarchy, so while <code>q.contentTag.exact=England</code> would only find items tagged with “England”, <code>q.contentTag.branch=England</code> would also find items tagged with “London”.

The default operator for a query element (i.e. if you don't specify one explicitly) is “text” if the element supports it, otherwise “exact”, otherwise you must specify the operator.

### Relevance

Culture24 supports relevance calculation. The calculation is based on the number of hits for your query terms in each item, and the weightings of the database fields in which those hits are found. (For example, a hit in a title field is considered more relevant than a hit in a descriptive field.)

Relevance calculation is implicitly enabled if you include `relevance` as one of your requested data elements, or if you specify `relevance` as your sort.

The `relevance` data element yields the numeric weighting that was calculated for each item in the result, e.g.

Request:

```
/venues/?q.name=British+Museum&elements=name,relevance
```

Item in response:

```
{
  name: "British Museum",
  relevance: 40
}
```

## Fetching an item by unique ID

As well as querying a set to retrieve an item by its ID (using the `uniqueID` query element as in previous examples), you can use a plain URL to fetch the item:

```
/{set}/{id}
```

For example

```
http://www.culture24.org.uk/api/rest/v1/venues/AM40
```

IDs are not case sensitive, however we'd recommend you always use upper-case IDs.

As with set requests and queries, you can also use the `elements URL` parameter to control what item elements are returned (see "Limiting the item data returned").

If no item exists in the database for the given ID, or the ID corresponds to an item that's in a different set, a 404 Not Found response is returned. The body of the response is an HTML page, so don't attempt to parse it.

If the ID does exist, the `result` member of the response object will be the item, returned as an object, e.g.

Request:

```
/venues/AM40?elements=uniqueID,name,link
```

Response:

```
{
  success: true,
  result: {
    uniqueID: "AM40",
    name: "Culture24",
    link: "http://www.culture24.org.uk/am40"
  }
}
```

## Examples and common recipes

*Note: you'll need to append your API key to all of these examples (add `&key=yourkey` to the end).*

Fetching a venue by ID as a JSON object

```
http://www.culture24.org.uk/api/rest/v1/venues/AM40
```

Fetching an event by ID as a JSON object

```
http://www.culture24.org.uk/api/rest/v1/events/EVENT423226
```

Returning XML rather than JSON

```
http://www.culture24.org.uk/api/rest/v1/venues/AM40?format=xml
```

Returning JSON-P rather than JSON

```
http://www.culture24.org.uk/api/rest/v1/venues/AM40?format=jsonp&callback=processVenue
```

Returning specific venue information rather than everything

```
http://www.culture24.org.uk/api/rest/v1/venues/AM40?elements=name,description,link
```

Fetching all the events (first batch of 10 events)

```
http://www.culture24.org.uk/api/rest/v1/events/
```

Fetching the next batch of 10 events

```
http://www.culture24.org.uk/api/rest/v1/events/?offset=10
```

Fetching all the events (first batch of 50 events)

```
http://www.culture24.org.uk/api/rest/v1/events/?limit=50
```

Fetching the next batch of 50 events

```
http://www.culture24.org.uk/api/rest/v1/events/?limit=50&offset=50
```

Find events containing the word "sleepover"

```
http://www.culture24.org.uk/api/rest/v1/events/?q=sleepover
```

Find events containing the word "sleepover" in the title

```
http://www.culture24.org.uk/api/rest/v1/events/?q.name=sleepover
```

Find sleepover events on the 15th June 2013

```
http://www.culture24.org.uk/api/rest/v1/events/?q.name=sleepover&q.date=05/06/2013
```

Find all events in June 2013

```
http://www.culture24.org.uk/api/rest/v1/events/?q.date.range=01/06/2013,30/06/2013
```

Find all events in June 2013 sorted by date

```
http://www.culture24.org.uk/api/rest/v1/events/?q.date.range=01/06/2013,30/06/2013&sort=date
```

Include the venue name in event search results

<http://www.culture24.org.uk/api/rest/v1/events/?elements=uniqueID,name,venue.name>

Find events suitable for all ages

<http://www.culture24.org.uk/api/rest/v1/events/?q.audience.exact=All+ages>

Find events tagged with Museums at Night 2013

<http://www.culture24.org.uk/api/rest/v1/events/?q.contentTag.exact=Museums+at+Night+2013>

Find exhibitions (events of type Exhibition)

<http://www.culture24.org.uk/api/rest/v1/events/?q.type=Exhibition>

Find events relevant to “Cross curricular skills” in the National Curriculum

<http://www.culture24.org.uk/api/rest/v1/events/?q.nationalCurriculumTag.exact=Cross+curricular+skills>

Find all the events for the National Maritime Museum (SE000088)

<http://www.culture24.org.uk/api/rest/v1/events/?q.venueID=SE000088>

Find all the events occurring in a coordinate bounding box

*If (Lat1,Long1) is the lower left corner of the box, and (Lat2,Long2) the upper right corner*

<http://www.culture24.org.uk/api/rest/v1/events/?q.position.box=Lat1,Long1,Lat2,Long2>

Find venues containing the words “british museum”, sorted by relevance

<http://www.culture24.org.uk/api/rest/v1/venues/?q=british+museum&sort=relevance>

## Set element reference

### Events

#### Data elements

Element	List?	Type	Notes
uniqueID	No	string	Culture24's unique ID for the event.
type	Yes	string	Taken from a list of event types, e.g. "Exhibition"
link	No	URL (string)	The URL for the event on the Culture24 website.
publicationDate	No	string	The date the event was added to the Culture24 database.
graphicUrl	No	URL (string)	The URL of a Culture24 thumbnail for this event. Not all events have an image.
name	No	string	The title of the event.
description	No	string	A description of the event.
shortDescription	No	string	A short (max 200 character) description of the event.
venueID	No	string	The venue that created this event.
instance	Yes	object: startDate startTime endDate endTime	The date/s and times on which the event is taking place. There may be multiple instances of an event, e.g. a workshop occurring over four Wednesdays might be a single event with four instances.
url	No	URL (string)	The URL for the event on the venue's website. Note that these URLs may not always include the http:// protocol portion.
audience	Yes	string	The intended audience for the event, taken from a list of audience types, e.g. "All ages".
language	No	string	Only if the event involves a particular language or languages.
charges	No	string	Free text indicating any admission charges for the event.
freeOfCharge	No	Y/N (string)	Y, N or blank
concessionsAvailable	No	Y/N (string)	Y, N or blank
registrationRequired	No	Y/N (string)	Y, N or blank
status	No	string	One of Cancelled, Fully booked, Planning, Confirmed
place	No	string	The name of the place at which the event is taking place.
nationalCurriculumTag	Yes	string	Tags corresponding to National Curriculum subjects, taken from a National Curriculum hierarchy.
contentTag	Yes	string	Tags applied to the event, taken from Culture24 tagging lists.
addressStreet	No	string	The address at which the event is taking place. (This may not be the event's venue's address.)

addressTown	No	string	
addressCounty	No	string	
addressPostcode	No	string	
addressCountry	No	string	
relevance	No	positive integer	The relevance weighting computed for this item based on your query parameters.

### Link elements

Element	Relation	Notes
venue	N:1	The venue that created the event, or where the event is taking place.

### Query elements

Unless otherwise specified, query elements search into the data elements of the same name.

Element	Supported operators	Notes
uniqueID	exact	Case insensitive
contentTag	text, exact, branch	
nationalCurriculumTag	exact, branch	
date	exact, range	Individual events may occur across a range of dates. In this context, “exact” means you are supplying a single date, and the system will match any events that are taking place on that date. “range” means you are supplying two dates, and the system will return any events within or overlapping these dates.
name	text	Free text search into the event name only
venueID	exact	Case insensitive
type	Text, branch	
audience	text, exact	
position	box	Use for a lat/long bounding box search (see the examples)
latitude <i>deprecated</i>	range	Combine a latitude and longitude range query to do a bounding box search. <i>Deprecated – use position instead.</i>
longitude <i>deprecated</i>	range	

### Sort elements

Element	Notes
publicationDate	The date the item was added to the Culture24 database.
relevance	The weighting of the item based on your query terms.
name	The title of the event.

addressCounty	The county in which the event is taking place.
place	The name of the location at which the event is taking place.
date	The date on which the event is taking place. See “Date sorting” above.
distance	The distance of the venue from a supplied point. See “Distance sorting” above.

## Venues

### Data elements

Element	List?	Type	Notes
uniqueID	No	string	Culture24's unique ID for the venue.
type	Yes	string	Taken from a list of venue types, e.g. "Museum", "Castle or defences", "Science centre".
link	No	URL (string)	The URL for the venue on the Culture24 website.
publicationDate	No	string	The date the venue was added to the Culture24 database.
graphicUrl	No	URL (string)	The URL of a Culture24 thumbnail for the venue. Not all venues have an image.
name	No	string	The name of the venue, e.g. "Victoria and Albert Museum".
otherNames	Yes	string	Common alternative names by which the venue is known, including misspellings. E.g. "V&A"
description	No	string	A (potentially lengthy description of the venue.
legalStatus	No	string	Taken from a Culture24 list. E.g. "Charitable", "Armed Service", "Commercial"
url	Yes	object: qualifier url description	URLs on the venue's website. Each URL is qualified with a qualifier taken from a Culture24 list, e.g. "Shop", and has an optional free text description. Note that URLs may not always include the http:// protocol portion.
email	Yes	object: address description	The venue's email contact email addresses. Each address may have a free text description, e.g. "General enquires", "Venue hire".
telephone	Yes	object: number description	The venue's contact phone numbers. Each number may have a free text description, e.g. "School visit enquiries"
fax	Yes	object: number description	The venue's fax numbers. Each may have a free text description.
addressStreet	No	string	The venue's address.
addressTown	No	string	
addressCounty	No	string	
addressPostcode	No	string	
addressCountry	No	string	
latitude	No	string	WGS84 latitude (as a string rather than a floating point number).

longitude	No	string	WGS84 longitude.
openingHours	No	string	Free text opening hours information.
charges	No	string	Free text pricing information.
discounts	Yes	string	A list of accepted discount schemes, taken from a Culture24 list.
travelDirections	No	string	Free text directions on how to get to the venue.
facilitiesInformation	No	string	Free text description of facilities at the venue.
facility	Yes	string	A list of specific facilities, taken from a Culture24 list.
service	Yes	string	A list of services offered by the venue, taken from a Culture24 list.
keyArtistOrExhibit	No	string	A free text description of the venue's key attractions.
collections	Yes	string	A list of collection types held by the venue, taken from a Culture24 list.
collectionsDescription	No	string	A free text description of the collections held by the venue.
collectionsDescriptionGraphic	No	URL (string)	The URL of a Culture24 thumbnail for the venue's collections.
constituentInstitution	Yes	string	Some venues are actually umbrella organisations for others, e.g. National Museums of Liverpool. This is a list of the other venue IDs represented by this venue.
lea	No	string	Local Education Authority, taken from a Culture24 list.
regionalAgency	No	string	The regional agency responsible for the venue.
contentTag	Yes	string	Tags applied to the event, taken from Culture24 tagging lists.
relevance	No	positive integer	The relevance weighting computed for this item based on your query parameters.

### Link elements

There are no link elements for venues.

See the examples for how to find all the events for a particular venue.

### Query elements

*Unless otherwise specified, query elements search into the data elements of the same name.*

Element	Supported operators	Notes
uniqueID	exact	Case insensitive

contentTag	text, exact, branch	
name	text	Free text search into the venue name and alternative names fields.
position	box	Use for a lat/long bounding box search (see the examples)
latitude <i>deprecated</i>	range	Combine a latitude and longitude range query to do a bounding box search. <i>Deprecated – use position instead.</i>
longitude <i>deprecated</i>	range	

### Sort elements

Element	Notes
publicationDate	The date the item was added to the Culture24 database.
relevance	The weighting of the item based on your query terms.
name	The name of the venue.
distance	The distance of the venue from a supplied point. See “Distance sorting” above.