

# HOME - European Digital Data Standard - Version 1.0



## Introduction

The HOME Digital Data Standard is build to create a univocal way to share student housing data. It also serves as a the technical specifications document to use when creating a JSON Feed, which allows housing providers and housing intermediaries to publish their student housing offers on the HOME solution, and with that, in the Erasmus+ App.

The HOME Digital Data Standard has been developed by the HOME consortium consisting of HousingAnywhere, European University Foundation (EUF), Politecnico di Milano University, Erasmus Student Network (ESN), International Union of Property Owners (UIPI) and Confia International.

Extensive research has been done on data used by large housing providers. A diverse selection of both parties focused on student housing specifically, as more general housing providers was used to create a broader and completer view in housing data. After mapping and sorting all data, various feedback session have been held with student housing providers to assess the completeness and useful of this standard. In the end resulting in the HOME Digital Data Standard.

## Update history

Version	Date	Updates
0.1	15/06/2021	First version
0.2	12/08/2021	Updated with HOME cohort feedback & technical feasibility Quality Labels.
0.3	08/12/2021	Updated with technical feedback from HA Engineering team to make the DDS more robust and fitting to the technical architecture.
0.4	02/09/2022	Updates done after market feedback from both public and private housing providers. Versioning updated to 0.1, 0.2, etc. to have 1.0 as launching version.
1.0	16/09/2022	First official version.

## Using the HOME feed

This document outlines the specifications for supplying room and apartment listings to be published on the Erasmus+ app. This format follows the JSON specification, so only valid JSON data can be accepted.

### Usage example:

1. A housing platform or provider of student housing has several listings they want to publish on the Erasmus+ app.
2. They create a JSON feed in the specified format with all the listings they want to share.
3. The created URL will be added to the HOME solution importer service and via the HOME API published on the Erasmus+ App.

4. Each feed will be read and updated once every day.
5. For each new unique listing added to the feed by the provider, a new listing will be published in the Erasmus+ app, based on the id (unique ID managed by the partner)
6. If an existing listing is no longer found in the feed it will be de-listed from the Erasmus+ App.
7. If a known listing is found in the feed, the existing listing will be updated in the Erasmus+ App.

## Data Fields

These fields are created to form the European Digital Data Standard for student housing. Next to general housing fields related to e.g. address, availability and price it includes fields that represent certain Quality Labels defined for student housing. These Quality Label are created within the Erasmus+ HOME project. A project that intends to standardize the offering and thereby increase the quality of student housing in Europe. An overview of the Quality Labels can be found further on in this document.

### General

Name	Type	Description	Required	HOME Quality Label connected
id	string (max 100 characters)	A unique ID for the listing managed by the external partner.		
groupid	string (max 100 characters)	A unique group ID, to be only used by PBSA's (purpose built student accommodation) or partners with multiple units of the same type. Units of the same unit type will be aggregated based on this field, while taking into account the correct availability.		
listingUrl	string	The URL of the property/listing from the partner. This URL should link back to the human readable listing page on the partner platform with all the listing details, terms and conditions to book the listings, and with the functionality to book the property.		
lastUpdated	string(timestamp)	A timestamp that is last updated on any change in the listing fields. Timestamp format is ISO 8601 compliant. Example, 2021-12-01T11:05:46.000Z		
housingType	string(enumeration)	The type of a listing. Possible values: house, apartment, studio, room.		
propertyType	string(enumeration)	The subtype of a listing. Possible values: private_property, university_owned_residence_hall, private_residence_hall		
address	object	The address comprises of the following components housenumber (string) street (string) city (string) state (string) neighbourhood (string) postalCode (string) countryCode (ISO ALPHA-2) (string)		
longitude	float	To be able to show the listing on the Map view of the Erasmus+ app, we need the longitude & latitude. If not provided, the map view for the listing will be disabled.  The latitude and longitude should be given as strings, internally it is casted as a pair of doubles to represent degrees latitude and degrees longitude. Unless specified otherwise, this must conform to the WGS84 ( <a href="http://www.unoosa.org/pdf/icg/2012/template/WGS_84.pdf">http://www.unoosa.org/pdf/icg/2012/template/WGS_84.pdf</a> ) standard. Values must be within normalized ranges.		
latitude	float	To be able to show the listing on the Map view of the Erasmus+ app, we need the longitude & latitude. If not provided, the map view for the listing will be disabled.  The latitude and longitude should be given as strings, internally it is casted as a pair of doubles to represent degrees latitude and degrees longitude. Unless specified otherwise, this must conform to the WGS84 ( <a href="http://www.unoosa.org/pdf/icg/2012/template/WGS_84.pdf">http://www.unoosa.org/pdf/icg/2012/template/WGS_84.pdf</a> ) standard. Values must be within normalized ranges.		
title	string	Name of the listing in plain text. Maximum of 50 characters. No html/markdown allowed, this is used to indicate the listing with a more familiar name.		
description	string	The full description of the listing in plain text. Should be at least 50 characters long. Maximum possible character limit is 8,000 characters. Markdown is allowed but with limitations upto the discretion of the client.		

rent	object	The rent object consists of the following components amount (integer) (in cents)	✓	✓
securityDeposit	object	The security deposit consists of the following components. amount (integer) (in cents)		
costs	Mapping [costs: CostStructure]	A mapping of listing costs (all other costs except monthly rent). For specifications see the 'costs' section below.	✓	✓
currencyCode	string	ISO 4217 currency code. For example: "EUR". Unless otherwise stated, this currency code will apply globally to all money related fields in the listing. Please make sure to use the official local currency of the country the listing is located in.	✓	✓
availability	object	Date range depicts the listing's availability, the components are startDate (YYYY-MM-DD) (string) endDate (YYYY-MM-DD) (string) (optional, if omitted listing will be shown as available till the end of time)	✓	
rentalConditions	object	Conditions around the rent, has the following components minimumStayMonths (integer) (The minimal rental period in months)		✓
facilities	Mapping [facilities: FacilityStructure]	A mapping with all currently supported facilities and whether this listing has them or not. For specifications see the 'facilities' section below.	✓	✓
rules	Mapping [rules: RulesStructure]	A set of rules that are to be respected for the listing. For specifications, see 'rules' section below.		
media	Array [object]	The media object consists of the following components, type (enumeration) (Possible values - image_raw - Images will be downloaded and scaled/compressed if needed. Minimum image size is 500KB Maximum file size per image is 50MB. video_raw - Raw videos will be downloaded and optimised for viewing. video_optimized - Video links that are already optimised for viewing e.g links from youtube, vimeo etc ) url (string) (URL should point directly to the media resource. Each media resource will be downloaded and scaled/compressed if needed. Minimum image size is 500KB Maximum file size per image is 50MB, maximum file size per video is 500 MB) title (string) (characteristic description of the media, e.g 'living_room_image')	✓	
maximumOccupancy	integer	Total number of places in the unit that can be occupied by habitants.		
floorNumber	string	Number of the floor/etage. Default value is unknown, this is a string with values like "1", "2", "3", etc. Internally its casted to an integer.		
bedroomCount	integer	Number of total bedrooms in the apartment		
bedroomSize	integer	In square meters, 0 denotes unknown.		
totalSize	integer	In square meters, 0 denotes unknown.		
contractEnglishTranslation	string(enumeration)	Is a translation of the rental contract available in English? Possible values: unknown,true, false.	✓	✓
communicationEnglish	string(enumeration)	Is communication with the landlord available in English? Possible values: unknown,true, false.	✓	✓
bookableOnline	string(enumeration)	Can the property/listing be booked online? Possible values: unknown,true, false.	✓	✓
internationalGuarantor	string(enumeration)	Is it allowed to have an international guarantor? Possible values: unknown,true, false.	✓	✓
energyLabel	string(enumeration)	The energy label of the property with possible values: A, B, C, D, E, F or G.		

## Costs

All costs, except monthly rent and security deposit, are presented as a mapping of costs objects.

These cost types are the most common cost types present in student housing and the ones having a serious impact on the monthly rental price.

The cost type `other-additional-costs` is created to be able to add any other costs than the standard cost types.

Key	Type	Description	Required	HOME Quality Label related
electricity	string (enumeration)	Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown	✓	
gas	string (enumeration)	Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown	✓	
water	string (enumeration)	Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown	✓	
internet	string (enumeration)	Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown	✓	
administrationFee	string (enumeration)	administrationFee can either be the administration fee charged by the booking platform or an administration fee to be paid to the property provider. Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown	✓	
cleaningServices	string (enumeration)	Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown	✓	✓
otherAdditionalCosts	string (enumeration)	The cost otherAdditionalCosts is created to be able to inform the tenant of any other costs that are not part of the above costs. Examples: waste taxes, broadcasting fee, etc. Possible values: included_in_rent,excluded_from_rent, not_appliable, unknown		

## Facilities

These are all the facilities and the possible values for each facility currently supported.

- If a facility is not listed in the feed it will be set to the default.
- The default for each facility is “unknown” (will not be shown on the User Interface).

Key	Type	Description	Required	HOME Quality Label related
furnished	string (enumeration)	Possible values: unknown,yes, no	✓	
bedroomLock	string (enumeration)	Possible values: unknown,yes, no	✓	✓
balconyTerrace	string(enumeration)	Possible Values: unknown, no, shared, private		
extraStorage	string(enumeration)	Possible Values: unknown, no, shared, private		
bathroom	string(enumeration)	Possible Values: unknown, no, shared , private	✓	
kitchen	string(enumeration)	Possible Values: unknown, no, shared, private	✓	
commonArea	string(enumeration)	Possible Values: unknown, no, shared, private		
carParking	string(enumeration)	Possible Values: unknown, no, shared, private		
bicycleParking	string(enumeration)	Possible Values: unknown, no, shared, private		
toilet	string(enumeration)	Possible Values: unknown, no, shared, private	✓	
wheelchairAccessible	string (enumeration)	Possible values: unknown,yes, no	✓	✓
visuallyImpairedAccessible	string (enumeration)	Possible values: unknown,yes, no		
airconditioning	string (enumeration)	Possible values: unknown,yes, no.	✓	
bed	string(enumeration)	Possible values: unknown,yes, no, single, double	✓	✓
closet	string(enumeration)	Possible values: unknown,yes, no	✓	✓
desk	string(enumeration)	Possible values: unknown,yes, no	✓	✓
chair	string(enumeration)	Possible values: unknown,yes, no	✓	✓

dishwasher	string(enumeration)	Possible values: unknown,yes, no	✓	✓
kitchenware	string(enumeration)	Possible values: unknown,yes, no	✓	✓
bath	string(enumeration)	Possible values: unknown,yes, no		
shower	string(enumeration)	Possible values: unknown,yes, no		
dryer	string(enumeration)	Possible values: unknown,yes, no	✓	✓
dryingRack	string(enumeration)	Possible values: unknown,yes, no	✓	✓
tv	string(enumeration)	Possible values: unknown,yes, no		
washingMachine	string(enumeration)	Possible values: unknown,yes, no	✓	✓
wifi	string(enumeration)	Possible values: unknown,yes, no	✓	✓
heating	string(enumeration)	Possible Values: unknown,yes, no	✓	
smokeAlarm	string(enumeration)	Possible values: unknown,yes, no		
24hrConciergeReception	string (enumeration)	Possible values: unknown,yes, no	✓	✓
videoSurveillance	string(enumeration)	Possible values: unknown,yes, no	✓	✓
elevator	string(enumeration)	Possible values: unknown,yes, no		
oven	string(enumeration)	Possible values: unknown,yes, no	✓	✓
stove	string(enumeration)	Possible values: unknown,yes, no	✓	✓
iron	string(enumeration)	Possible values: unknown,yes, no	✓	✓
ironBoard	string(enumeration)	Possible values: unknown,yes, no	✓	✓
sportsFacilities	string(enumeration)	Possible values: unknown,yes, no	✓	✓
pool	string(enumeration)	Possible values: unknown,yes, no	✓	✓
gameRoom	string(enumeration)	Possible values: unknown,yes, no	✓	✓
sharedStudyArea	string(enumeration)	Possible values: unknown,yes, no		
microwave	string(enumeration)	Possible values: unknown,yes, no	✓	✓
bedLinen	string(enumeration)	Possible values: unknown,yes, no	✓	✓
pillowDuvet	string(enumeration)	Possible values: unknown,yes, no		
towels	string(enumeration)	Possible values: unknown,yes, no	✓	✓

## Rules


A set of rules linked to the listing.

Key	Type	Description	Required	HOME Quality Label related
animals	string(enumeration)	Possible Values: unknown, allowed, not_allowed, discussable		
playingMusic	string(enumeration)	Possible Values: unknown, allowed, not_allowed, discussable		
smoking	string(enumeration)	Possible Values: unknown, allowed, not_allowed, discussable, outsideOnly		
couples	string(enumeration)	Possible values: unknown,allowed, not_allowed, discussable		
overnightVisitors	string(enumeration)	Possible values: unknown,allowed, not_allowed, discussable		

## HOME Quality Labels

Below is the logic behind the different HOME Quality Labels that is implemented. A listing should adhere to the rules of a certain Quality Label to get it labelled accordingly. The rules are based on a selection of the data fields above or a logic based on these data fields.

Icon	Label	Criteria	Indicator	Rules
	<b>International friendliness</b>	Accommodation adequate to the needs of international students  <i>A listing should qualify for all the mandatory fields, and at least one of the non-mandatory fields.</i>	• Wi-fi ( <b>mandatory</b> )	wifi = yes
			• Bed ( <b>mandatory</b> )	bed = yes OR single OR double
			• Desk ( <b>mandatory</b> )	desk = yes
			• Online booking available ( <b>mandatory</b> )	bookableOnline = true
			• Short-term contract available ( <b>mandatory</b> )	minimumStayMonths < 4
			• Basic furniture	closet = yes and chair = yes and
			• Contract translation in English available ( <b>mandatory</b> )	contractEnglish = true
			• International guarantor allowed	internationalGuarantor = true
• Communication in English	communicationEnglish = true			
	<b>Room Quality</b>	Room provided with all necessary furniture.  <i>A listing should qualify for 4 out of 6 indicators to get the label.</i>	• >single bed	bed = double
			• Bed linen	bedLinen= yes
			• Pillow and Duvet	pillowDuvet = yes
			• Bath towels	towels = yes
			• Desk and chair	desk = yes and chair = yes
			• Closet/Drawers	closet= yes
	<b>Supersecure</b>	Accommodation providing above average safety features, including against fire accidents	• Lock on room door	bedroomLock = yes
			• Surveillance of the building (either video or concierge)	24hrConciergeReception= yes OR videoSurveillance = yes
	<b>Premium Accommodation</b>	Accommodation providing exclusive features  <i>A listing should qualify for 3 out of 5 indicators to get the label.</i>	• Sports facilities	sportsFacilities = yes
			• Pool	pool= yes
			• Game Room	gameRoom= yes
			• Surveillance of the building (either video or concierge)	24hrConciergeReception= yes OR videoSurveillance= yes
			• Cleaning service	cleaningServices = present in costs
	<b>Well-equipped kitchen</b>	Well-equipped accommodation, displaying above average kitchen appliances.  <i>A listing should qualify for all the mandatory fields, and at least 2 out of 3 non-mandatory fields.</i>	• Stove ( <b>mandatory</b> )	stove = yes
			• Oven ( <b>mandatory</b> )	oven = yes
			• Microwave	microwave = yes
			• Dishwasher	dishwasher= yes
			• Kitchenware	kitchenware= yes
	<b>Well-equipped laundry</b>	Well-equipped accommodation, displaying above average laundry appliances  <i>A listing should qualify for the mandatory field, and at least 2 out of 3 non-mandatory fields.</i>	• Washing machine ( <b>mandatory</b> )	washingMachine= yes
			• Iron	iron = yes
			• Iron board	ironBoard= yes
			• Drying rack / Dryer	dryer= yes OR dryingRack = yes

	<b>Wheelchair Accessible</b>	Accommodation entirely accessible for persons with reduced mobility and/or using a wheelchair	<ul style="list-style-type: none"> <li>• Wheelchair accessible</li> </ul>	wheelchairAccessible= yes
---	------------------------------	---	---	---------------------------

### Feed Example

This is an example feed result for 1 room in Copenhagen.

```

{
  "id": "AZ-18-004-01-01 H",
  "groupdId": "AZ-18",
  "listingUrl": "https://www.partnerwebsite.com/listings/unique-
id",
  "lastUpdated": "2021-12-01T11:05:46.000Z",
  "housingType": "apartment",
  "propertyType": "private_property",
  "address": {
    "houseNumber": "40",
    "street": "Coolsingel",
    "city": "Rotterdam",
    "state": "South-Holland",
    "postalCode": "3011 AD",
    "countryCode": "NL"
  },
  "latitude": 9.94948,
  "longitude": 48.42339,
  "title": "Room in Apartment Klerkegade",
  "description": "This is an example room description, should be
50 characters atleast.",
  "rent": {
    "amount": 500000
  },
  "securityDeposit": {
    "amount": 800000
  },
  "costs": {
    "administrationFee": "excluded_from_rent",
    "electricity": "included_in_rent",
    "gas": "included_in_rent",
    "internet": "included_in_rent",
    "water": "included_in_rent",
    "cleaningServices": "unknown",
    "otherAdditionalCosts": "not_applicable"
  },
  "currencyCode": "EUR",
  "availability": {
    "startDate": "2021-10-01",

```

```
        "endDate": "2026-11-30"
    },
    "rentalConditions": {
        "minimumStayMonths": 2
    },
    "facilities": {
        "furnished": "yes",
        "bedroomLock": "no",
        "balconyTerrace": "private",
        "extraStorage": "no",
        "bathroom": "shared",
        "kitchen": "shared",
        "commonArea": "shared",
        "carParking": "no",
        "bicycleParking": "yes",
        "toilet": "shared",
        "wheelchairAccessible": "no",
        "visuallyImpairedAccessible": "yes",
        "airconditioning": "yes",
        "bed": "single",
        "closet": "yes",
        "desk": "yes",
        "chair": "yes",
        "dishwasher": "no",
        "kitchenware": "yes",
        "bath": "yes",
        "shower": "no",
        "dryer": "no",
        "dryingRack": "yes",
        "tv": "yes",
        "washingMachine": "yes",
        "wifi": "yes",
        "heating": "yes",
        "smokeAlarm": "yes",
        "24hrConciergeReception": "yes",
        "videoSurveillance": "no",
        "elevator": "no",
        "oven": "yes",
        "stove": "yes",
        "iron": "yes",
        "ironBoard": "unknown",
        "sportsFacilities": "no",
        "pool": "no",
        "gameRoom": "no",
        "sharedStudyArea": "unknown",
        "microwave": "no",
        "bedLinen": "unknown",
        "pillowDuvet": "unknown",
        "towels": "yes"
    },

```



```

"rules": {
  "animals": "not_allowed",
  "playingMusic": "discussable",
  "smoking": "outsideOnly",
  "couples": "allowed",
  "overnightVisitors": "allowed"
},
"media": [{
  "type": "image_raw",
  "url": "https://cdn.example.com/photos/1.jpg",
  "title": "living_room_image"
},
{
  "type": "image_raw",
  "url": "https://cdn.example.com/photos/2.jpg",
  "title": "bedroom_image"
},
{
  "type": "image_raw",
  "url": "https://cdn.example.com/photos/3.jpg",
  "title": "shed_image"
},
{
  "type": "video_raw",
  "url": "https://cdn.example.com/videos/1.mp4",
  "title": "walkthrough"
}
],
"maximumOccupancy": 1,
"floorNumber": "4",
"bedroomCount": 1,
"bedroomSize": 14,
"totalSize": 140,
"contractEnglishTranslation": "true",
"communicationEnglish": "true",
"bookableOnline": "true",
"internationalGuarantor": "true",
"energyLabel": "C"
}

```

## Validation Tool

The HOME consortium created a validation tool for developers to check the structure of their JSON Feed. It is built by exposing an endpoint to which you can send the JSON set-up of one single listing/advertisement. It will return any errors in the set-up or an "OK" if the set-up is correct.

The endpoint can for example be accessed using a program like Postman, to fire a POST request to the validation tool. Make sure you set the Content-Type in the headers on 'application/json'

request: POST '<http://api.housing.uni-foundation.eu/validate>'  
header 'Content-Type: application/json'

```
curl --location --request POST 'http://api.housing.uni-foundation.eu
/validate' \
--header 'Content-Type: application/json' \
--data-raw '{
```

For the raw-date please use your JSON set-up for one single listing/advertisement. An example can be the feed example in the section above.

