



Proposal: Model Capacity to Take Referrals

Authors	Matt Marshall (Open Data Services) Kate Lambacher (KCL Software Solutions)
Document Version	2024-08-28

Summary

We propose adding new schemas, `unit` and `service_capacity`, to HSDS in order to model service capacities where needed. We also propose a new optional field `service.capacities` to link service records with capacity models. These fields address real-world needs around providing capacity information within HSDS compliant systems.

This represents new optional features in HSDS which would result in a MINOR upgrade to the specification of implemented.

Background and Motivation

The problem we're trying to solve

The HSDS `service` schema provides the `status` field which may be one of the following values:

- `active`
- `inactive`
- `defunct`
- `temporarily closed`

This informs humans and digital systems whether the service is currently active and taking referrals, but does not provide any information about the current capacity levels of the service.

Capacity information is important to service provision because it allows services and referrers to understand nuance in the availability of the service. A service may be experiencing heavy load but still has capacity for a few more referrals in a given time-frame. In other circumstances, a service may have multiple tiers of capacity for different types of service users and thus may be in a state of both availability and unavailability.

This capacity information is important for humans and the digital systems because it allows them to make more informed decisions about if, and how, to present information about the service to a service user or a referrer.

The benefits of the proposed update

We believe that adding schemas and fields to model service capacity provides the following immediate benefits:

- It provides a generic reusable model of a unit which can be used elsewhere in the specification where appropriate
- In certain datasets, it allows direct capacity comparison between different services and providers where they share the same units
- It becomes possible to model the capacity of a service across multiple different types of capacity per service
- It becomes possible to display the capacity of a service via an interface, allowing humans to make use of this information
- It allows capacity information to be recorded over time if changes are stored elsewhere, meaning that predictive models may be used to estimate when services may experience influxes or conversely when they may become available again

We believe that there are also secondary and tertiary benefits to modeling service capacity which result from making this data available to human analysts such as researchers. These include allowing them some measures by which to gauge service demand under various circumstances, to plan for future service provision or funding, and supporting interest groups evidencing a need for more service provision where needed.

The evidence for the need

Implementers across different national contexts have expressed a desire to be able to model service capacity in some way within HSDS.

Implementers in the UK¹ have noted that they have had to consider inelegant workarounds in API feeds which “switch [services] on and off according to their capacity”. While `service.status` may be used to report on the overall status of the service this does not provide the necessary detail for understanding the actual capacity of the service at a more useful, granular, level.

Proposal Details

We propose the creation of some new schemas and a new optional field on the `service` schema, in order to provide a generic model for service capacity in HSDS.

We propose adding a new `unit` schema to model an individual unit which a service provides e.g. beds, or meals. The `unit` schema allows a service to define *what* they are providing. Where applicable and permitted by source data, service providers can also use `unit` to provide details of the standardized schemes which define the unit. This supports machines and digital systems interpreting the data.

We propose creating a `service_capacity` schema to model the capacity of a service. This new schema makes use of the `unit` schema to define *what* the service provides and includes more fields which define *how much* is available. We have proposed a **minimal model** for `service_capacity` to begin with, to accommodate future changes such as eligibility when they are more understood.

Finally, we propose adding `service.capacities` as a new field which is an array of `service_capacity` objects. This creates a 1-to-many relationship between `service` and `service_capacity`, allowing a service record to define multiple forms of capacity with human-readable and machine-readable information.

Schema changes

We propose the following changes to the HSDS Schemas:

- Creation of a new `unit` schema
- Creation of a new `service_capacity` schema

¹ <https://github.com/openreferral/specification/issues/221>

- Creation of a new field `service.capacities` defined as an array of `service_capacity` objects on the `service` schema

Proposed `unit` schema

These changes result in the creation of a new schema.

We propose the creation of a `unit` schema with the following properties.

Field	Title	Description	Type	Required? ²
<code>id</code>	Identifier	The identifier for the unit object. Each unit must have a unique identifier.	String (uuid)	True
<code>name</code>	Name	The human-readable name for this unit e.g. "Bed" or "Hours"	String	True
<code>scheme</code>	Scheme	The scheme which formalizes the unit, if applicable e.g. "SI" for Standard International Units such as Kilogram, Litre, etc.	String	False
<code>identifier</code>	Unit Identifier	The identifier of the unit taken from the scheme if applicable e.g. "kg"	String	False
<code>uri</code>	URI	The URI to the definition of the unit, if applicable	String (uri)	False

Proposed `service_capacity` schema

These changes result in the creation of a new schema.

We propose the creation of a `service_capacity` schema with the following properties.

Field	Title	Description	Type	Required? ³
<code>id</code>	Identifier	The identifier for the <code>service_capacity</code> object. Each	String (uuid)	True

² In the scope of the `unit` schema only

³ In the scope of the `service_capacity` schema only

		service_capacity must have a unique identifier.		
unit	Unit	The Unit information associated with this service capacity, used to define what is available.	unit	True
available	Available	The number of units available as of the last update.	Number	True
maximum	Maximum	The maximum number of units that can be available for this service, if applicable	False	False
description	Description	A Human-Friendly description of this service capacity e.g. "Beds available for people experiencing homelessness"	String	False
updated	Date Updated	The datetime when this service_capacity object was last updated or changed. Should have millisecond accuracy.	String (datetime)	True

Proposed `service.capacities` field

These changes affect the `service` schema.

We propose adding a new field, `service.capacities` to model adding capacity information to service records.

Field	Title	Description	Type	Required?
capacities	Capacities	The details of	Array of	False

		capacities of this service. See <code>service_capacity</code> .	<code>service_capacity</code>	
--	--	---	-------------------------------	--

API Changes

While these changes do not directly affect the HSDS API Specification, we expect that conformant APIs will return `service` objects with the new `service.capacities` field, `service_capacity`, and `unit` objects where they are present. This is because some API calls return data formatted with the latest definitions of `service`.

Further Considerations

Dependencies

We cannot identify any direct dependencies of this work. Service capacity is an inherently tricky thing to model, and has a relationship with eligibility that must be acknowledged. For example; a service may provide a certain number of beds available to everyone, and another number of beds reserved for a particular group of people.

We have chosen to provide a minimal model for modeling service capacity, in order to leave room for other work related to eligibility to inform design choices as to how we may best model this in the future. Getting the model wrong now would require a MAJOR version change to correct, which we see as undesirable at this point.

Risks

We can identify a single risk stemming from this proposal: if services records are not updated regularly, then there is a risk of digital systems misrepresenting the service capacity. This could have knock-on effects in the real world, which directly affect service users.

We believe that our model mitigates this risk slightly by making `service_capacity.updated` a mandatory field within the `service_capacity` schema. This design choice is such that the last available capacity update is always recorded and we hope that interfaces use this data appropriately e.g. to highlight

potentially stale information to people using the systems so that they may adjust their expectations appropriately.

Documentation

Existing HSDS documentation will automatically update to include information about the new fields on the `service` schema. We will manually add the `unit` and `service_capacity` schemas to the Schema Reference page of the HSDS documentation.

It may be necessary to provide examples and guidance on how to use the new fields and structures added by this proposal. We believe that this can be done responsively based on feedback from the community.

Considerations for existing Profiles

We are not aware of any profiles which implement similar or conflicting behavior, and so we believe that there are no direct impacts on existing prominent HSDS Profiles.

Existing profiles may choose to rebase onto the latest version of HSDS if they wish to make use of the new features, or else explicitly override them.

Considerations for existing Publishers

These changes have been implemented in a way which accommodates existing publishers and does not render any existing HSDS data invalid. This is because the new features constitute a MINOR upgrade to HSDS.

Existing publishers and implementers may choose to ignore the new features and continue to publish valid HSDS 3.0 data.

If publishers and implementers wish to make use of the new features, they will need to adjust their systems to handle the slight increase in complexity represented by this proposal.

In particular, publishers storing records in a normalized database such as SQL or PostgreSQL may need to adjust their schemas to support both the new `unit` and `service_capacity` objects as a separate entity (or table, in tabular representations) but also the fact that the `service` schema must be updated with a new field.

While we believe that this is a relatively straightforward adjustment to make to systems, we also recognize that such adjustments take time and effort to

accommodate when consuming and storing HSDS records and in adjusting analysis workflows.

Considerations for Data Users

These changes introduce new capabilities for data users with minimal costs in terms of increased complexity of the data. Human analysts and digital systems will benefit from the new schemas and fields which allow for declaring known service capacities.

These changes will permit digital systems to display this information in interfaces, making it easier for service users and referral workers to see at-a-glance the estimated capacity of the service. Human analysts will also be able to evaluate service capacity within a given locale, or across a particular sector. This could aid in planning future service provision, directing funding efforts, or evaluating service provision efficacy across time.

The new fields and schemas are optional, which means that data users do not need to adjust analysis or computational workflows to accommodate them if they are not required. This therefore represents a minimal increase in complexity for existing data users.

Review

Upon reviewing this proposal please complete the table below:

Name	Date Reviewed	Accept ⁴ (Y/N)	Comments

⁴ Accepted proposals may not be integrated immediately into the next version of the specification based on capacity and resources available to the implementation team. If not accepted, they will be revisited at the next development cycle.